CHAPTER 1
INTRODUCTION

1.1 Background

Software engineering is a systematic, disciplined, quantifiable approach to design, develop, operate, and maintain the software. It admits the techniques and procedures, frequently determined by a software development process, with the purpose of improving reliability and maintainability of software systems. The discipline of software engineering includes knowledge, tools and methods for requirement analysis, design, construction, testing and maintenance. There are several research area in Software Engineering such as, Requirement Engineering, Reengineering, Usability Engineering, Object Oriented Software Engineering, Client Server Software Engineering, Empirical Software Engineering, Metrics and Measurement, Usability Engineering etc.

Requirement engineering is one of the important and trust research area in Software Engineering. It is the most effective phase of software development process, which involves collecting the effective requirement from the user. The target of the requirement engineering is to gather the quality requirement from the stakeholder. It helps to analyze information system and planning of software systems. Also, it is a systematic approach through which the software engineer collects requirements from different sources and implements them into software development processes. Requirement engineering contains a set of activities for discovering, analyzing, documenting, validating and maintaining a set of requirements for a system.

Furthermore, the successful software depends on effective requirement elicitation. Elicitation is an effort made to gather the project

requirements from the concerned stakeholders. Unfortunately, this process has not been considered by various analysts due to this, it increases the cost and time Moreover, it can also lead to the project which does not meet the demands of each stakeholder or in the worst case lack of requirement elicitation can also leads to software failure [17]. Thus, it can be said that the requirement elicitation is of paramount importance in order to clearly define the requirements of each and every stakeholder. Requirement elicitation involves various techniques for clearly identifying and understanding the project requirements. As per the report of Standish Group, the primary cause of failure of a software project is the incomplete and sometimes unsatisfactory requirements [155].

The leading cause of poor requirements is the lack of proper elicitation. Moreover, ineffective elicitation often causes budget overruns and redesigning the entire process. Elicitation is also essential in a way that it helps the stakeholders to understand the business problem accurately. So, it is crucial for the analysts performing the elicitation, to produce the requirements that are clearly understandable, useful and relevant [15].

Business analysts sometimes think that requirements gathering is an easy task, but this task is just like digging an archaeological site i.e., the real information is buried deep down and the titbits that are initially dug can only indicate what lies beneath but still it is insufficient and requires more study and digging. So, it becomes essential that an appropriate elicitation technique is used to gather relevant information from the stakeholders [17].

The current artifact analysis technique can be used when the business is well structured with proper documentation. The Root cause analysis uses the approach of asking the “5 Whys” which helps in

understanding the root cause of a problem. This approach involves interactions with the stakeholders so as to understand and define the business problem. Then the stakeholders are contacted in order to understand why a specific problem is occurring. This way, usually after asking “why” 5 times, the analyst is able to find the root cause of a problem. This technique helps to fully understand a business problem before moving into the solution mode.

Observation is another useful method to define a business problem. Observations allow the analysts to understand the interaction of the stakeholders with the system. The most effective way to receive a vast amount of information at once is Brainstorming. This allows the analysts to come up with multiple ideas [15]. Moreover, this method helps to understand the process and uncover the information that has not been discussed before.

Interviews are also an effective means to get information [17]. The opportunity of interacting with the stakeholders allows clear details on the requirements of the projects and any doubts of the analysts or the stakeholders can be cleared then and there. The survey is another method of information elicitation that allows information gathering from different stakeholders of the same project.

Requirements workshops are one of the most effective techniques in requirements elicitation. It also causes an effective team building. Prototyping is also a useful tool for business analysts to determine if the solution is actually as per the requirements of the stakeholders. It also allows the stakeholders to give their suggestions and feedbacks. Requirement determination techniques should be wide enough to set up boundary conditions for the target system, however, they still should

focus on the establishment of requirements as different to design activities.

The information gathering should start with the organization for which the product has to be developed and the context of the information has also to be recorded in order to understand the need of the stakeholders. Inefficient elicitation techniques can lead to ineffective problem statements and ultimately the inefficient project. This is because the superficial elicitation techniques are not multidimensional and hence do not gather the information effectively. Therefore it is imperative to choose the right data elicitation technique in order to define the problem statement correctly. The elicitation techniques should be stakeholder centric and must focus on the requirements of the stakeholder rather than the model designing part.

1.1.1 Quality Requirements

Quality requirements are often neglected or compromised while concentrating on end user requirements. These quality requirements are safety, reliability, performance etc. Though they are non-functional, yet they describe system functionality. This sometimes done to cut down the cost and as a result software contracts generally do not have specific quality requirements but some generalized elements of quality.

1.1.2 Security in Software

As assets got connected to software, security concerns for software grew more and more. Over the years, the level of threats to software systems have varied depending upon the environments in which software systems are used and how they process the data [10]. Internet and connectivity were considered as a social and commercial breakthrough but it has increased the opportunities for those who want to electronically exploit others. Software attacks and hacking incidents actuated the

software world to concentrate on the security of the software product. Constant and successful attack stories make it clear but still software is developed with vulnerabilities inside [10].

1.1.3 Secure Software Engineering

The focus of security requirement engineering in the early stages of the software development lifecycle proves to be cost effective and brings out robust design [13]. For collecting information related to the security of software, various information gathering methods are used.

Security requirement elicitation techniques are divided into two categories i.e. formal and informal. Formal security requirement elicitation techniques follow a systematic procedure and a definite number of steps [23].Lack of flexibility and limitation in defining the scope of (formal) security requirement elicitation techniques limit the use and efficiency of technique. Due to limitations in the scalability of the method, formal techniques are hard to use for big projects [23]. There are predefined roles and repetition of processes informal techniques to conduct the requirement specification.

On the other hand, informal techniques are more flexible and it becomes easy to define the steps according to the nature of the system under consideration. Despite the flexibility of informal techniques, they lack a systematic approach. Repetition of the process in informal techniques is not easy as compared to formal methods. Since there are no predefined steps in informal security requirement elicitation techniques, most of the software professionals avoid using them [12].

Informal security requirement elicitation techniques lack a proper structure and usually leave a few security problems unaddressed. In addition, particular roles are not defined for the stakeholders in order to

conduct the specification [23]. These are some of the main reasons that informal techniques are generally avoided by security requirement engineers.

The main idea behind security requirement elicitation is to use elicitation methods, which are used for collecting requirements related to the functioning of various requirements like security. Since elicitation techniques are focused on functional requirements [2, 5], it is difficult to use the same technique for capturing non-functional requirements. For this purpose, many of the elicitation techniques were extended and were used in an extended form [5], to capture requirements that are non-functional (e.g. security). Figure 1.1 presents, the various phases to resolve the issues of software development process.

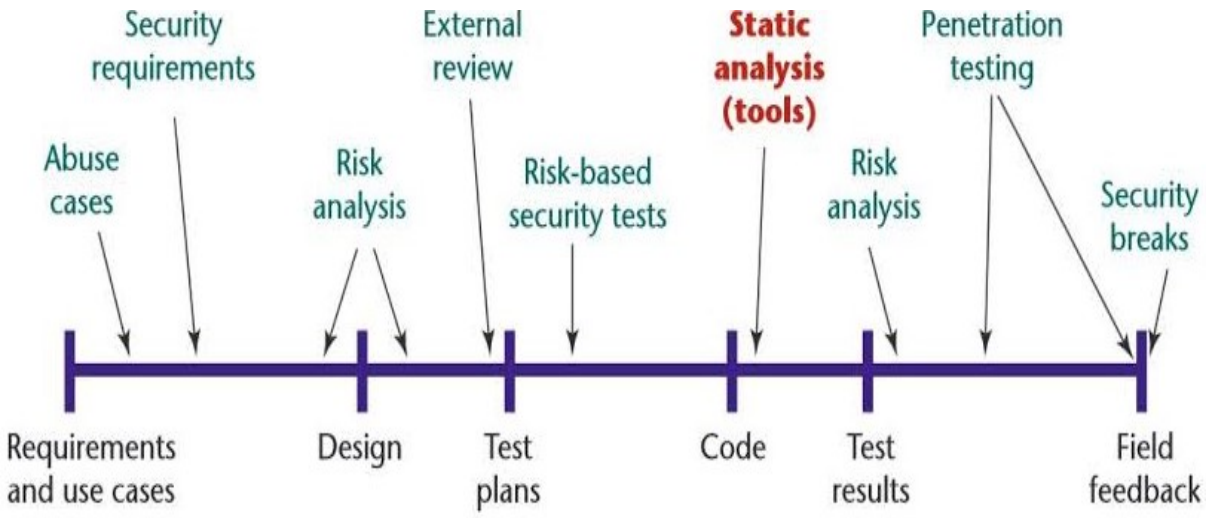

Fig.1.1Various Steps of Software Development Process

Researchers have given various techniques for information gathering related to security [3, 27] where some of the techniques were combined with other elicitation techniques [2, 26] to produce better results. All the ongoing and past work [5,14,16,22,23] have aimed upon developing attack resistant and secure software systems.

1.2 Importance of Security Requirement Engineering

The importance of security requirements is that if in any project, the security requirements are not taken into consideration, then the system can be tested only during its implementation. This causes a problem if during the implementation, the project is found to be a failure then the whole task of production has to be repeated. If security requirements are considered, they are discussed independently from activities of requirement engineering. This again leads to undermining of security aspects.

As discussed before, the environment in which a project has to work is dynamic and liable to change so it is clear the security issues also keep on changing. Thus, the security requirement activity is not a onetime exercise but an iterative phenomenon. Users have certain assumptions for the software they use. These assumptions need to be converted in the form of security requirements during the development stage. Another critical point of concern is that of an attacker. The attacker keeps on looking for the conditions that allow for an attack. So, it becomes imperative for the requirement engineer to think with a point of view of the attacker while developing software.

The attack patterns are very well discussed in chapter 2 of Software Security Engineering: A Guide for Project Managers [5]. Security is an important issue and associated with more processes. Often, the security requirements are considered as negative requirements, and so the presumption that “the system will not allow any kind of attack” is usually not feasible. For validating the mechanism such as levels of security, backups etc, it is essential to understand the assets and essential services which are shown in figure 1.2.

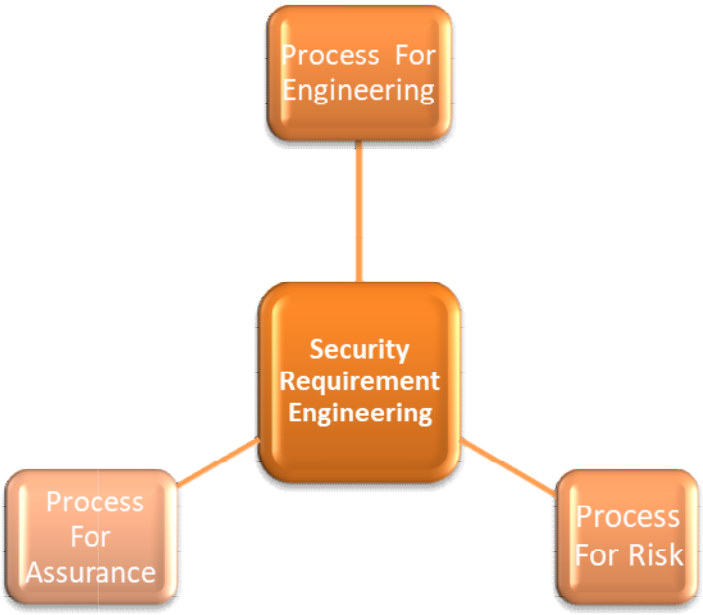

Fig. 1.2 Security Requirement Engineering Process

1.2.1 Significance of Requirements Engineering

Requirement Engineering (RE) is a very important process of software engineering and it starts with the collection of information for determining requirements. It is also known by the term requirement elicitation. As stated by various researchers, poor requirement engineering is the most significant failure of many software projects. In various surveys [1,7,10] poor requirement engineering was the cause of almost 50% of failed products.

Stakeholders are consulted for requirements engineering. These stakeholders in turn gather information from end users, and developers. Not only this, the stakeholders also take note of the organizational structure and the legalities and the general environment of the organization where the desired system is to be used. Since, different stakeholders have different approaches to gathering information, so one technique of data eliciting is not sufficient. Requirement Elicitation is all about gathering information and understanding what a customer or various

stakeholders want. It is the most critical and information-based activity of requirement engineering.

It is evident from various surveys [7] and reports that improper, incomplete requirement elicitation leads to software project failure. So, there is a need to improve effective requirement gathering and more attention is required in the elicitation process. It is necessary to understand the requirement elicitation methods and it is also imperative to use them in a different context. So now it is clear that requirement engineering plays a vital role in the success of a project.

A study has shown a return on investment ranges around nearly twenty percent if security analysis is done at the start of development. As per the report of the National Institute of Standards and Technology (NIST), faulty software in terms of reliability and security needs $59.5 billion in a year for repairs. This is a direct loss to the economy. So, it can be seen that improvements in this area is required in order to save money and resources. The anomalies based on requirements are among the significant reasons of:

- Over budgeted projects
- Delay in projects
- Reduced and cancelled projects
- Poor quality applications
- Products that are not used significantly

Moreover, requirement engineering is done in a changing environment that increases complexity. Requirement engineering on various projects faces several problems like:

- Not including all stakeholders which can lead to incomplete data requirements.

- ❖ Requirements emphasize on implementation part rather than describing the actual problem.
- ❖ Requirements are decided without any modeling or simulation resulting in an inefficient project.
- ❖ There is ambiguity in the requirements specification. Sometimes they are not testable and they are also not capable of validation.
- ❖ Weak requirements management is another problem. It means that the information gathering is often restricted to prioritizing, and scheduling of data.

1.3 Security Requirement Elicitation

Rushby describes the concept of security as something "which must not happen" [24]. This description of security is quite vague but it is used by Sindre create an informal security requirement elicitation technique [25]. If we take the inverse of this description i.e. what "must happen", we touch use cases. The use case technique is a pictorial and textual representation of the functional requirements of the software. That means, a representation of things that "must happen" in a software system. Misuse cases are said to be inverse of Use cases [25].Figure 1.3 shows the specific domain of the Elicitation Process.

In other words, we can say Misuse cases represent "what must not happen" in a software system. This description of the Misuse case technique is vague and ambiguous which is the same as the concept of security described in Rushby's [24]. Misuse case is an informal security requirement elicitation technique which is an extension of Use case requirement elicitation technique [20]. Input for Misuse cases is at least one Use case [16,25,26]. There can be one or many misuse cases for a single use case. Misuse creation depends upon the creativity of the person creating misuse cases [4,14]. It means that there is no measurement of how many maximum misuse cases can be created for a particular use case.

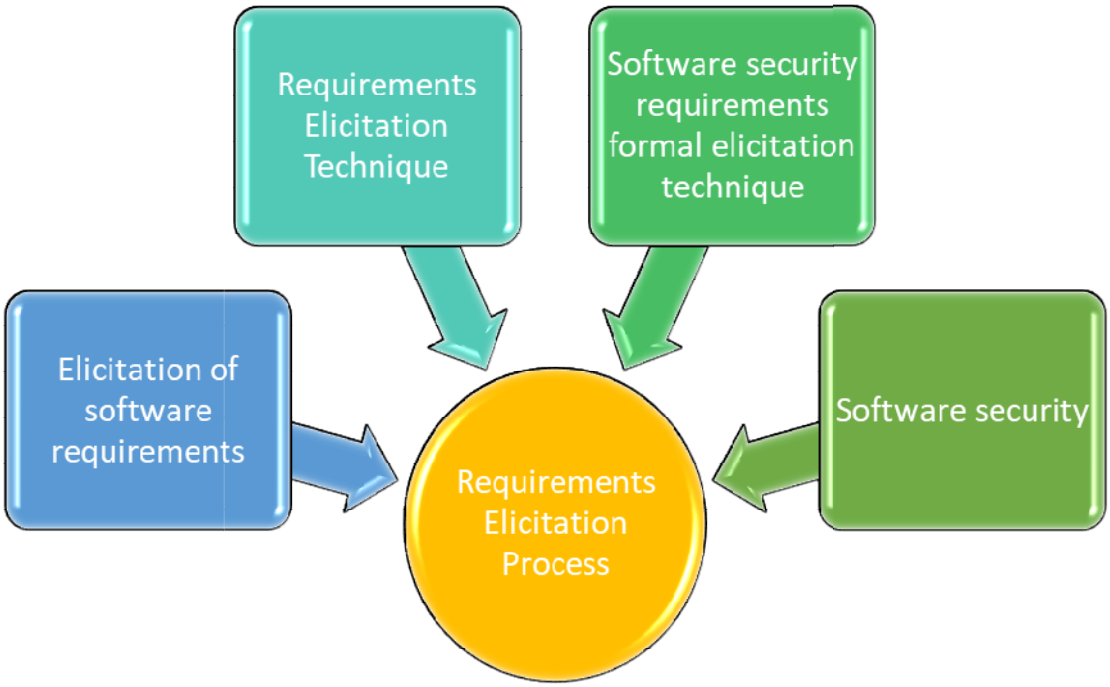

Fig 1.3 Elicitation Process Domains

This vague concept of misuse case creation establishes its relationship with Rushby's description [24]. It may be said that misuse case are a realization of what "must not happen" during software operation. According to Mouratidis, a system's security requirements are defined by a system's security constraints. Same concept of security is presented by others who define security as a system's constraint [18, 19]. This concept of security is seen in I* framework which is an informal requirement elicitation technique. I* uses the idea of goals, actors and dependencies to gather system's requirements. The concept of dependencies used in I* can be defined as "obligations on the actors" [21]. I* is used for getting those requirements of a system that are functional in nature, but using the concept of system's constraint, I* is also used for capturing security requirements [7,20]. TROPOS, a formal technique for requirement elicitation is based upon I* modelling.

Although TROPOS was not developed with security on the mind, a set of security constraints was proposed in order to enable TROPOS to consider security aspects [3,7] Secure TROPOS, an extension of Tropos methodology, emerged as a formal technique for security requirement elicitation [2,12,19]. Secure TROPOS lacks in clear definition of 19 system assets. Moreover, the idea behind Secure TROPOS methodology does not talk about what system services are being constrained and its effect on the system's functionality [11]. If we look at CLASP, it consists of five views which contain twenty-four activities, in total.

Each of the twenty-four activities of CLASP is further divided into discrete process components and linked to one or more specific project roles. CLASP is a formal technique it focuses on removing vulnerabilities in the software environment. CLASP emphasizes more on white box testing [9].Therefore, despite the detail in different views and activities of CLASP, it lacks in determining the types of bugs to be fixed. CLASP activities usually focus on architectural and technical weaknesses. These activities ignore business level threats, CLASP does not have any transparent methodology of how to realize or understand its activities. Moreover, CLASP lacks tool support. We see that not one description of software security encompasses the whole problem area i.e. security issues in software. It is because of the variety of resources involved in software [9].

Different natures of software systems, type of resources involved, different architectural styles and life cycle models are some of the reasons for varying efficiency of security requirement elicitation techniques and frameworks. Some researchers confine security to authorization, integrity, confidentiality and availability of services focus on assets, activities, goals and restrictions. Security requirement elicitation techniques and frameworks are constructed on the concept of security. The way security

is presented by predecessors, in research, is reflected later in the realization of the concepts by the successors in the form of methods and techniques [10,22,23].

Every requirement elicitation technique and framework is the realization of a concept and the way security is presented by one or more researchers [11]. It has become clear, after in-depth study of security concepts, that a flaw or shortcoming in a concept of security is reflected in the realization of the idea later. It means that if a security concept has flaws, almost the same defects are reflected in understanding of the concept whether the concept is realized by the same or different researcher(s).

While studying existing techniques and frameworks, we saw that frameworks proposed in academia are seldom used in industry at large. Most organizations have their own set of steps or framework to address the security needs of a system. To understand the actual use of security requirement elicitation frameworks in industry, we searched for security requirement frameworks used in the software industry [8].

We selected literature available from two organizations i.e. IBM and CISCO. We were surprised to know that these organizations have their own frameworks and set of defined rules to address security in their systems. For example, CISCO uses standards like ISO17799 and NISTSP800. CISCO is using an approach in which they have integrated standards like ISO17799 and NISTSP800 with different steps and procedures to be able to implement security in all phases of software development. IBM created a framework using the holistic approach to business security [6]. The security framework consists of 5 steps that enable the organization to secure the solutions and the services. The framework used by IBM characterized security requirements with respect

to business policies which is again a customized framework. Frameworks proposed by researchers are based on different research methods. Researchers conducted detail studies in order to address security requirements and to make software work better [11, 21].

There is not much literature available which shows that different frameworks proposed by researchers are actually used in the industry on a wide scale. Before proposing a new framework, it was important to look into the published material regarding different frameworks proposed by researchers. For understanding the existing frameworks and know whether we need another framework or not, we conducted a literature review.

1.3.1 Research Questions for Security Requirement Elicitation

a) What are the software security requirements elicitation techniques that are widely used?

b) What are the two widely used techniques for security requirement elicitation i.e. which satisfies these evaluation criteria.

- Easy to Implement
- Flexibility
- Adaptability
- Learnability
- Understandability
- Scalability
- Visual output

c) Which informal technique for security requirement elicitation has the highest rating for following evaluation criteria?

- Easy to Implement
- Flexibility
- Adaptability
- Learnability
- Understandability

- Scalability
- Visual output

These research questions help us in evaluating security requirement elicitation techniques according to the above mentioned criteria.

1.4 Search Terms

Initially, we search papers with keywords and then we combined the keywords with the Boolean operators to get the desired output.

- Software Requirements
- Requirements Elicitation
- Requirements Elicitation Process
- Techniques used to elicit Security Requirements
- Requirements Elicitation Technique
- Formal Requirements Elicitation Methods
- Requirements Gathering
- Formal Requirements Elicitation Technique
- Elicitation of software requirements
- Techniques of requirement elicitation for software
- Methods of software requirements elicitation
- Formal Elicitation Technique
- Informal Elicitation Technique
- Software security
- Security Requirement Elicitation Technique
- Software security requirements
- Software security requirements elicitation
- Software security requirements formal elicitation technique

1.5 Research Issues

The importance of security requirement is that if in any project, the security requirements are not taken into consideration, then the system

can be tested only during its implementation. This causes problem because if during implementation the project is found to be a failure the whole task of production has to be repeated. If security requirements are considered, they are considered independently from activities of requirement engineering. This again leads to undermining of security aspects. As discussed before, the environment in which a project has to work is dynamic and liable to change so it is clear the security issues also keep on changing. Thus, the security requirement activity is not a onetime exercise but an iterative phenomenon. Security is an important issue which is associated with all the steps of SDLC. Often, the security requirements are considered as negative requirements, and so the presumptions that "the system will not allow any kind of attack" are usually not feasible. For validating the mechanism such as levels of security, backups etc, it is important to understand the assets and essential services.

1.6 Statement of the Problem

After discussing all these issues related to our research, we can outline the problem statement as follows:

- The requirements elicitation aims at simplifying the most difficult, most error-prone and most communication intensive software development process.
- Software security requirements elicitation techniques are a tool in which the security lapses in the software development process are found and addressed by gathering the relevant information.
- A critical and best suited attribute of security are identified to improve the requirements elicitation security.
- Out of the many available techniques, the best and the most suitable techniques and approaches are used for requirements elicitation process.

Keeping this in mind, the researcher has formulated a problem as under in order to device a mechanism to predict software security in the development life cycle:

"Security Requirement Elicitation Framework for Secure Software Development"

1.7 Aims and Objectives

In order to achieve the most general goal of working out a framework for security requirement elicitation technique, following objectives are set forth:

- Critically examine the literature on software security, requirement elicitation, requirement elicitation techniques, and security requirement elicitation framework if available.
- To understand common software requirement elicitation mistakes that cause security problems in a system, at later stages.
- Study formal and informal techniques for security requirement elicitation.
- Develop an implementable systematic procedure for security requirement elicitation by integrating soft computing technique with two widely used formal techniques for security requirement elicitation.
- To propose a concrete, implementable framework for security requirement elicitation.
- To validate and test the proposed framework theoretically as well as sample tryouts.

1.8 Significance of the Study

The search was basically focused on different available elicitation techniques related to security requirements and their effectiveness in various situations. Inclusion and Exclusion criteria were also taken into account to search material. In order to find authentic articles, we used checklist points for quality assessment. We applied checklist points by

manually reading the articles. Following are the quality criteria which were taken into consideration:

- The research methodology is understood by the reader.
- Limitations of the study are reported.
- Primary studies were selected based on empirical evidence i.e. data must be available in qualitative or quantitative form.

1.9 Limitations

After reviewing and analyzing all the articles available in online and offline data repositories, for the systematic review, we found some limitations in our systematic literature review. Some search terms or strings were not working correctly to find the articles from databases related to security requirements elicitation. We tried our best to find the articles which are available in the last eight to nine years so that we can find the latest research gaps in this area but yet we may have missed some articles that were not available to us.

- Lack of resources to conduct the experiment i.e. we were not able to find any industrial contact to conduct the experiment. Our experiment required software analysts from the industry. Response from industries was not positive because software organizations have their own ongoing projects therefore it was not possible to engage someone from the industry.
- If we have more time, we should choose a more complex method i.e. more steps in the framework. Just using more time with a simple method will not improve the number of problems found. We had to select the method depending on several issues like the budget of the stakeholders.
- We have a few data points in our experiment that may affect the generalization of results. As mentioned earlier, our results and

conclusions are inferred from one system which was randomly selected.

Our framework is work intensive and we can see from the experiment that its activities need time to execute. This unaddressed issue can be a limitation of the study. The motive of our work was to find a greater number of security requirements than a formal or informal technique. Reducing activities of the framework would simply be at the cost of security requirements. Making a less work intensive Framework and testing its applicability becomes out of scope for us in the present study.

1.9.1 Delimitation of the Method Based on Conversations

These techniques not only give the information for requirement engineering, but also help to understand the feeling and goals of various stakeholders. Through conversation-based techniques it is easy to uncover the details with the help of repeated questioning and follow-up from the person.

1.9.2 Delimitation of Methods Based on Observation

These methods help in getting pivotal information for finding a solution to a problem. These methods are very helpful when the development team is inexperienced.

1.9.3 Delimitation of Analytical Techniques

These methods are very useful as they help in requirement information elicitation and product development by analyzing the data that is already available. The knowledge of experts and their opinion plays an important role in this approach. Moreover, the information flow has a hierarchy in these techniques; Analytic Methods have numerous advantages as “People” are not the only source of information in terms of requirements. Experts Knowledge and Opinion play an important role in requirement

maturity. Moreover, these techniques have a hierarchical flow of information and are cost effective.

1.10 Thesis Outline

The rest of the thesis has been organized into the following chapters:

Chapter 2

This chapter includes a literature survey related to our topic. An extensive literature study has been mentioned here that gathers all the research aspects of secure requirement elicitation frameworks, methods and techniques to produce effective secure requirement elicitation by different researchers around the world.

Chapter 3

In this chapter, the need of a framework for the Security Requirement Modeling Process has been discussed. This chapter also talks about minimizing software design mistakes in the early stages of the software development process. Moreover, the importance of security requirements has also been discussed in this chapter.

Chapter 4

In this chapter, implementation and validation of the proposed framework is elaborated followed by the execution of an industry-based case study. The objective of this chapter is to discuss different types of methodologies such as implementation through fuzzy AHP, fuzzy ANP along with the comparison of the results obtained through different methods.

Chapter 5

Validation of the framework in the context of the security estimation process is done in this chapter. In addition, the empirical validation, theoretical validation and statistical analysis of the estimated values of

security requirements have been done by using security estimation process.

Chapter 6

This chapter concludes the proposed research work and also gives the future scope of this research work.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

The software design depends on needs of its stakeholders and the environment in which it will be deployed [28, 56]. Effectively defining and removing program specifications is a key issue in the progress of a project. Generally speaking, the cost of fixing a fault after device implementation is costly as compared to rectifying it [28]. Because requirements can change during growth, managing the varying requirements is important. Requirement engineering process acceptance in software development depends on social, technical, and economic issues [32,34,38]. An important issue in Requirement Engineering (RE) is to perform a formal interview for collecting specifications so as to prevent project failures. The need for capturing device without the use of proper technique is the major factor in large software projects failure. Furthermore, poor management of specifications may also be attributed to failures in software projects.

The research indicates that failure or abandonment of a program triggers the major loss in software industries for the aforementioned reasons [61]. Therefore, this research focuses on the collection and analysis of requirement issues between users and software developers, which is a necessary part of requirement engineering. It has been noted that the interviews between software developers and users are considered best practice as part of software development programs in terms of methods for gathering customer requirements [38]. The cycle of productive user interactions is the best requirement engineering exercise. The problem of effective social and organizational participation plays a major role in requirement engineering. One of the challenges is that each

group will view annotations in the light of its own assumptions of context. This is particularly problematic for non-interactive participation, where less opportunity is available to check that the reader has interpreted criteria as expected. Thus the interaction difficulty between user and developer is the major problem in requirement engineering, which can affect the development of the software [31]. The frequency and importance of participation activities characterizes a stage of requirement engineering. During this process various stakeholders will be able to communicate their needs to the analysts.

Security is a problem for designing and developing secure software nowadays. It is also noted that most software developers do not integrate security mechanisms into the Software Requirement Specification (SRS) [32]. For this reason, many of the potential customers are unable to provide software developers with their security related details, through which a timeline and cost of security can be accurately calculated. However, stakeholders are unaware of the security aspects that are important for system security. Therefore, the security and security provision should be specified or included in SRS document [30]. SRS document efficacy also has a consistency impact on the development of software because the software is to be built upon.

2.2 Security Requirement Elicitation

The initial task of Software Requirement engineering is Elicitation of the Security Requirement. Requirement engineering cycle therefore begins from collection of requirements of security, which is a prerequisite for all other major development activities [39]. Security Requirement Elicitation is a crucial Software Requirement Engineering process, because the security requirements obtained at this point will determine the performance of the software product developed [59]. The consistency of the collected security requirements forms the basis of the software

product and thus defines the software project's performance. Active Security Requirement Elicitation greatly increases the consistency of security requirements resulting in the production of a product with quality software [52]. For a good Security Requirement Engineering process, therefore, it is imperative to properly understand and apply the elicitation techniques [60].

The researchers suggest that "Security Requirement Elicitation is the method of gathering appropriate security specifications from end users which is an initial and primary aim of any and all processes in software engineering"[55]. As compared to other Security Requirement Engineering practices, the term "requirement elicitation" is relatively new. "Security requirement Elicitation is still commonly referred to as the selection, capture, acquisition, identification, identification, invention, creation, discovery and fact-finding of security requirements." Whatever the reason, they were unable to represent real operations, it is now well known and agreed that security requirements are elicited instead of just being caught or collected [56]. "Security Requirement Elicitation is the process of having users, consumers and other stakeholders to obtain the security requirements of a program. The activity is sometimes also referred to as the compilation of security requirements" [46].

Hickey describe Security Requirement Elicitation as reading, uncovering, collecting, surfacing and/or finding client, consumer and other potential stakeholders needs [40]. Another concept of Security Requirement Elicitation is the method of defining software or system security requirements from diverse sources through interviews, seminars, workflow and task review, document analysis, and other mechanisms [57].

The concept of Security Requirement Elicitation is further explained as “Security Requirement Elicitation is about learning and understanding

the needs, preferences and expectations of users and customers, with the ultimate goal of communicating them to all stakeholders and, in particular, system developers".

Security Requirement Elicitation is committed to the identification, detection, acquisition and production of stakeholder needs. Security Requirement Elicitation refers to this method as "security trawling" [58]. To highlight the fact that more quality security requirements are gathered through effective Security Requirement Elicitation mechanism [47]. Security requirements are pursued that the appropriate project requirement is gathered. As stated, Security Requirement related information gathering is related to the other Security Requirement Engineering activities, and in particular the review phase.

2.3 Existing Models of Security Requirement Elicitation

As mentioned above, the number of software requirement engineering models built in different contexts over the years are present. Between them, they have differences although the core tasks remain the same. We also provide versatility in addressing the contextual variations between each group [58,61,62,63,64].

An elicitor that face a number of problems that may be compounded by the number of tools and techniques that complicate the job. The software requirement elicitation tasks are also performed in gradual fashion and not in conjunction with other software development life cycle(SDLC) operations [48]. To identify the problem of current process models, the core components of the security requirement elicitation process need to be defined to support the development of more versatile models claimed that the security requirement elicitation consists of knowing (i) the problem domain to be solved, (ii) the organization's business domain, (iii) method of running a system, (iv) the application

domain of the software system. The author notes that a good security requirement elicitation mechanism will include (i) object setting, (ii) context information acquisition (iii) intelligence organization (iv) security requirements collection of stakeholders [33].

A study emphasizes that the security requirement elicitation gathering plan should include Security Requirement gathering priorities, Security requirement elicitation strategies and processes, Security requirement elicitation Product activities, schedule and resource forecasts, and developmental elicitation risk. Although they have addressed the basis of a Security requirement elicitation process, they are providing names of areas where study is to be done [37]. There are variety of models that have been developed to tackle this situation to provide more precise and efficient guidance for the requirement elicitation process. In this article the author suggested a single Security Requirement System Elicitation model based on mathematical principles. This allows researchers to gain organized and full process knowledge. A step-by-step approach was proposed which is based on the quality characteristics of ISO 9126 as a guide for the elicitor to gather individual, social and organizational factors that can improve the quality of the software project produced [35]. Security requirement elicitation is still at a high level of abstraction that is able to provide researchers with more detailed guidance for conducting Security requirement elicitation processes. The explanation behind this discrepancy is the Security Requirement System Elicitation guidelines are denoted by different researchers, and various variables and components.

One of those more significant is the type of SDLC process model being implemented. The type of selected SDLC model, such as the Waterfall, Spiral, Evolutionary, or Incremental models, each has its own respective SRE model which in turn affects the specific Security requirement elicitation mechanism [48]. Descriptive investigation of

specific and related issues of typical SRE process is required for getting the idea of Security requirement elicitation process. To summarize the Security Requirement gathering process, a structure is required for effective Security requirement elicitation process that assists the elicitor in understanding and selecting methods for the Security requirement elicitation process, also addresses core requirement elicitation problem areas. A structure for successful elicitation of the security requirements is suggested in this study.

2.4 Security Requirement Elicitation Methods

For getting requirements related to security from stakeholders and concerned organizations, an elicitor uses the Security requirement elicitation technique. More commonly, a technique is a procedure that does something or a functional approach that is applied to a particular task [42]. This technique will direct both the elicitor and the stakeholder during the elicitation process to avoid the blank slate syndrome that occurs when requesting information on the elicitation of requirements [33]. There are a variety of techniques from different sources that can be used for Security requirement elicitation process in real life scenario. A study has listed only a handful of conventional elicitation methods such as observation, interview [40]. Recently, the Security requirement elicitation surveys analyzed additional and up-to-date strategies based on goals, situations, perspectives and awareness of domains [36].

Some of those elicitation strategies that are commonly used are listed in this chapter to assess. Although this list is by no means complete or exhaustive, it is assumed to reflect the range of available techniques mentioned in the literature and are in use at present [43]. Techniques of Security requirement elicitation are divided into different categories that enable the elicitor to understand their function and use. The definition

includes key characteristics of each technique to keep it simple to understand [44].

2.4.1 Interviews

This is a tested and one of the most common security requirement elicitation methods. Interviews involve taking information and opinions from humans and give the very relevant information. Interviews can quickly efficiently collect information from stakeholders. The quality of the data collected from interviews, such as the usefulness of the information, can vary depending on the skill of the interviewer or elicitor [45]. Interviews are divided into three types: unstructured, structured, and semi-structured. Unstructured interviews are conversational in nature where the interviewers have limited control over the direction of interactions. In this method, there is a risk of neglecting some topics as there are no fixed questions to be asked. Moreover the unstructured interviews processes focus more on specific areas, while neglecting others [30]. The dynamic nature of unstructured interview requires a significantly skilled elicitor for good data collection whereas structured interviews are conducted using a predetermined set of questions. The success of structured interviews depends on identifying the correct questions to be asked, when they should be asked, and who should answer them on given information structured interviews for Security requirement elicitation can be utilized to assist this technique. It requires limited training, less time and a novice analyst can perform evaluations [34].

The ability to elicit effective information, probe, and follow-up, interviews are generally considered to be good for discovering opinions, feelings, goals, attitudes and beliefs in various issues and problems. However, the interviews in Security requirement elicitation process are not cost effective and take time [43].

2.4.2 Surveys

The survey is used to collect requirements throughout the target people. The method takes into account the entire geographic areas for accumulating requirements. It is used simultaneously to gather data from multiple users and if it is of effective design the data is processed easily [35].

2.4.3 Questionnaires

Questionnaires are mainly used during the early stages of Security Requirement Engineering and may consist of open-ended and/or closed questions [54]. Questionnaires need to be effective in terms, concepts, and boundaries of the domain and should be well established and understood by the participants and the questionnaire designer.

Questions should be able to avoid collecting unnecessary data and give effective and a quick way to gather large volume of information. However, they are limited in the depth of knowledge they are eliciting, and in general provide little supporting contextual information. Unlike interviews, questionnaires lack interactivity, and consequently do not provide the opportunity to explore further on a new topic or expand on fresh ideas. Also, they do not provide means for participants to clear their doubts [37].

2.4.4 Group Work

Group Work Specific stakeholders are encouraged to conduct group meetings in cooperation to evoke program requirements [34]. It is a technique that is mostly used but this needs a lot of practice to be achieved. This strategy is very useful in resolving consumer disputes so as to get them to one table. Each and every element of specifications is addressed and uses group work to provide relevant suggestions. The stakeholders provide the direct comments about the program

specifications during group work. Compared to other requirements engineering techniques, it has some drawback such as, a lot of effort is required. Both stakeholders can sometimes participate at the same time as participants can be occupied with other tasks [33].

2.4.5 Brainstorming

Brainstorming is a strategy in which different groups of stakeholders take part in talk session for finding different views and information [33] and then those ideas and views are organized and studied. All stakeholders present should participate actively and be innovative with equal value, and all ideas produced should be registered, no matter how impractical they may seem. Brainstorming sessions are not typically intended for making crucial decisions [37].

2.4.6 Joint Application Development (JAD)

It is a hybrid methodology for analyzing a business and caters to an issue that is of interest to different stakeholders. JAD offers the fast decision-making process regarding the problem and its solution. It meets the demands of rapid changeability. It also offers well-formatted structured approach to gathering requirements. Additionally, it provides direct coordination between all project stakeholders [34,42]. Because JAD offers a quick solution, it is sometimes unable to validate exhaustively in a minimum duration of time.

2.4.7 Requirement Workshops

Requirement workshop is a series of various types of meetings held by stakeholders to elicit project requirements [51].

2.4.8 Observation

Observation is one of the most widely used and traditional technique used as contextual technique [54]. Observation requires good ability to

monitor situations and understand what is happening. Observation requires long duration so as to get the idea of what is being done.

2.4.9 Protocol Analysis

This is the type of conference, in which people noisily address customer needs. Review of the protocol allows for active participation of all stakeholders [54].

2.4.10 Prototyping

A useful method of collecting complete data [54] is to provide stakeholders with prototypes of the program to understand the problem and to provide support for investigating possible solutions. Prototype along with group work and interviews is standard technique of elicitation.

For system prototyping various methods are there. In many instances prototypes this approach is costly in terms of time and cost to manufacture. A benefit of using this strategy, however, is that it allows stakeholders, and more specifically consumers, to play an active role in the creation of specifications [75].

2.4.11 Group Meeting

Brainstorming in a more structured form can be described as Group Meeting [55]. Each participant share the views and views are accepted by the moderator till every view is discussed. The group then works through each idea in sequence to clarify its details, ask questions, and offer comments, thereby creating a shared understanding for each idea. The discussed ideas are then voted upon anonymously as to their importance and/or relevance to the problem. Until a general decision is achieved, the steps of discussion and voting may continue to take place as necessary several times [54]. It helps balancing the effect of every participant especially in a politically or socially sensitive situation [62]. Group

Meeting is the technique in which participants look for group solution as a process of problem solving rather than by negotiation.

2.5 Security Requirement Elicitation Technique Selection

The preceding sections of the literature review identify that each of the elicitation technique has both advantages and limitations [44]. The Security requirement elicitation methods show description and limits of many techniques of information collection for security requirement. The methods of collecting requirements tend to fall into two categories: those that yield rich results but are costly and those that are less expensive but also less informative [51]." Varieties of requirement gathering methods are there and so wide range of techniques, alternative techniques can be used in certain cases, allowing for greater process versatility and more options for the elicitor and stakeholders. It can also be seen that most of these approaches do not come from the traditional Software Engineering fields. Requirement elicitation methods are mostly derived from social sciences, organizational theory, group dynamics, information engineering and very often from practical experience. Many methods are therefore useful in producing knowledge related to domain area [53].

Almost all the strategies for eliciting criteria are not formal and include human-to-human contact. Of all the strategies, group work is particularly effective because it would seem that groups are better at handling complex tasks such as requirement elicitation process than individuals because they have a broader range of skills and abilities to draw from. Group work strategies are also useful for gathering requirements as they involve participants, engaging clients, encouraging dialogue, teamwork, creating ideas, finding solutions, and making decisions. The advantage of Requirements Workshops is that they integrate with other elicitation techniques [43, 53]. In addition, these group strategies are naturally very useful for the collection of

requirements as software development is essentially a community effort [51]. Debate is response of organized seminars and personal interactions. Latest works show that one-on-one interviews were found to be more effective for small projects [53], while facilitated workshops were more efficient for larger projects in accordance with the dynamic system development method (DSDM) [52].

2.6 Existing approaches on Multi Criteria Decision Making Techniques

Multi-Criteria Decision Making (MCDM) method is useful to deal with the uncertainties and ambiguity of human judgment [65]. The software can be made more secure and reliable by finding the priorities of the security attributes based on their weightage and ranking. Multi Criteria Decision Making process is useful for prioritization of security attributes [67]. The results obtained through this strategy can be used to enhance the security of software and applied into the prioritized elicitation methods. This method is appropriate for software industries because every organization have its own rules to develop the software. As such, the prioritization and ranking of the attributes of security has not been attempted. Further, security testing is the key to develop secure software system so it becomes necessary to find the priority of security attributes [68]. MCDM method is appropriate to select the importance of security attributes and help during the selection of elicitation techniques [69]. The various techniques of MCDM that helps to find out the weightage of attributes, their rankings and their respective relationship among each other are discussed in detail:

Mohd. Sadiq et al. discussed about using Fuzzy Analytical Hierarchy Process (F-AHP) and Fuzzy-TOPSIS method for selecting requirements for software. In this research paper, they are discussing about two types of software requirements functional and non-functional in which numbers

of factors are effected with respect to requirements. In this paper, MCDM is included in determining relations between sub-criteria and criteria result [66].

Author proposed the following steps: (1) ISM- applied in determining relationship between sub-criteria and criteria, result is utilized construct decision making network. (2) VIKOR is used for determining ranks of location of solar plants finally; this work concludes that the use of hybrid methods for better results is advised in spite of a single MCDM method [172].

The author discusses the use of ANP method and informs that this method supports dependencies and feedback elements in the network. Further the author explores this by structuring the problem as a weighted graph and uses the concept of compatibility between inter-dependent matrices in the ANP. Since, ANP is a time consuming activity and challenging activity the author proposed two upgrades of how to automatize some parts of the ANP to be less complex and more appropriate for users.

In this paper, the author stated that actual situation of decision making is rather dynamic process and not static. In this work Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) is used for the work selection of CNC Lathe and further weights are assigned by entropy method. The best selection is made after the ranking of alternatives and finally proposed selection methodology is justified by MATLAB programming [71].

The author explained Analytic Network Process (ANP) method as a multi-criteria theory of measurement used to derive relative priority scales of absolute number from individual judgments. The ANP is a

special case of AHP with its independent parameters based on decisions [72]

The author explained that DEMATEL method is a methodology that can be used for researching and solving complicated and intertwined problem groups. DEMATEL process visual representation as end product in form of impact-relations map by which, respondents can organize their own actions. DEMATEL method uses MCDM to construct interrelations between the criteria [75]. The author proposed a method known as Maximum Mean De-Entropy (MMDE) algorithm to find out interrelationships between various criteria for evaluating effects in a real life example of E-Learning programs.

From the study of above mentioned literature review, we observed that Multi-criteria decision making (MCDM) is one of the best techniques to resolve the problems pertaining to uncertainty of the selection of security attributes to boost the security of software. The hybridization of fuzzy technique with multi criteria decision making approach can enhanced the overall security apparatus and thus will improve the security of software in a much stronger way [69].

2.7 Prioritization of Requirement Elicitation Techniques

A number of requirement elicitation techniques are available that can be used by the elicitor. However, there is a need to prioritize the requirement elicitation technique that can meet the specific needs of every stakeholder. To achieve this task, we have to follow criteria through which prioritized rankings have been established [68]. It is therefore important to prioritize software specifications to achieve customer satisfaction for implementation [64]. Because prioritizations include a small subset of stakeholders, the results are skewed towards those participating in the process.

2.8 Conclusion

One of the most critical and important activity of software development is Security requirement elicitation, poor execution of elicitation can lead to complete failure of the final project. The efficiency of software depends on security requirements and how much the product meets the requirements. Improper security needs leads to faulty software. Good quality security play pivotal role in project success. From the above literature survey it is concluded Security requirement elicitation process is one of the most important and critical activity of requirement engineering. It can therefore be said that Security requirement elicitation process requires an extensive skill set combined with experience to be performed well. A framework is needed that not only identifies the core problems area of elicitation but also suggest the guidelines and the solution to these core problems areas. To achieve these goals an efficient Security requirement elicitation framework is proposed in this research study that gives suggestive solution to these problems.

CHAPTER 3

FRAMEWORK FOR SECURITY REQUIREMENT ELICITATION PROCESS

3.1 Introduction

Requirement elicitation is the process of conformance of stakeholder's actual requirements. There is a various requirement elicitation techniques are available in the literature, but most of the framework do not represents the complete elicitation approach into the requirement engineering process. This process, ensure that the software conforms to its specification and that the customer ultimately receive what they ordered.

The quality of software systems depends on early activities in the software development process, of which the requirement elicitation is one. When requirements are not managed properly, a project can fail or become more costly than intended, and the quality of the software developed can decrease [122]. At the software development process, it is particularly important to identify all problems at the time of development phase by using techniques of requirement elicitation and then resolve the entire problem step by step.

For that, security requirement elicitation techniques play a crucial role in order to construct a high quality software system. Therefore we have presented the novel approach for requirement elicitation process into the requirement engineering process [148]. The proposed model for elicitation of requirements deals with problems that can lead to failed software project. This is done by incorporating stakeholder requirement early in the process of software development. In this research study, an

efficient security requirement elicitation framework is proposed which addresses the core problems of elicitors.

3.2 Security Attributes

Security Elicitation is a multi-dimensional and comprehensive process that involves a large gamut of operations divided into several stages to ensure in-debt analysis of security related challenges and threats and ways to mitigate the problems that could affect the operations of a software system. The seven set attributes are integrity, confidentiality, authentication, effectiveness, availability, access control and authorization. These seven attributes are the basic fundamentals of security without which security of software cannot be ensured. The main reason of using these attributes is to plug in gaps in the software structure so that software security breach could not be made [139].

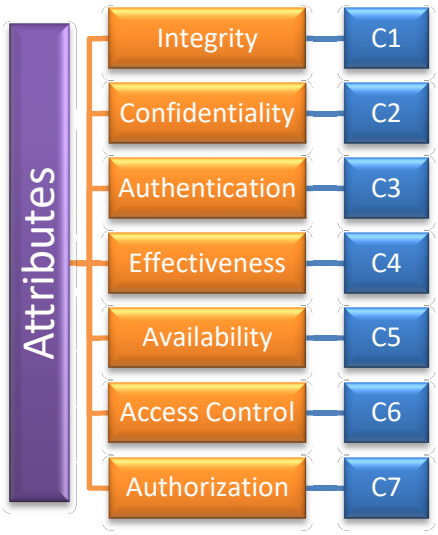

Fig. 3.1 Security Attributes

3.2.1 Integrity

Integrity is the accuracy of the data at storage or during transmission. It assures that the data received by the end user does not get corrupted or tampered during the transmission can be defined as the accurateness of

data/information at storage or during transmission. In a more expanded form integrity can be insured both at the source and destination which can prevent the unauthorized use of data [140].

3.2.2 Confidentiality

Confidentiality insures that the data is not disclosed to any unauthorized user. Inability to maintain confidentiality can lead to data breach and the leak sensitive data to unauthorized persons. Confidentiality should be taken care of in the security requirement elicitation phase so as to develop a secure software design [144].

3.2.3 Authentication

Authentication is the process of identifying the legitimate user requesting access to the system. A user name and password is the most common method of authenticating any user to provide the access. The process of authentication involves mechanism which validates authentic users or multiple users to access information, with the help generated valid tokens corresponding to each user. This authentication can be in a form of an OTP (One Time Password) or SMS, biometric authentication, secret question, token based authentication like RSA secure ID token etc. [93].

3.2.4 Effectiveness

Effectiveness is the ability to produce the expected outcome or expected results meeting the user requirements. An ineffective system will lead to undesirable output [104].

3.2.5 Availability

Availability attribute insures that a system is ready and available for use by an authorized user whenever needed. The availability of a system may be compromised in case of a denial of service attacks [104]. The

availability of a system conforms that the system is ready to be used to all the needed functionalities. The system should be designed in multiple sub systems so that the availability of the system is not jeopardized in case of the failure of any of the sub systems.

3.2.6 Access Control

Access control limits the way the system should be used by its legitimate users. The users are required to present credentials in order to access the specific functionality of the system. The users are decided into levels based on their access controls. Some of the users may be given full control of the system like the administrators while other users may be given only limited access like the end users based on their specific use of the system [138].

3.2.7 Authorization

Authorization is the process which authenticates a user, a computer, a network device or an authenticated principle to specify access rights to perform certain actions such as perform checks, making changes in the database, or download information etc. Since it acts as a gateway to further access the system, it plays a key role in preventing malicious attacks, security breach or theft [130]. The system can only be accessed by the authorized/appointed user, and only the authorized user can perform certain specific operations within a trust domain.

3.3 Quality Requirements

Quality requirements are often ignored in software development. Few requirements are non-functional but they give the functional details of a system.

As per general conventions there is some sort of mission including systems that are sensitive. System developers have understood the value of requirements responsible for software development [82]. However

many systems completely ignore the quality requirements or behaviour in inadequate manner. This may lead to failure of power systems of software's associated with unmanned spacecraft etc.So quality requirement is of paramount importance and should be considered in software development [83].

3.4 Requirement Elicitation Issues

The process of elicitation of requirements may seem easy but it is typical in a way that many questions need to be asked from users, stakeholders etc [87]. There are different requirement elicitations problem in requirement engineering:

3.4.1 Problem of Scope

If the limit of software usage is not defined, it will lead to inefficient or failed software. So it is important to take care of the context of software. Maintaining a strategic distance from relevant issues can prompt necessities which are fragmented, not verifiable, pointless and unusable. This issue emerges when the limit of software (that is, degree) isn't characterized appropriately.

3.4.2 Problem of Understanding

Clients have incomplete knowledge of their needs and the problem is not defined clearly. It may lead to the end product that may not fulfil the demands of stakeholders. Users have incomplete understanding of their needs, capabilities and limitations of system and poor knowledge of problem domain [142]. Problems of understanding during elicitation can lead to requirements which are ambiguous, incomplete, inconsistent and even incorrect because they do not address the stakeholder's true needs.

3.4.3 Dynamism in Requirements

It is important for the software to be flexible so that changes can be made as per changing requirements. During the time it takes to build up a framework, the client's needs may settled. If such changes are not accommodated, the original requirements set will become incomplete, unreliable with the new condition and possibly broken because they capture information that has been since become outdated.

3.4.4 Knowledge is Tactic

It means the knowledge or information that is difficult to understand. We think of knowledge as something that can be verified in words, visualized and educated. However, this isn't always the case. Tacit knowledge is a class of knowledge that's difficult to communicate.

3.4.5 Requirement Change

They change with time so one should not allow set of requirements that is irrelevant. Requirements change over some interval. The requirements elicitation process itself is an understanding experience for users and ideas discussed at one point may cause them to change their minds about prior decisions. We must be alert to avoid taking a set of requirements that is obsolete by the time the elicitation process is completed [135].

3.4.6 Limited Domain Knowledge

Generally issue with the requirement elicitation process is that people are oblivious of the exact requirements, so it depends on expertise of a software developer to document the requirement properly. The figure 3.2 has shown the various domain of Requirement elicitation process which are directly affect to problem domains.

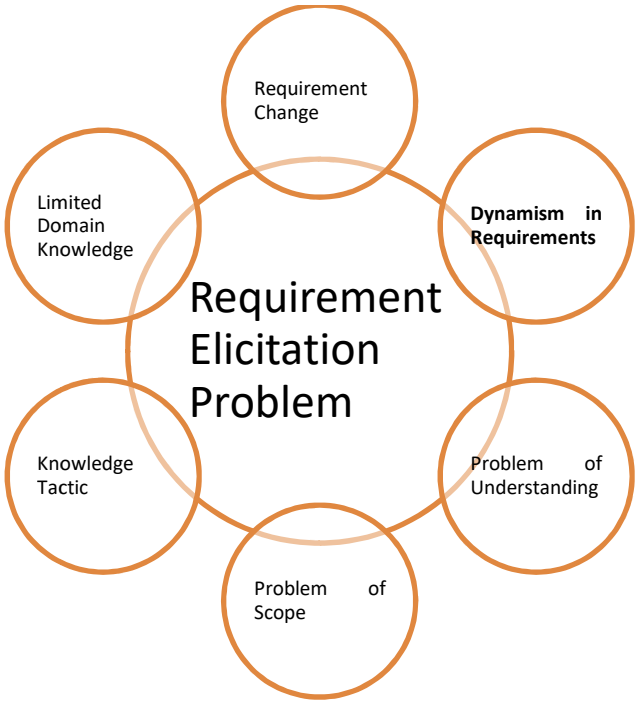

Fig 3.2 Requirement Elicitation Process

3.5 Problem Solving Framework

Everybody can be benefitted from having a good problem solving skills as these problems are encountered on daily basis. Some of these requirement elicitation problems which are discussed above are more complex for solving. It will be wonderful to have the ability to solve all requirement elicitation problems efficiently and timely without any difficulty [136].

In this framework, the process of solution of the problem is described. At initial stage all the requirements are collected from stakeholder and then focus on the objective of the organization and how goals can be achieved is documented. After this process, the solution of the required problem is achieved by the developer. S/w Engineer resolves all these problems with software, hardware and training. After completing the process it reaches to the solution domain process. Finally, the problem

(fig 3.3) may be resolved and the solution comes out of the above process implements in the SDLC.

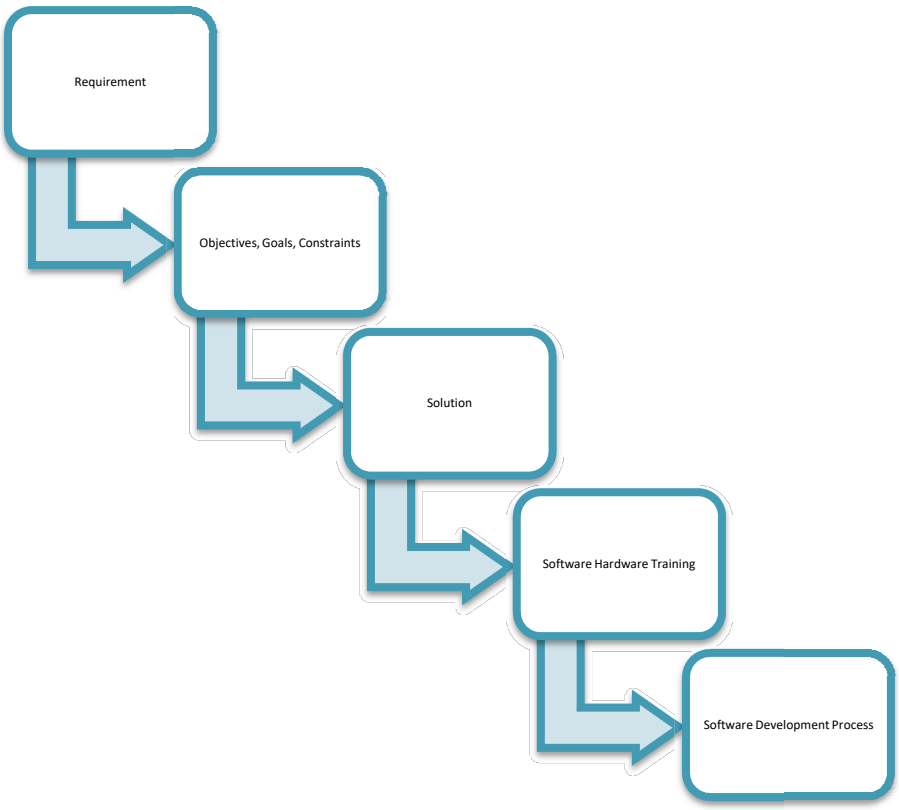

Fig 3.3 Problem Solving Framework

3.6 The Framework

A specific framework of methodology is required for elicitation of requirements. There are different requirement elicitation processes in requirement engineering. This framework (figure 3.3) consists of several phases based on finding of literature review and critical observation done in the previous chapters.

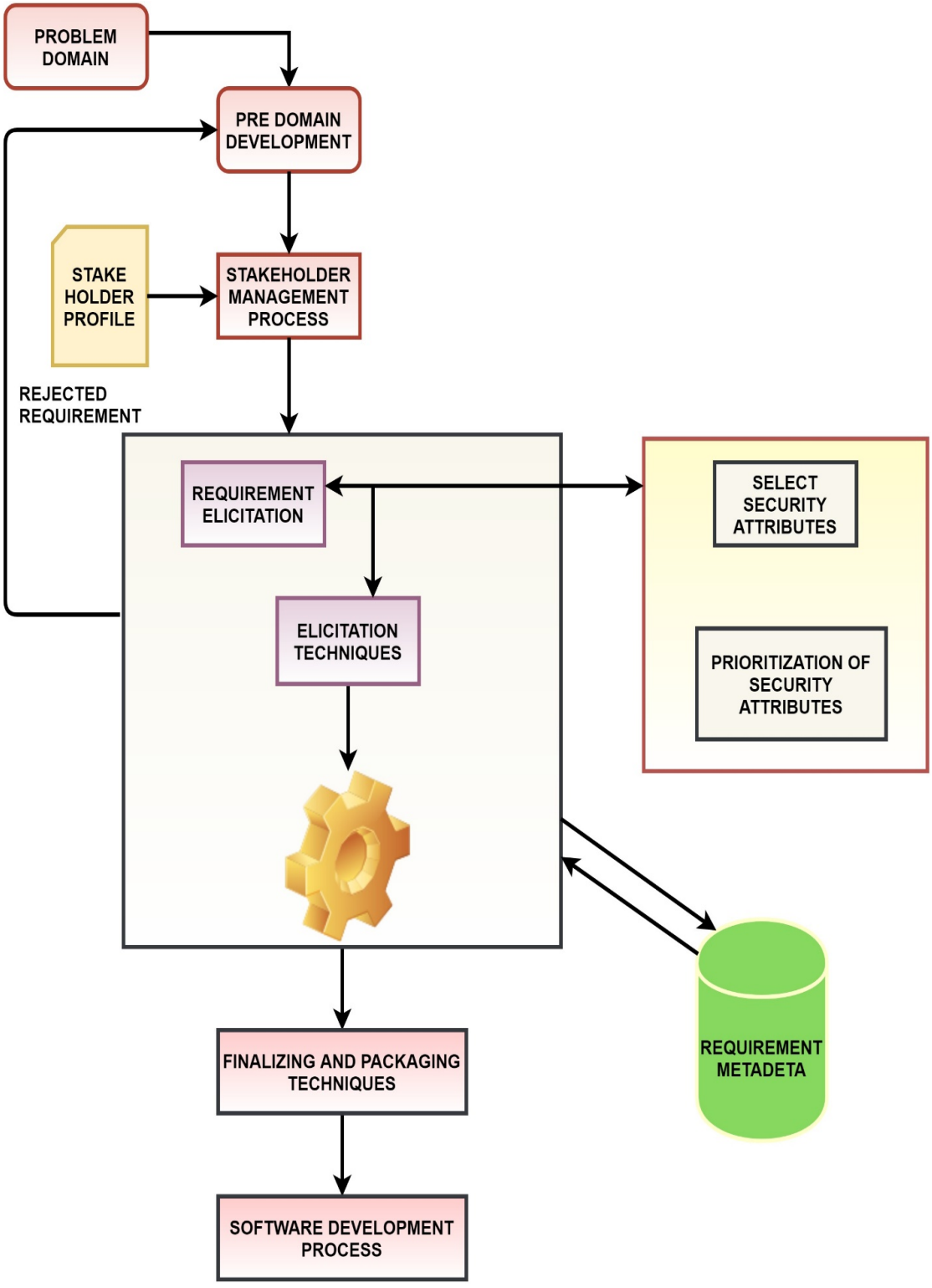

Fig 3.4 The Framework

Proposed Algorithm

Step 1 : Start.

Step 2 : Identify the Problem Domain.

Step 3 : Pre Domain Development.

Step 4 : Use Stakeholder profile to define Stakeholder Management Process.

Step 5 : Select and prioritize the Security Attribute along with each requirement elicitation technique.

- Required elicitation technique.
- Use Requirement Metadata repository.
- Rejected requirement goes back to STEP 3.

Step 6 : Use Finalizing and packaging Techniques to facilitate the Software Development process.

Step 7 : End.

3.6.1 Phases of Framework

This framework has 9 stages, which are explained here:

3.6.1.1 Problem Domain

It is important to identify domain of the problem for properly understanding the requirements. This solely depends on domain expert. Figure 3.1 provides examples of Requirement Elicitation Problems, such mappings and correlation. Requirements applicability questions are organized in a hierarchical fashion, with high-level questions selecting a large set of requirements that is successively pruned using specific questions that are related to fewer requirements. To systematically support such selection, the characteristics and constraints of security requirements should be captured as attributes in the SRE based on related decision-making activities in the problem domain.

3.6.1.2 Pre Domain Development Process

Users generally have little knowledge of their needs and this can lead to improper elicitation of information and wrong list of requirements. Domain analysis is an activity occurring prior to system analysis. It aims to identify features common to a domain of applications, selecting and abstracting the objects and operations that characterize those features. The generalization of the systems in an application domain aims to define domain models that transcend specific applications. Generally speaking, domain analysis should support extraction, organization, analysis and abstraction, understanding, representation and modelling of reusable information and assets from the Requirement Elicitation Process.

3.6.1.3 Stakeholder Management Process

The work involves the development of appropriate strategies to execute the work with the identified stakeholders in stakeholder management, analysing their requirements and impacts, stakeholders and process. Stakeholders' needs and expectations should be considered. Managing conflicting interests in project decisions and activities and to involve stakeholders is also important [90]. It is a part of all stakeholder management process. You are supposed to project manager to manage the ability to identify the needs of stakeholders and influence them effectively. A viewpoints hierarchy that captures different perspectives and related stakeholders of a security requirement. Documentation is provided to stakeholder related to operational security measures. Tasks defined above must be communicated to different researchers. To reach good results during the security specification, the requirements analyst needs to spend special attention with the Stakeholders. Analyst shall consider they have not enough security experience and so, there is a big chance to security be the last thinking. To avoid it, elaborating a questionnaire is a good approach.

3.6.1.4 Elicitation of Needs or Requirement

By analysing these articles we understood different security requirement elicitation techniques and their benefits. Secondly we analyse the application of combined requirements engineering techniques for efficient and successful requirements engineering process for real life complex project. We cannot say that one technique is better than the others [91]. There are number of security requirement elicitation techniques which facilitates in detecting different security vulnerabilities. There are number of security requirement elicitation techniques which facilitates in detecting different security vulnerabilities. Security requirement elicitation techniques are divided into two categories i.e. formal and informal.

3.6.1.5 Techniques of Elicitation

It refers to gathering of information for understanding requirements of software. Improper elicitation leads to failure of project. Formal security requirement elicitation techniques follow a systematic procedure and definite number of steps. Besides following the standard steps, there exists a definite order to follow these steps. Lack of flexibility and limitation in defining scope of (formal) security requirement elicitation techniques limit the use and efficiency of technique [93]. Due to limitations in scalability of the method, formal techniques are hard to use for big projects. There are predefined roles and repetition of process in formal techniques to conduct the requirement specification. On the other hand, informal techniques are more flexible and it becomes easy to define the steps according to the nature of the system under consideration. Despite the flexibility of informal techniques, they lack a systematic approach [136]. Repetition of the process in informal techniques is not easy as compared to formal methods. Since there are no pre-defined steps in informal security requirement elicitation techniques, most of the software professionals avoid using them. Informal security requirements

elicitation techniques lack a proper structure and normally leave few security problems un-addressed. In addition, particular roles are not defined for the stakeholders in order to conduct the specification [141]. These are some of the main reasons that informal techniques are normally avoided by the security requirement engineers. The individual elicitation techniques are described in detail in chapter 2.

3.6.1.6 Select Security Attributes

It refers to setting of priorities of security attributes during elicitation. Lack of techniques for setting priorities of security attributes effect on time release of software.

3.6.1.7 Prioritization of Security Attributes

For the prioritization of the various security attributes, we have applied the traditional AHP and the fuzzy-AHP technique. To confirm the accuracy of the results, both the techniques had been applied on the same dataset.

3.6.1.8 Finalization and Packaging Techniques

It refers to finalizing the designing part and handling for production. Packaging techniques are used to organize the design in such a way that each part is easily understood and produced as it is. The designer's last responsibility will likely be to attend the first press check. A press check is where the designer meets with the printer on-site to review the specifications (specs) of the job and to oversee the first print run in order to establish quality standards or ensure that the print run meets previously established quality standards.

3.6.1.9 Software Development Process

It’s the onset of development process and it involves documenting data of previous phases. The software goes through a brief test run so as

to see if it is working properly. It's a common practice among companies providing custom software development to disregard security issues at the early phases of software development lifecycle (SDLC). With such an approach, every succeeding phase inherits vulnerabilities of the previous one, and the final product cumulates multiple security breaches.

3.6.2 Limitations

Limitations that are considered during the implementation of framework are as follows:

- The requirement gathering process is a human centric process. Thus, it needs to include the behavior or skills of the people that directly or indirectly involved with the software project.
- The software project is influenced by several parameters. Therefore, it is important to include the project attributes in our study.
- The process models involved in the development of software needs some set of elicitation techniques according to the behavior. So, we have also included the process models as an important parameter for the study.

3.7 Discussion

We have observed that the available requirement elicitation models do not provides the complete futures of the security requirement elicitation techniques. From the detailed study of literature review, it can b seen that so far no researcher has cognate security attributes with to their respective weightage and ranking the propounded framework given in this chapter has come up with the idea of associating security attributes with requirement elicitation techniques and further ranking them in accordance to their applied usability in a certain case, which will help to save a considerable amount of product cost and time in the development of a software application. Further, this framework is used by those

organizations that develop the quality software which completely meets out with the user's requirement.

3.8 Summary

Problems related to requirements elicitation occur while working on large software products. So, special preparation is required in this case and it is important to choose the most appropriate elicitation technique.

Requirement Elicitation is a critical and important activity of requirement engineering. According to the various surveys studied in the thesis poor requirement engineering leads to project failure and requirement elicitation is one of the major factors. The proposed framework in the research work provides support for effective requirement elicitation. Requirement elicitation framework solves basic issue encountered by analysis during information gathering. The proposed framework supports both types of requirement, functional and non-functional. There is no distraction among functional and non-functional requirement in the proposed requirements. Requirement elicitation framework is categorized into three stages, Pre-elicitation, Elicitation and Post-elicitation. Pre-Domain development and Stakeholders Management are the pre-elicitation activities. In this step the domain is thoroughly studied, analysed, verified and validated to understand the problem domain. If the problem domain is analysed properly, the analysts can easily identify the functional and non-functional requirements related to the problem domain. In stakeholder's management, the key stakeholders are identified for collecting requirement from stakeholders during elicitation phase.

In Requirement Elicitation selection of elicitation technique is based on ANP technique. In Post – Elicitation stage both functional and non-functional requirements are stored in repository for further analysis and

finally the requirements are prioritized as early requirement prioritization using fuzzy system. The activities and processes proposed in the requirement elicitation framework supports both functional and non-functional requirements and effective requirements are elicited.

CHAPTER 4

IMPLEMENTATION OF THE FRAMEWORK

4.1 Introduction

Analytical Hierarchical Process (AHP) is considered to be sufficient in evaluating a group decision, but Saaty found that Analytic Network Process (ANP) is more useful in providing their weighting with succinct decisions [103]. However, it was observed that more reliable relationship has been established in the case of fuzzy ANP, which provides full priority analysis by decision-makers [103] Fuzzy ANP therefore operates with judgmental feedback from a community of decision-makers to build a network of elicitation techniques according to their value or priority. The researchers provide ANP, which is the advanced form of AHP. ANP is well suited for dealing with complexities and uncertainty of human discretion [73].

ANP technique is more sophisticated and developed than the AHP technique, while the fuzzy ANP technique is an improvement over the standard ANP approach by which better results can be obtained. This research provides a platform for analysing security requirement elicitation techniques through the use of quantitative network system and fuzzy ANP methods [143]. We have collected data from various experts from different academic and industry fields. The goal is to determine the safety criteria for elicitation strategies in terms of their weight and levels, according to the various inputs from the experts. Security design approaches were chosen on behalf of these tests to reduce and meet these research characteristics for software's long life and security in the future [147].

Many decision based issues are not organised in hierarchy because they are related to the interaction between elements of lower level and that of higher level. This interaction is structure that is like a network and not hierarchy based [142]. Nonetheless, ANP is crucial in establishing a network system where the criteria on same level are dependent on each other. Another discrepancy in the calculation method between AHP and ANP is the introduction of an enhanced definition "supermatrix" in ANP [103].

ANP is an enhanced AHP interpretation. As per the suggestions of Saaty, AHP can be utilised for solving the matters related to autonomy on criterion or alternatives, and ANP can be used for solving the issue related to dependency on criterion or alternatives [149]. Figure 4.1 outlines the actual difference between AHP and ANP as shown below:

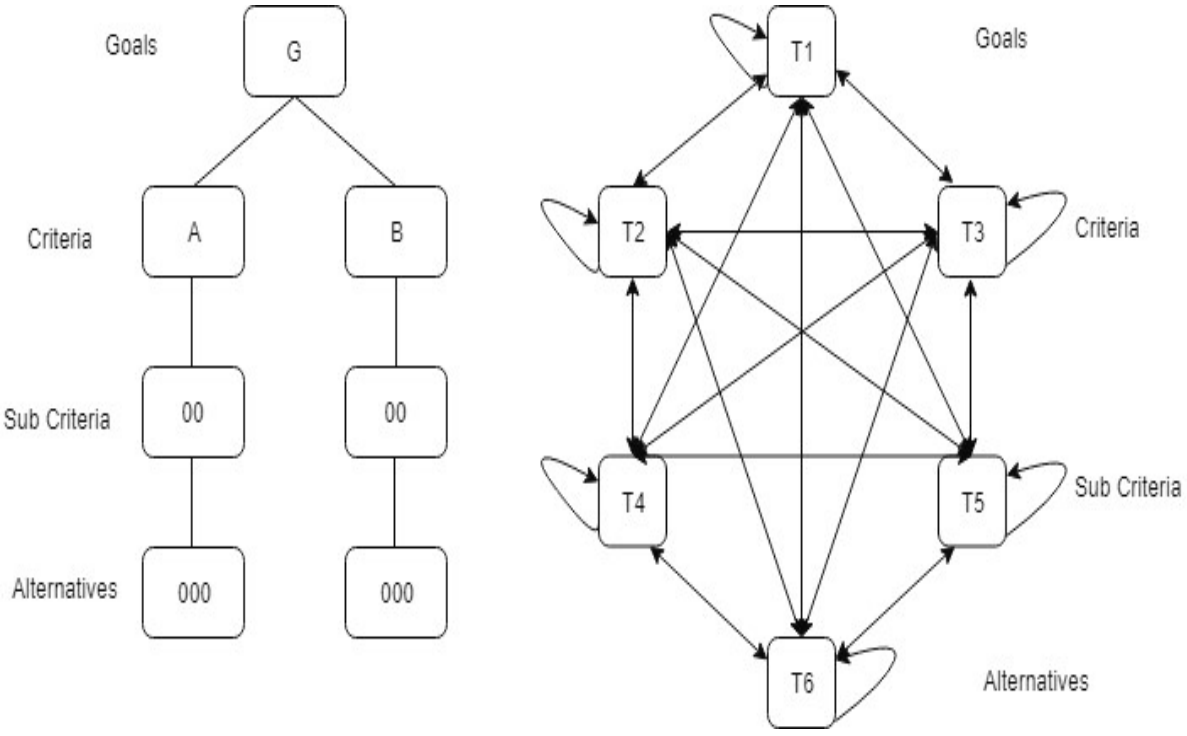

Fig.4.1 Difference between AHP and ANP Structure

4.2 Steps in FANP for the Evaluation of Security Requirement Elicitation Techniques

There are several recent applications and research works available in the literature that gives better results to the ANP system [103]. In 2018, the author proposed a network structure of multitudes of security requirement elicitation techniques, through which a complete relationship and interdependencies among these attributes are realized by using FANP; Yazgan applied Fuzzy ANP to pick the shipping rules about selection of idea and used approach that is based on Fuzzy based ANP [97].Faulty Behaviour Risk (FBR) at workplace by using ANP model based on fuzzy logic [149]. The steps of ANP are given in figure 4.2:

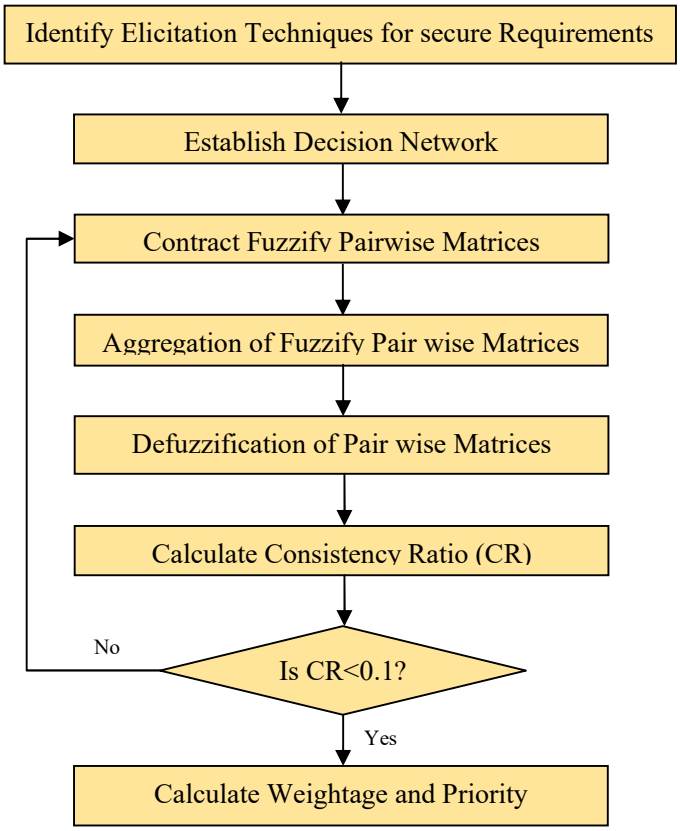

Fig.4.2. Methodology used in Analytic Network Process

At initial stage construction of network structure with criteria, sub-criteria and alternatives are established. Then groups of all items taken

for prioritization and ranking are calculated after these steps. Because ANP is purely based on the system of networking, all the relationship between these clusters is established within things in each cluster. As a consequence, there are few different relationships that have some impact. Nonetheless, the items showing the effect directly may be considered normally dependent in a generic hierarchy whereas the items that have indirect effect on dependency flow by some other criterion. Factors showing self-interaction are also a more dynamic effect and the last one is a reciprocal effect showing interdependencies between parameters [148].

4.2.1 Formation of Pair Wise Comparison Matrices by Analytic Network Process

This process is achieved by the elements within the clusters. The main objective is to create pair wise correlations with control hierarchy on behalf of the criterion or sub-criterion. The effect is obtained on each cluster that is determined by this criterion [107]. As a result, essential weights of all items are calculated and relationships are formed between items by which decision makers compare two items in order to eventually decide the contribution of all factors. Eigenvector are utilized for obtaining the local priority vector which can be achieved through equation 1:

$$PX = \lambda_{enb} X \qquad (1)$$

Where P=Pair wise Comparison Matrix, X=Eigen Vector and λenb=Eigen Value; X can be calculated by using Normalization Algorithm.

The comparisons between attributes or techniques of security requirement elicitation are calculated by the following equations (2) to (7):

$$P = \left[p_{ij}\right]_{n \times n} \; ; \; i = \overline{1,n}; j = \overline{1,n} \qquad (2)$$

$$Q = [q_{ij}]_{n\times 1} \; ; \; i = \overline{1,n} \tag{3}$$

$$q_{ij=}\frac{p_{ij}}{\sum_{i=1}^{n} p_{ij}} \tag{4}$$

$$R = [q_{ij}]_{n\times n} \; ; \; i = \overline{1,n}; j = \overline{1,n} \tag{5}$$

$$X_i = \sum_{j=1}^{n} \frac{R_{ij}}{n} \tag{6}$$

$$X = [X_i]_{n\times 1} \tag{7}$$

4.2.2 Formation of Supermatrix and Limit Supermatrix

In order to achieve global priority in an interdependent structure the appropriate super matrix columns are assigned with vectors having localised priority. The final structure of the obtained supermatrix is identical to step 9 of the Markov chain. Long-term impacts of things are calculated by increasing the super matrix power [104]. In supermatrix each element pretends to be the relationship between two objects in the structure as a whole. The matrix power increases to 2l+1, where l is a number which is large arbitrarily. This is done so as to make equal the primacy of weights. The new matrix established is therefore known as the Super Limited Matrix In addition, to find the consistencies of each element for comparison, the following expressions (8) to (12) are used:

$$S = [p_{ij}]_{n\times n} \times [X_i]_{n\times 1} = [s_i]_{n\times 1} \tag{8}$$

$$T_i = \frac{s_i}{X_i} \; ; \; i = \overline{1,n} \tag{9}$$

$$\lambda = \sum_{i=1}^{n} \frac{T_i}{n} \tag{10}$$

$$\mathrm{CI} = \frac{(\lambda - \mathrm{n})}{(\mathrm{n}-1)} \tag{11}$$

$$\mathrm{CR} = \frac{\mathrm{CI}}{\mathrm{RI}} \tag{12}$$

Where CI: consistency indicator, RI: random indicator and CR: consistency index. Its value less than 0.10 shows consistency of acceptable level [96].

4.2.3 Fuzzy Method

Because of vagueness or imprecision, Zadeh first suggested the fuzzy set theory in 1965 to deal with uncertainty [69]. Fuzzy logic deals with subjective reasoning and tries to solve the problem by assigning values to a wide range of ambiguous and imprecise data spectrum. Mathematical logic is used by assigning values to the incorrect data array in order to achieved at the most reliable conclusion possible [106]. Triangular fuzzy membership function uses three points or parameters where each Triangular Fuzzy Number (TFN) displays a linear representation on both sides left and right:

$$\mu_A(W) = \begin{cases} (W-a)/(b-a), & W \in [a,b] \\ (c-S)/(c-b), & W \in [b,c] \end{cases}$$

For the accompanying arbitrary pair wise comparisons of expert opinions, triangular fuzzy numbers are used in the proposed study. Table 4.1 displays the conversion scale for triangular fuzzy membership functionality. This scale is used to convert values of linguistic types into fuzzy scales embedded in this model. To find the relative weights of techniques of security requirement the author adopted Cheng's method [102].

Table 4.1 Linguistic Scale

Linguistic Scales	Fuzzy Scale	Reciprocal Fuzzy Scale
Equal	(1,1,1)	(1,1,1)
Evenly Important	C_1=(1,1,3)	(1/3,1,1)
Softly important	C_3=(1,3,5)	(1/5,1/3,1)
Important in essence	C_5=(3,5,7)	(1/7,1/5,1/3)
Strongly important	C_7=(5,7,9)	(1/9,1/7,1/5)
Totally important	C_9=(7,9,9)	(1/9,1/9,1/7)
Middle values between two adjacent	C_2, C_4, C_6, C_8	

4.3 Implementation through Fuzzy ANP

Specifically, traditional analytical hierarchy processes and analytical network process techniques do not necessarily support or provide a clear picture for assessing in ambiguous, dynamic and imprecise circumstances [156]. So, by using fuzzy ANP in such situations where decision-makers/stakeholders are unsure about the weight level is necessary to solve the uncertainty issue, making FANP an effective decision-making tool in multi-criteria problems. Fuzzy ANP methodology has been shown to be very effective in solving decision-making problems when there are interdependencies within the networks between different elements. The triangular fuzzy membership function is useful to construct a matrix of comparison between the different elements in pairs [158]. The method discussed in the proposed priority of security requirement elicitation techniques include: Questionnaire(T1); Data Analysis (T2); Interview (T3); Observation (T4); Requirement Workshop (T5); Prototyping (T6); Group Discussion (T7), Brainstorming (T8). Evaluation of Security requirement elicitation techniques are typically a qualitative measure. As a result, it becomes a challenging task to quantitatively determine the characteristics of security requirement elicitation. Weights and ranks of security requirement elicitation techniques play an important role in highly secure software development. A multi-criteria decision-making (MCDM) framework was used for the development of secure requirements to prioritize security requirement elicitation techniques [69, 71]. Figure 4.3 reflects all the possible relationships and interdependence among the security requirement elicitation network diagram.

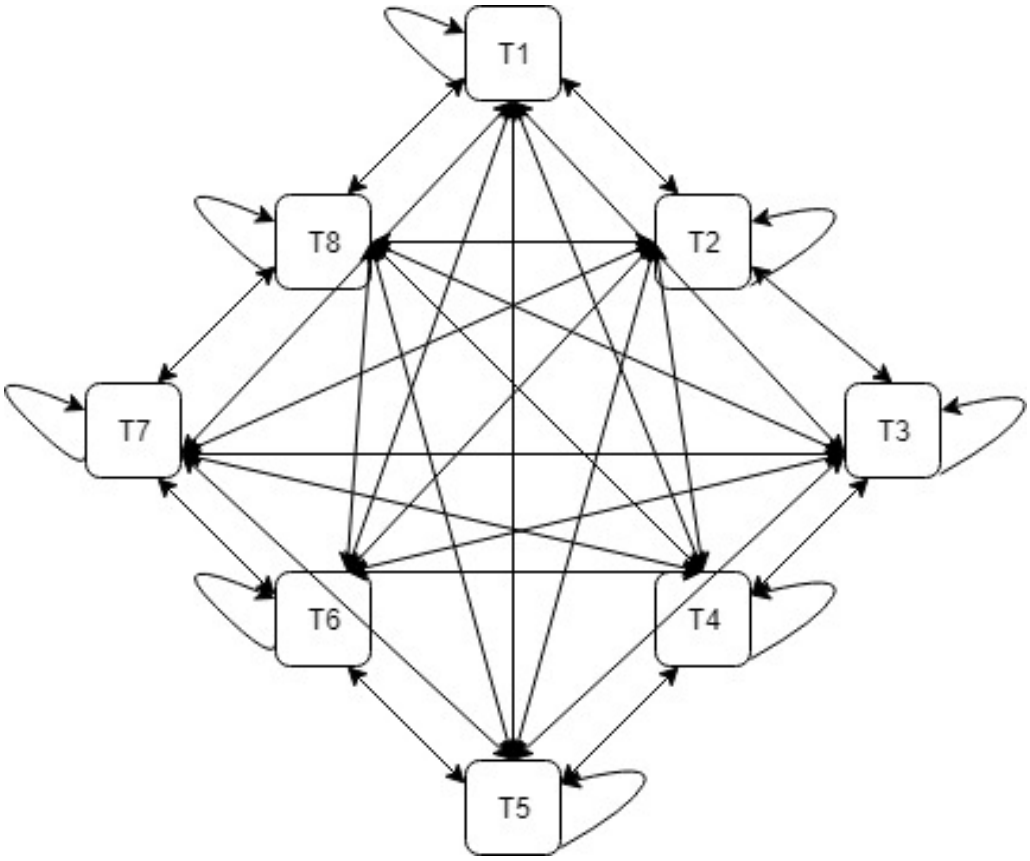

Fig. 4.3 Network Structure Formation in ANP Method

The study aims at using a model of MCDM methods which is hybrid in nature, for attributes of security requirement elicitation. Table 4.2 shows the name of the cluster and its symbols [71]. To define the requirements for security requirement elicitation techniques, extensive literature review has been done along with the involvement of professionals who have defined the criteria in eight classes or clusters as shown in table 4.2. From table 4.3 to table 4.8, alpha-cut method for de-fuzzification of local security requirement elicitation techniques priorities and creation of super matrix from all local priority vectors is shown.

Table 4.2 Description of Security Requirement Elicitation Techniques Cluster		
S. N.	**Cluster Name**	**Cluster Representation**
1	Questionnaire	T1
2	Data Analysis	T2
3	Interview	T3
4	Observation	T4
5	Requirement Workshop	T5
6	Prototyping	T6
7	Group Discussion	T7
8	Brainstorming	T8

Table 4.3 Pair wise comparison matrix								
	Questionnaire (T1)	Data Analysis (T2)	Interview (T3)	Observation (T4)	Requirement Workshop (T5)	Prototyping (T6)	Group Discussion (T7)	Brain Storming (T8)
Questionnaire(T 1)	1.0000, 1.0000, 1.0000	0.6600, 1.1700, 1.6900	0.7000, 0.9500, 1.3500	1.5250, 2.3540, 2.9010	1.6920, 2.4140, 3.1470	1.5490, 2.3540, 2.9010	0.5520, 0.6390, 0.9050	0.4510, 0.8500, 1.0000
Data Analysis (T2)		1.0000, 1.0000, 1.0000	1.1900, 1.5800, 2.1500	0.4450, 0.8500, 1.0000	1.4590, 1.8590, 2.2150	0.4510, 0.8500, 1.0000	1.5530, 2.2000, 2.8500	1.0850, 1.5000, 1.5420
Interview (T3)			1.0000, 1.0000, 1.0000	1.0850, 1.5000, 1.5420	1.6050, 2.3360, 3.1470	1.0850, 1.5000, 1.5420	1.5490, 2.3540, 2.9010	0.3980, 0.5850, 0.6620
Observation (T4)				1.0000, 1.0000, 1.0000	1.4960, 1.9780, 2.3540	0.9450, 1.0810, 1.6370	0.4510, 0.8500, 1.0000	0.4510, 0.8500, 1.0000
Requirement Workshop (T5)					1.0000, 1.0000, 1.0000	1.1870, 1.5780, 2.0280	1.0850, 1.5000, 1.5420	1.0850, 1.5000, 1.5420

Prototyping (T6)						**1.0000, 1.0000, 1.0000**	**0.3980, 0.5850, 0.6620**	**0.3980, 0.5850, 0.6620**
Group Discussion (T7)							**1.0000, 1.0000, 1.0000**	**0.4510, 0.8500, 1.0000**
Brainstorming (T8)								**1.0000, 1.0000, 1.0000**

Table 4.4 Defuzzification by using Alpha-Cut Method

	Questionnaire (T1)	Data Analysis (T2)	Interview (T3)	Observation (T4)	Requirement Workshop (T5)	Prototyping (T6)	Group Discussion (T7)	Brain Storming (T8)	Weights
Questionnaire (T1)	**1.0000**	**1.1700**	**0.9900**	**2.5630**	**2.6670**	**2.3440**	**0.9750**	**2.5450**	**0.1688**
Data Analysis (T2)	**0.5850**	**1.0000**	**1.6300**	**1.2750**	**1.8530**	**1.7940**	**2.5450**	**2.1200**	**0.1470**
Interview (T3)	**1.1210**	**0.5710**	**1.0000**	**0.9890**	**2.6060**	**0.6910**	**2.1200**	**1.8850**	**0.1770**
Observation (T4)	**0.38600**	**0.8250**	**1.850**	**1.0000**	**2.1770**	**0.7710**	**1.8850**	**1.7670**	**0.1343**
Requirement Workshop (T5)	**0.3480**	**0.5400**	**0.3840**	**0.4590**	**1.0000**	**1.8210**	**1.7670**	**2.5450**	**0.1022**
Prototyping (T6)	**0.4750**	**0.5570**	**1.4850**	**1.2580**	**0.5458**	**1.0000**	**1.4360**	**2.1200**	**0.0901**
Group Discussion (T7)	**1.0710**	**0.4140**	**0.4720**	**0.5290**	**0.5660**	**0.6850**	**1.0000**	**1.8850**	**0.1006**
Brainstorming (T8)	**0.5570**	**1.4850**	**1.2580**	**0.5458**	**0.5570**	**1.4850**	**1.2580**	**1.0000**	**0.0800**
								CI	**0.00237**

Table 4.5 Supermatrix								
	Questionnaire (T1)	Data Analysis (T2)	Interview (T3)	Observation (T4)	Requirement Workshop (T5)	Prototyping (T6)	Group Discussion (T7)	Brain Storming (T8)
Questionnaire (T1)	1.0000	0.2520	0.1510	0.2250	0.2450	0.3450	0.2480	2.3440
Data Analysis (T2)	0.2920	1.0000	0.3470	0.2170	0.2470	0.2450	0.2230	1.7940
Interview (T3)	0.2570	0.2511	1.0000	0.2170	0.1547	0.2214	0.2230	0.6910
Observation (T4)	0.2300	0.2277	0.2140	1.0000	0.1314	0.0645	0.1180	0.7710
Requirement Workshop (T5)	0.1551	0.1514	0.0847	0.1660	1.0000	0.0450	0.0250	1.8210
Prototyping (T6)	0.0688	0.0947	0.0230	0.0890	0.0750	1.0000	0. 1240	1.0000
Group Discussion (T7)	0.1100	0.2500	0.2230	0.1000	0.1400	0.1111	1.0000	0.6850
Brainstorming (T8)	0.1100	0.2500	0.2230	0.1000	0.1400	0.1111	1.1121	1.0000

Table 4.6 Weighted Supermatrix								
	Questionnaire (T1)	Data Analysis (T2)	Interview (T3)	Observation (T4)	Requirement Workshop (T5)	Prototyping (T6)	Group Discussion (T7)	Brain Storming (T8)
Questionnaire (T1)	0.5120	0.1350	0.0540	0.1450	0.1150	0.1450	0.1170	0.1270
Data Analysis (T2)	0.1460	0.5000	0.1530	0.1270	0.1478	0.1270	0.1850	0.1780
Interview (T3)	0.1240	0.1250	0.5000	0.1160	0.0788	0.1780	0.1250	0.0780

Observation (T4)	0.1250	0.1140	0.1160	0.5015	0.0640	0.0780	0.0544	0.0780
Requirement Workshop (T5)	0.0750	0.0800	0.0440	0.0831	0.0490	0.0780	0.0145	0.5780
Prototyping (T6)	0.0340	0.0500	0.0120	0.0442	0.0370	0.5780	0.0978	0.1270
Group Discussion (T7)	0.1100	0.2500	0.2230	0.1000	0.1400	0.1111	0.5000	0.1780
Brainstorming (T8)	0.1100	0.2500	0.2230	0.1000	0.1400	0.1111	0.1101	0.5000

Table 4.7 Limit Supermatrix

	Questionnaire (T1)	Data Analysis (T2)	Interview (T3)	Observation (T4)	Requirement Workshop (T5)	Prototyping (T6)	Group Discussion (T7)	Brain Storming (T8)
Questionnaire (T1)	0.1789	0.1780	0.1789	0.1780	0.1789	0.1782	0.1785	0.1783
Data Analysis (T2)	0.1571	0.1571	0.1571	0.1571	0.1571	0.1571	0.1571	0.1570
Interview (T3)	0.1682	0.1682	0.1682	0.1682	0.1682	0.1682	0.1682	0.1682
Observation (T4)	0.1251	0.1251	0.1248	0.1250	0.1249	0.1251	0.1251	0.1251
Requirement Workshop (T5)	0.0922	0.0922	0.0920	0.0922	0.0918	0.0920	0.0922	0.0920
Prototyping (T6)	0.1031	0.1031	0.1031	0.1031	0.1031	0.1031	0.1030	0.1031
Group Discussion (T7)	0.1126	0.1126	0.1126	0.1126	0.1126	0.1126	0.1126	0.1126
Brainstorming (T8)	0.0628	0.0628	0.0628	0.0628	0.0628	0.0628	0.0624	0.0627

Table 4.8 Global Priorities of Elicitation Techniques		
Elicitation Techniques	**Global Priorities**	**Ranks**
Questionnaire (T1)	**0.1789**	**1**
Data Analysis (T2)	**0.1571**	**3**
Interview (T3)	**0.1682**	**2**
Observation (T4)	**0.1251**	**4**
Requirement Workshop (T5)	**0.0922**	**7**
Prototyping (T6)	**0.1031**	**6**
Group Discussion (T7)	**0.1126**	**5**
Brainstorming (T8)	**0.0628**	**8**

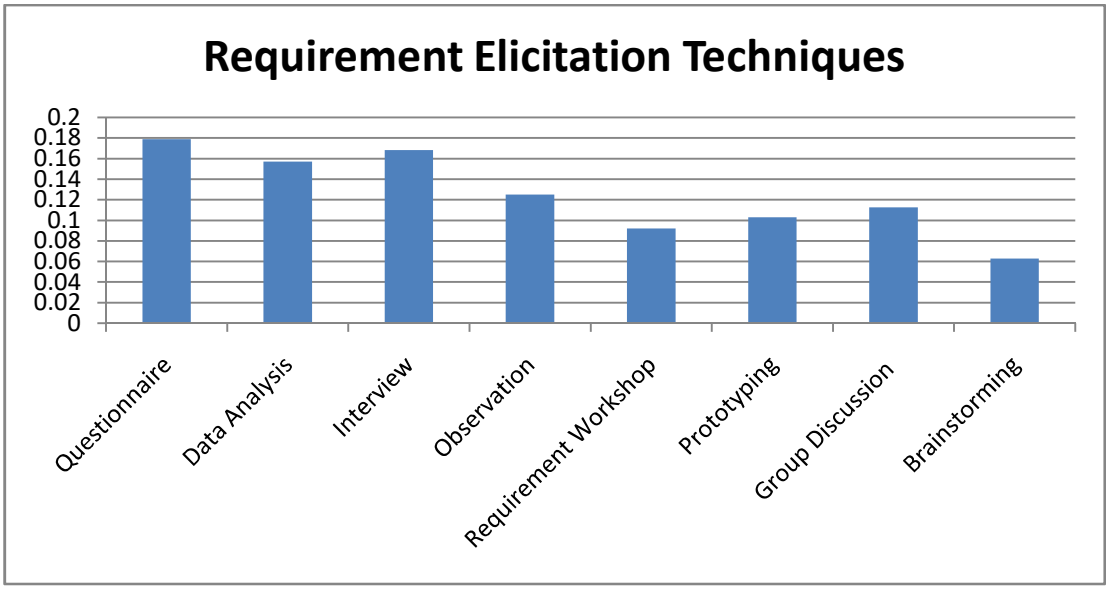

Fig. 4.4 Weights Affected by the Clusters on the Techniques of Security Requirement Elicitation

4.4 Security Attributes Assessment through Fuzzy AHP

The objective of this contribution is to identify the priorities of security requirement attributes factors. A questionnaire has been prepared

for this. Therefore, to answer the questionnaires, there is a need of experts who are thorough professionals in the field of requirement engineering [144]. Fuzzy AHP is chosen to assess the importance of security requirement attributes factors because it can control the participant's recommendations. It can also turn qualitative inputs into quantitative outcomes in the form of weight and rating, which can be used to assess the functional safety in a better way. Also, the matrix for the pair wise correlation is constructed using the Fuzzy AHP technique [146]. Expert opinions are converted to numerical values to evaluate the weight of security requirement attributes. The formulas (Eq. given below) are used to translate the values of numeric type to Triangular Fuzzy Number (TFN). They are referred to as (l_{ij}, m_{ij}, h_{ij}) where, l_{ij} is value given to if possible, mij is most likely and hij is extreme events. Furthermore, the following TFNs are known as:

$$n_{ij} = [l_{ij} m_{ij} h_{ij}] \ where \ l_{ij} \leq m_{ij} \leq h_{ij} \quad (13)$$

$$l_{ij} = \min (J_{ijk}) \quad (14)$$

$$m_{ij} = (J_{ij1} J_{ij2} \ldots\ldots\ldots\ldots\ldots J_{ijk}) 1/k \quad (15)$$

$$h_{ij} = \max (J_{ijk}) \quad (16)$$

The equation (14) and (15) has a term J_{ijk}which gives comparative value given by expert k between two criteria. Where i and j are a pair of criteria that participants are judging. Geometric mean of stakeholder scores is used to calculate the value for specific comparison. After getting the value of TFN corresponding to the comparison of a pair a matrix of the order of nxn is obtained. The size of the matrix is 9x9; twenty-five participants are considered together for achieving the considerable consistency. Fuzzy judgement matrix are generated and evaluated qualitatively by using TFN membership function and pair-wise comparisons after qualitative evaluation [153]. Table 4.9 shows the matrix that is prepared by the scholars. It includes various experts.

Table 4.9 Fuzzy Pair-Wise Comparison Matrix							
	Integrity C1	Confidentiality C2	Authentication C3	Effectiveness C4	Availability C5	Access Control C6	Authorization C7
Integrity C1	1.0000, 1.0000, 1.0000	0.1100, 0.3000, 4.0000	0.1300, 0.5000, 6.0000	0.1100, 0.2200, 4.0000	0.1100, 0.4900, 8.0000	0.1100, 0.6800, 8.0000	0.1700, 1.5400, 6.0000
Confidentiality C2		1.0000, 1.0000, 1.0000	0.1100, 1.2100, 8.0000	0.1100, 0.3100, 5.0000	0.1100, 0.6100, 9.0000	0.1700, 1.6300, 9.0000	0.1700, 1.2500, 8.0000
Authentication C3			1.0000, 1.0000, 1.0000	0.1100, 0.1900, 0.5000	0.1100, 0.4400, 6.0000	0.1100, 0.4700, 6.0000	0.1700, 2.2700, 9.0000
Effectiveness C4				1.0000, 1.0000, 1.0000	0.1700, 2.7700, 8.0000	0.1700, 3.700, 9.0000	0.1300, 0.5300, 6.0000
Availability C5					1.0000, 1.0000, 1.0000	0.1700, 1.8200, 9.0000	0.1700, 0.9400, 9.0000
Access Control C6						1.0000, 1.000, 1.0000	0.1700, 1.3900, 9.0000
Authorization C7							1.0000, 1.0000, 1. 0000

Based on the measured TFN values, defuzzification is performed to generate a quantitative value. The method of defuzzification used in this work is obtained from equation (17-19), commonly referred to as the process of alpha slicing. A fuzzy set's alpha cut has all the entities [152]. The alpha threshold varies between zero to one. The alpha threshold value used here is 0.5. Which have an alpha threshold value that is greater than or equal to its membership value, represented by α. Alpha cutting allows one to define a fuzzy set as a crisp set composition. Crisp sets $\mu\alpha,\beta(\beta ij)$ define clearly whether or not an element is a part of the set. Equations (17-19) show the method of cutting alpha.

$$\mu_{\alpha,\beta}(n_{ij}) = [\beta . n_{\alpha}(I_{ij}) + (1-\beta) . n_{\alpha}(h_{ij})] \quad (17)$$

Therefore,

$$\alpha(l_{ij}) = (m_{ij} - l_{ij}) . \alpha + l_{ij} \quad (18)$$

$$\alpha(h_{ij}) = h_{ij} - (h_{ij}) = h_{ij} - (h_{ij} - m_{ij}) . \alpha \quad (19)$$

α and β are used for expert preferences in these formulas. These two values range from 0 to 1. The result is shown in Table 4.10 by using formula (Eq 17-19) with and at 0.5 shows that the CR value is less than 0.1.

Table 4.10 Defuzzified Pair-Wise Comparison Matrix

	Integrity C1	Confidentiality C2	Authentication C3	Effectiveness C4	Availability C5	Access Control C6	Authorization C7
Integrity C1	1.0000	1.1800	1.7800	1.1400	2.2800	2.3700	2.3100
Confidentiality C2	0.8500	1.0000	2.6300	1.4300	2.5800	3.1100	2.6700
Authentication C3	0.5600	0.3800	1.0000	0.2500	1.7500	1.7600	3.4300
Effectiveness C4	0.8800	0.7000	4.0800	1.0000	3.4300	4.1400	1.8000
Availability C5	0.4400	0.3900	0.5700	0.2900	1.0000	3.2000	2.7600
Access Control C6	0.4200	0.3200	0.5700	0.2400	0.3100	1.0000	2.9900
Authorization C7	0.4300	0.3800	0.2900	0.5600	0.3600	0.3400	1.0000
						C.R.=0.0560	

Table 4.10 shows that the CR value is less than 0.1, so it is right to evaluate AHP. The next step is to determine the Fuzzy pairwise

comparison matrix's value and individual vector. The own vector is calculated to determine aggregate weight of specific criterion. Considering that μ is the own vector and λ is pair-to-pair fuzzy based matrix we get:

$$[\mu_{\alpha,\beta}(n_{ij}) - \lambda \mathrm{I}].\mu = 0 \qquad (20)$$

Transformation forms the basis of Equation (20) where the unit matrix is represented. Using eq. 4.1-4.6, it is possible to acquire the weights for specific criteria for all other relevant criteria. The security requirement attributes attribute ranks and weights are shown in table 4.11.

Table 4.11 Weight and Priority Attributes

	Weight	Percentages	Ranks
Integrity	**0.0991**	**9.91 %**	**5**
Confidentiality	**0.2704**	**27.04 %**	**2**
Authentication	**0.1132**	**11.32 %**	**3**
Effectiveness	**0.2963**	**29.63 %**	**1**
Availability	**0.0966**	**9.66 %**	**4**
Access Control	**0.0670**	**6.70 %**	**6**
Authorization	**0.0574**	**5.74 %**	**7**

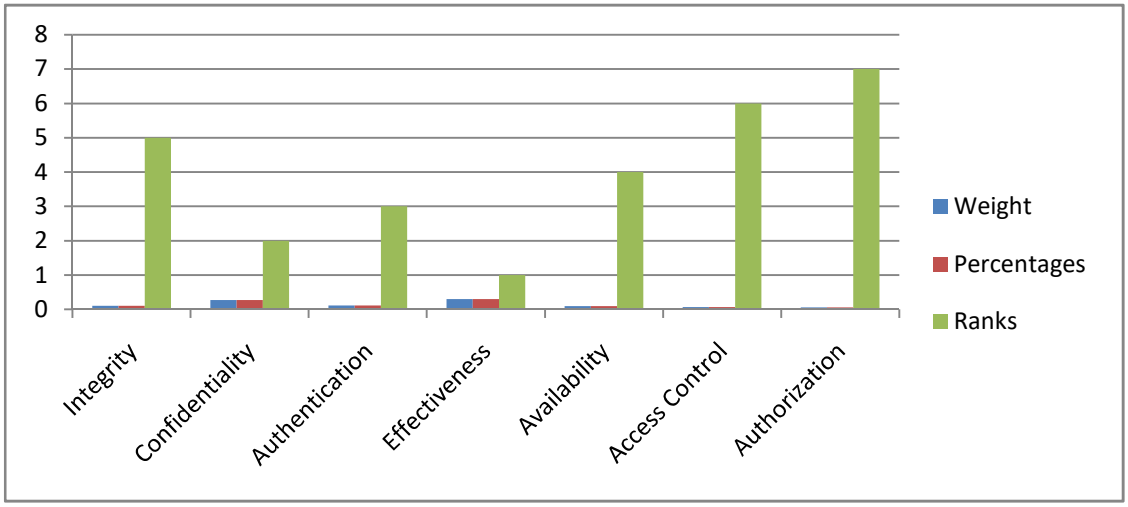

Fig.4.5Graphical representation of Weight and priority attributes

Table 4.12 Difference between fuzzy AHP and AHP				
Attributes	Fuzzy AHP		AHP	
	Weights	*Priority*	*Weights*	*Priority*
Integrity	0.0991	5	0.1268	4
Confidentiality	0.2704	2	0.1448	2
Authentication	0.1132	3	0.1405	3
Effectiveness	0.2963	1	0.3038	1
Availability	0.0966	4	0.1072	5
Access Control	0.0670	6	0.0727	7
Authorization	0.0574	7	0.1042	6

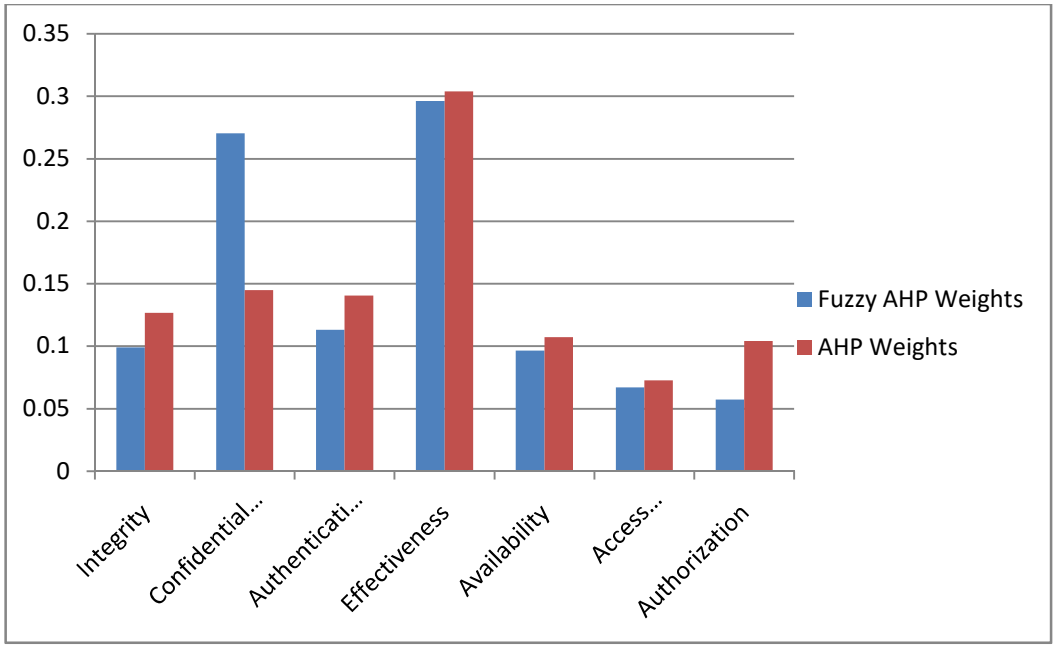

Fig. 4.6 Graphical representation of the comparison

The results obtained were rated as follows: Integrity (0.0991), confidentiality (0.2704), Authentication (0.1132), Effectiveness (0.2963), Availability (0.0966), Access- Control (0.0670) and Authorization (0.0574). The effectiveness holds the highest priority among these six attributes, according to the weights and priority. There are different security requirement attributes in the actual scenario that are present in

the process of software development. In this study, only six security requirement attributes have been defined and prioritized, affecting security. AHP is used as another tool to verify the results. Table 4.12 demonstrates correlations between Fuzzy AHP and AHP methods.

A comparison between the two methods is shown in Table 4.12. For accuracy of calculation, we compare it with AHP. The difference between these two methods is negligible as the correlation coefficient is 0.97925. This prioritization further helps to calculate the impact of these attributes on security requirement. This research also tries to provide a new methodology for calculating numeric measures from the qualitative ones while prioritizing the security attributes.

Classification of attributes related to security is done on the basis of priority and it is used by the experts to concentrate on meeting the demand of users and increase security for more time. This thesis contributes in determining by establishing hierarchy. This will help the developers to develop secure software. This work will further lead to high level of customer satisfaction.

4.5 Interpretations

In the interpretations, we explain how to quantify security attribute of requirement elicitations process, such as confidentiality, effectiveness, integrity, authentication, access control, authorization. After interpretations, we can say that instances are really a measure or affect to requirement elicitations process. This is done by AHP and ANP that explains the attributes under consideration. Here we have obtained dependent weight for security factors with different mentioned technique of requirement elicitation using some mathematical method and also calculated the dependent rank on each and every technique of requirement

elicitation, table 4.13 has been used to show the above-mentioned procedure.

Table 4.13 Dependent Weight and Dependent Rank Security Metrics

Techniques	Weights	Factors	Weights	Dependent Weights	Percentage	Dependent Ranks
T1	0.1789	Effectiveness	0.0991	0.05301	5.301%	1
		Confidentiality	0.2704	0.04838	4.838%	3
		Authentication	0.1132	0.02025	2.025%	15
		Availability	0.2963	0.01728	1.728%	20
		Integrity	0.0966	0.01773	1.773%	19
		Access Control	0.0670	0.01199	1.199%	30
		Authorization	0.0574	0.01027	1.027%	35
T2	0.1571	Effectiveness	0.0991	0.04984	4.984%	2
		Confidentiality	0.2704	0.04548	4.548%	5
		Authentication	0.1132	0.01904	1.904%	16
		Availability	0.2963	0.01625	1.625%	23
		Integrity	0.0966	0.01667	1.667%	22
		Access Control	0.0670	0.01127	1.127%	32
		Authorization	0.0574	0.00966	0.966%	40
T3	0.1682	Effectiveness	0.0991	0.04655	4.655%	4
		Confidentiality	0.2704	0.04248	4.248%	6
		Authentication	0.1132	0.01778	1.778%	18
		Availability	0.2963	0.01518	1.518%	25
		Integrity	0.0966	0.01557	1.557%	24
		Access Control	0.0670	0.01053	1.053%	37
		Authorization	0.0574	0.00902	0.902%	42
T4	0.1251	Effectiveness	0.0991	0.03707	3.707%	7
		Confidentiality	0.2704	0.03383	3.383%	8
		Authentication	0.1132	0.01416	1.416%	26
		Availability	0.2963	0.01209	1.209%	29
		Integrity	0.0966	0.0124	1.240%	28

		Access Control	0.0670	0.00838	0.838%	44
		Authorization	0.0574	0.00718	0.718%	46
T5	0.0922	Effectiveness	0.0991	0.03336	3.336%	9
		Confidentiality	0.2704	0.03045	3.045%	11
		Authentication	0.1132	0.01275	1.275%	27
		Availability	0.2963	0.01088	1.088%	38
		Integrity	0.0966	0.01116	1.116%	33
		Access Control	0.0670	0.00754	0.754%	45
		Authorization	0.0574	0.00646	0.646%	49
T6	0.1031	Effectiveness	0.0991	0.03055	3.055%	10
		Confidentiality	0.2704	0.02788	2.788%	12
		Authentication	0.1132	0.01167	1.167%	31
		Availability	0.2963	0.00996	0.996%	39
		Integrity	0.0966	0.01022	1.022%	34
		Access Control	0.0670	0.00691	0.691%	48
		Authorization	0.0574	0.00592	0.592%	51
T7	0.1126	Effectiveness	0.0991	0.02732	2.732%	13
		Confidentiality	0.2704	0.02493	2.493%	14
		Authentication	0.1132	0.01044	1.044%	36
		Availability	0.2963	0.00891	0.891%	43
		Integrity	0.0966	0.00914	0.914%	41
		Access Control	0.0670	0.00618	0.618%	52
		Authorization	0.0574	0.00529	0.529%	53
T8	0.0628	Effectiveness	0.0991	0.01861	1.861%	17
		Confidentiality	0.2704	0.01698	1.698%	21
		Authentication	0.1132	0.00711	0.711%	47
		Availability	0.2963	0.00607	0.607%	54
		Integrity	0.0966	0.00622	0.622%	50
		Access Control	0.0670	0.00421	0.421%	55
		Authorization	0.0574	0.00361	0.361%	56

Table 4.14 After Removing Redundancy of The Attributes Final Rank			
Security Attributes	**Weights of Attributes after Removing the Redundancy**	**Percentage**	**Final Rank**
Effectiveness	**0.05301**	**29.63 %**	**1**
Confidentiality	**0.04838**	**27.04 %**	**2**
Authentication	**0.02025**	**11.32 %**	**3**
Availability	**0.01773**	**9.91 %**	**4**
Integrity	**0.01728**	**9.66 %**	**5**
Access Control	**0.01199**	**6.70 %**	**6**
Authorization	**0.01027**	**5.74 %**	**7**

Table 4.14 shows the security order as given by the Weights of Attributes after removing the Redundancy and the final rank of the seven attributes based on the rank and percentage.

In this chapter, the author applied analytical network process that is based on fuzzy logic and made an attempt to examine the priority of requirement elicitation techniques by using the data received from various industrialist, researchers and experts in order to realize the activities of the real world. The original data used for the estimation was kept the same in order to check the relationship between the various methods on an equal basis. The results of these methods should be clearly showing the consistency of the findings based on the data fed.

4.6 Summary

In this chapter we have shown the assessment based on AHP, ANP and Fuzzy Based AHP, ANP. Processes has been implemented and shown here. For this alpha-cut method based defuzzification of local security requirement elicitation techniques priorities and creation of super matrix

from all local priority vectors is also shown. Fuzzy ANP methodology has been implemented here which results very effective in solving decision-making problems when there are interdependencies within the networks between different elements. The triangular fuzzy membership function is also used to construct a matrix of comparison between the different elements in pairs. The method discussed in the proposed priority of security requirement elicitation techniques include. Fuzzy logic deals with subjective reasoning and tries to solve the problem by assigning values to a wide range of ambiguous and imprecise data spectrum. Mathematical logic is used by assigning values to the incorrect data array in order to achieved at the most reliable conclusion possible. There is the allocation of localised vectors (that are priority based), to the appropriate super matrix columns in order to achieve global priority in an interdependent structure.

CHAPTER 5

FRAMEWORK VALIDATION

5.1 Introduction

Validity is basically the "measure of what is intended to be measured to the field" i.e. it explains how the recorded data is related to the field under study. Validity is classified as: Security Attributes, content validity, construct validity, criterion validity and Weight metric. Requirement elicitation judges the security attributes of a product quantitatively. Elicitation process explains how data is relevant to the AHP process and ANP process [155].

Validity establishes that given input is yielding the expected output. Forming category of security attributes priority wise is helpful for the developers to focus on meeting the client's needs and enhance the security level. The proposed work establishes the hierarchy which is utilized in requirements engineering. This work may help professionals to focus on crucial attributes related to the security of software.

We have used around 280 expert opinions to analyse and claim our results accurately and minutely. The weight of security attributes with respect to elicitation techniques for the experiment-0 has been taken from the table 4.14 in chapter 4. For this we have divided the expert opinions into 7 separate parts of 40 each and tested it on all the factors of security in respect of requirement elicitation.

5.2 Empirical Validation

The validation is done for developing the software's that have the desired features. Validation is all about studying, surveying about the product which has to be developed. These processes are usually carried out in

laboratories or classrooms. But many validation techniques are not fit for validating security attributes. Some are only used for initial validation. So after knowing about these issues, the proposed framework has some guidelines for requirement elicitation process.

Table 5.1 Weights of Security Attributes Experiment 0 vs Experiment 1 (First Set)

Security Attributes/ Weights of Attributes	Experiment – 0	Experiment - 1
Effectiveness	**0.05301**	**0.04031**
Confidentiality	**0.04838**	**0.05738**
Authentication	**0.02025**	**0.05505**
Availability	**0.01773**	**0.02413**
Integrity	**0.01728**	**0.02058**
Access Control	**0.01199**	**0.01569**
Authorization	**0.01027**	**0.01243**

This table (5.1) has used to show the experimental results of weight for the set of first 40 project values on two distinct Experiments on all the selected security requirement parameters. This shows the weight metric of security attributes. Table 5.1 shows the security attributes in ordering of Weightage perspective with two experiments (experiments-0 and experiments-1). Further, there are many values of security attributes used to test the validity. Table 5.1presents the complete values with appropriate results. The comparative study of above table 5.1 and results has been presented in figure 5.1. Hence, experiment -0 and experiment-1 are closed and bound to each other.

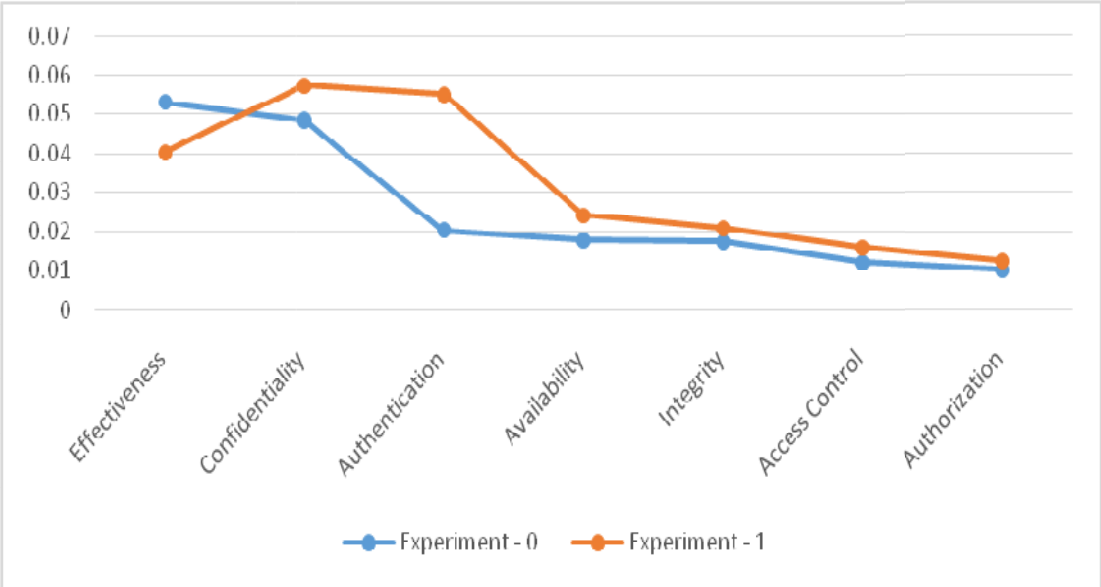

Figure 5.1 Graphical representation weight variations of different security attributes (Set One)

Table 5.2 Weights of Security Attributes Experiment 0 vs Experiment 2(set two)		
Security Attributes/ Weights of Attributes	**Experiment -0**	**Experiment – 2**
Effectiveness	**0.05301**	**0.13621**
Confidentiality	**0.04838**	**0.13138**
Authentication	**0.02025**	**0.09805**
Availability	**0.01773**	**0.09653**
Integrity	**0.01728**	**0.09288**
Access Control	**0.01199**	**0.08729**
Authorization	**0.01027**	**0.08857**

The table 5.2 have shown priority basis (Weights of Attributes) security attributes and given correspondence attributes values in the form of experiment-2.In order to identified relationship between the two measure experiments (experiment-0 and experiment-2) and for validation of internal attribute, various statistics based methods are there for analyzing the data. AHP and ANP with fuzzy is commonly used. This shows the weight metric of security attributes. The figure 5.2 shows the experiment-0 and experiments-2 of the different types of test for the validation process.

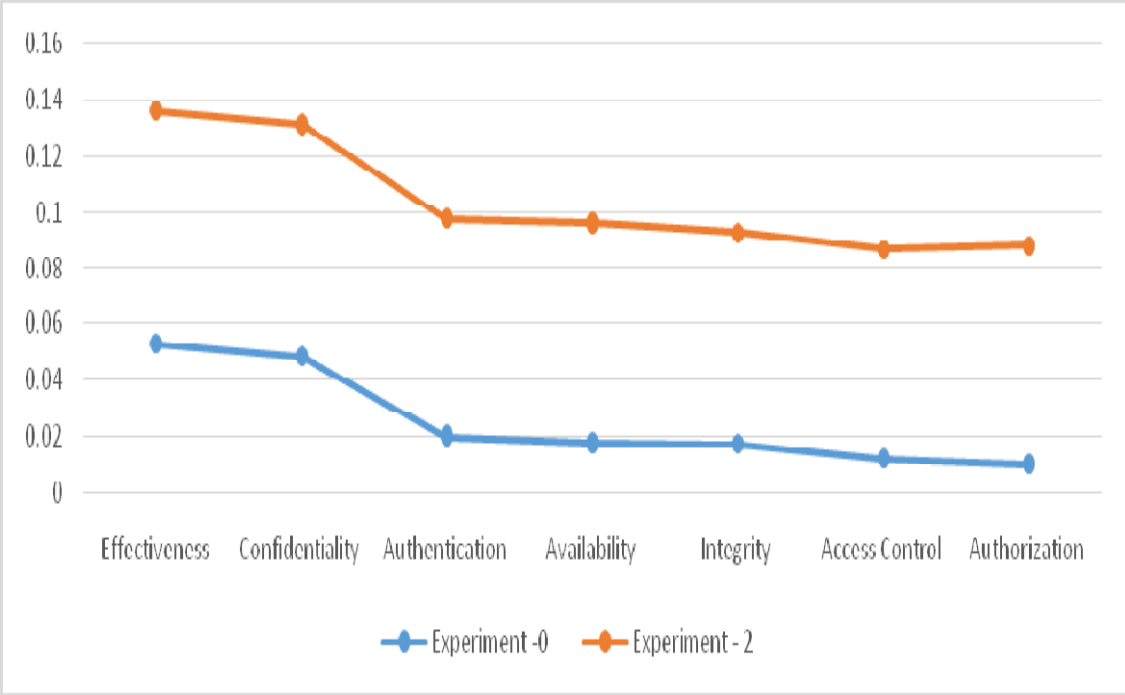

Figure 5.2 Graphical representation weight variation of different security attributes (Set Two)

Table 5.3 Weights of Security Attributes Experiment 0 vs Experiment 3 (set three)		
Security Attributes/ Weights of Attributes	**Experiment -0**	**Experiment -3**
Effectiveness	**0.05301**	**0.08901**
Confidentiality	**0.04838**	**0.08778**
Authentication	**0.02025**	**0.05325**
Availability	**0.01773**	**0.05013**
Integrity	**0.01728**	**0.04868**
Access Control	**0.01199**	**0.04369**
Authorization	**0.01027**	**0.04267**

Firstly, we have selected attributes value including the Weights of Attributes (table 5.3), whose goals were similar. Other impacts, such as relationship with two sets (experiment-0 and experiment-3) are discussed in figure 5.3.

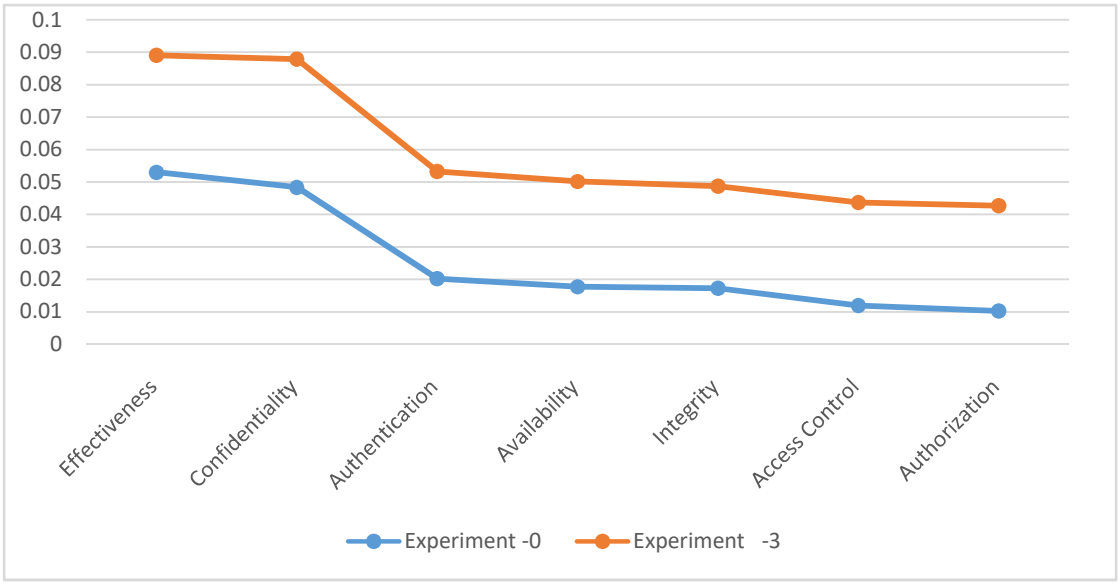

Fig 5.3 Graphical representation weight variation of different security attributes (Set three)

Table 5.4 Weights of Security Attributes Experiment 0 vs Experiment 4(set four)		
Security Attributes/ Weights of Attributes	**Experiment – 0**	**Experiment - 4**
Effectiveness	**0.05301**	**0.13041**
Confidentiality	**0.04838**	**0.12818**
Authentication	**0.02025**	**0.09025**
Availability	**0.01773**	**0.08753**
Integrity	**0.01728**	**0.08708**
Access Control	**0.01199**	**0.07969**
Authorization	**0.01027**	**0.08327**

In table 5.4, security attributes based process for validation will be exclusively considered for work related to set four (experiment-0 and experiment 4), such as priority attributes. In this observation, researcher develops the set 4 such as experiment- 4 on the basis of AHP and ANP process. Researchers of a new experiment-4 try to validate their work in experiment 0. This may be a convenient way for validation which is correlated in various perspectives. The complete impact analysis of table 5.4 have presented in figure 5.4.

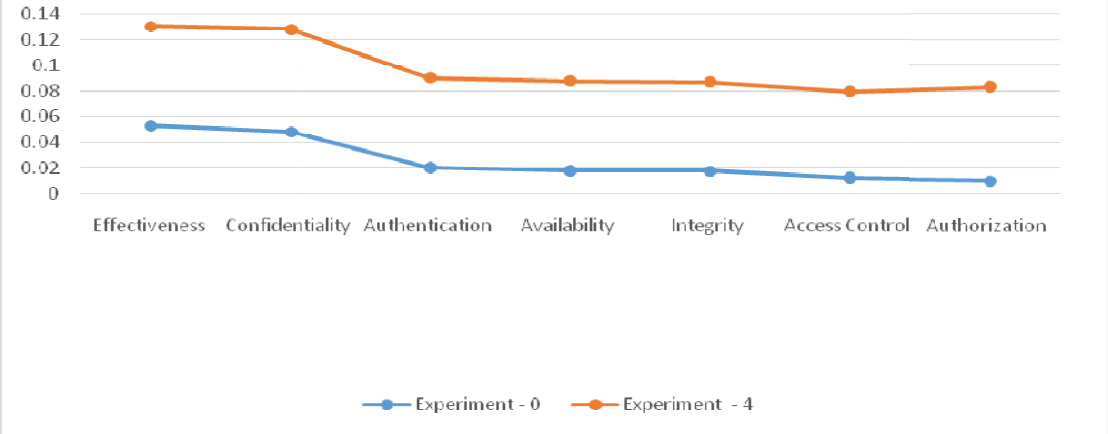

Fig 5.4 Graphical representation weight variation of different security attributes (Set Four)

Table 5.5Weights of Security Attributes Experiment 0 vs Experiment 5(set five)		
Security Attributes/ Weights of Attributes	**Experiment -0**	**Experiment -5**
Effectiveness	**0.05301**	**0.05601**
Confidentiality	**0.04838**	**0.03638**
Authentication	**0.02025**	**0.02785**
Availability	**0.01773**	**0.01753**
Integrity	**0.01728**	**0.03918**
Access Control	**0.01199**	**0.02249**
Authorization	**0.01027**	**0.01297**

Security Attributes is all about finding which method of a phenomenon yields significant results. In order to, we establish the closed relation with values of experiment 0 and experiment 5.

Security attributes values for experiment 5 is of importance because it is all about the factor of consistency throughout the experiment 0. For an exploratory study, it is shown in figure 5.5 that security attributes should be close to experiment 0 and experiment 5.

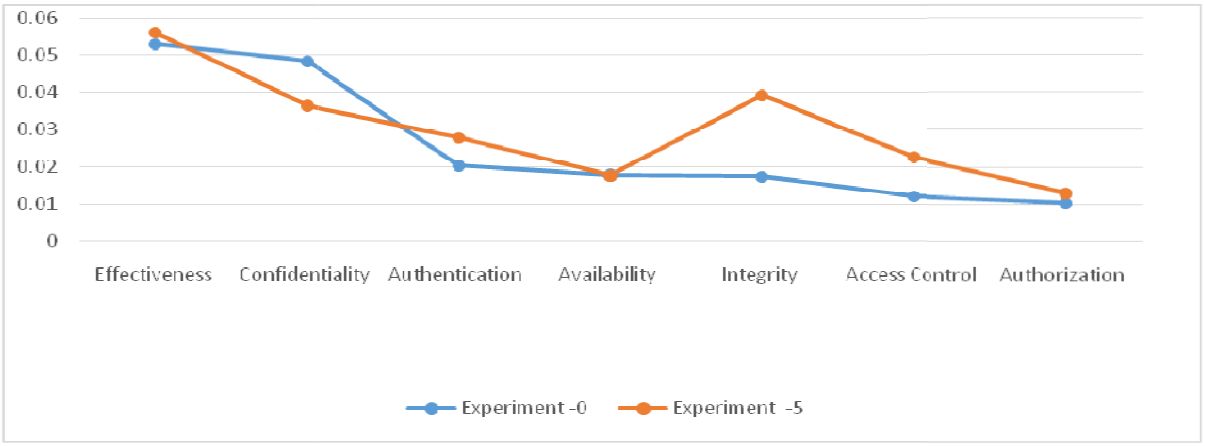

Fig 5.5 Graphical representation weight variation of different security attributes (Set Five)

Table 5.6 Weights of Security Attributes Experiment 0 vs Experiment 6(set six)		
Security Attributes/ Weights of Attributes	**Experiment – 0**	**Experiment - 6**
Effectiveness	**0.05301**	**0.05071**
Confidentiality	**0.04838**	**0.05308**
Authentication	**0.02025**	**0.01665**
Availability	**0.01773**	**0.01773**
Integrity	**0.01728**	**0.00528**
Access Control	**0.01199**	**0.00649**
Authorization	**0.01027**	**0.00857**

The approach like this is further proven by various researchers that reveals the robustness of statistics based on parameters.. This part of thesis talks about two sets where one has empirical validation as the main part of experiment-6.We want to show the result of both values set six. In this respect (table 5.6), the values have presented in figure 5.6 by the experiments set (experiment 0 and experiment 6).A comparison with experiment 0 and experiment 6 gives significant data about the use of latest recorded observations.

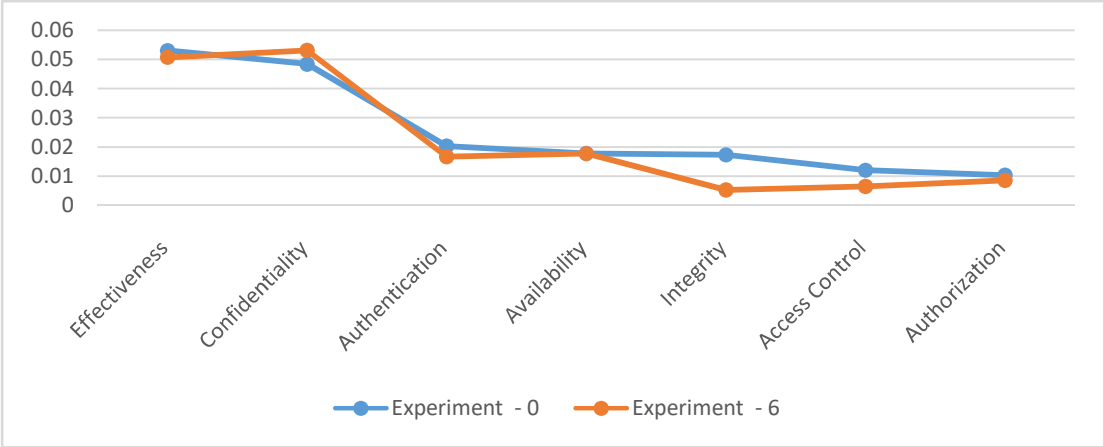

Fig 5.6 Graphical representation weight variation of different security attributes (Set Six)

Table 5.7 Weights of Security Attributes Experiment 0 vs Experiment 7(set seven)		
Security Attributes/ Weights of Attributes	**Experiment – 0**	**Experiment - 7**
Effectiveness	**0.05301**	**0.04961**
Confidentiality	**0.04838**	**0.06038**
Authentication	**0.02025**	**0.01205**
Availability	**0.01773**	**0.01753**
Integrity	**0.01728**	**0.00502**
Access Control	**0.01199**	**0.02129**
Authorization	**0.01027**	**0.00427**

The last step of validation is analyzing the outcomes (table 5.7) with two sets of experiments. This is shown in the table. The analyzed results and their comparisons will enable the software expert to pacify the clients for applying metric program in validation. The calculated values of experiment 0 and experiment 7 are depicted in graph shown below in figure 5.7.

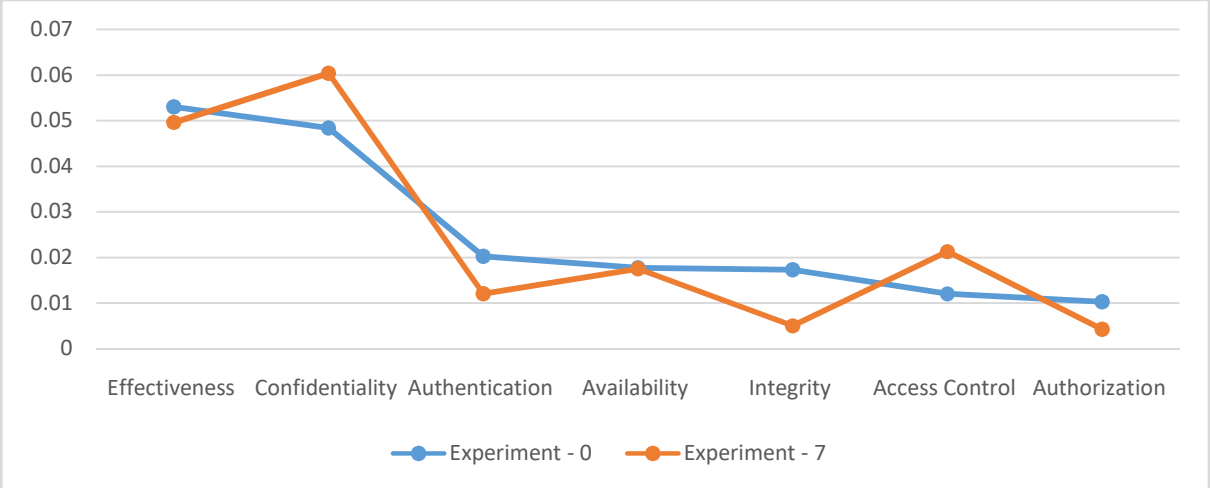

Fig 5.7 Graphical representation weight variation of different security attributes (Set Seven)

As discuss above table 5.1 to 5.7, computing experiments data sets is most effective to security attributes. Different experts from different perspective are trying to research on this environment. As shown in Figure 5.1 to 5.7, graph is about experiments v/s sets. Experiments are considered in each case of objective of given in this chapter. Here in this study, security in requirement elicitation process has the average number of quantifications which is around 40 projects. As shown in Figure 5.1 to 5.7, the difference between number of observations with experiment 0 and experiment 1 to 7. In our study total numbers of quantifications are shown in 7 tables. There is huge gap between the experiment 0 and experiment 1 to 7. This difference shows that most of analyses are done as per our research contributions. In this portion, there are AHP and ANP related to their implementation. This works is distributed either in requirement elicitations process or on a security attribute. The Figure 5.1 to 5.7 which shows the graphical representation of it.

To facilitate our experimentations (simplify analysis and calculations), we assigned rank of security attributes to the various techniques according to the rules. Table 5.8 present, the empirical study we conducted results of experiment-0 captures the evolution of project value-1 to project value-7. Table 5.8 address security metrics at experiment 0 for all seven projects. We used confidentiality, effectiveness, integrity, Authentication, Access Control, Authorization metrics values.

Table 5.8 Security metrics at Experiment 0 for all seven Experiments								
Security Attributes/ Weights of Attributes	Experiment-0	Experiment-1	Experiment-2	Experiment-3	Experiment-4	Experiment-5	Experiment-6	Experiment-7
Effectiveness	0.05301	0.04031	0.13621	0.08901	0.13041	0.05601	0.05071	0.04961
Confidentiality	0.04838	0.05738	0.13138	0.08778	0.12818	0.03638	0.05308	0.06038
Authentication	0.02025	0.05505	0.09805	0.05325	0.09025	0.02785	0.01665	0.01205
Availability	0.01773	0.02413	0.09653	0.05013	0.08753	0.01753	0.01773	0.01753
Integrity	0.01728	0.02058	0.09288	0.04868	0.08708	0.03918	0.00528	0.00502
Access Control	0.01199	0.01569	0.08729	0.04369	0.07969	0.02249	0.00649	0.02129
Authorization	0.01027	0.01243	0.08857	0.04267	0.08327	0.01297	0.00857	0.00427

5.3 Statistical Analysis

F-test is utilized for comparison of statistics based model that are fitted to a dataset. F-test usually comes into play when model is fit to the data through the least squares method [163-158].

5.3.1 Hypothesis Test

Hypothesis is more necessity for accepting the proposed work [164]. A F-test analysis apply for verifying the significance between experiment 0 and experiment -1to 8 are shown in Table 5.9 to 5.15. In table 5.9 to 5.15 shows variance value of two variables means that squared differences from the Mean of experment-0 and experiment -1to 8 simultaneously. On the basis of hypothesis test, we observe that complete value of our research hypothesis and easy to choose the kind of best suited test for your research work. In this test use the seven independent variables for secure requirement elicitations process. We shall test the

hypothesis at 95% confidence interval. Since there are seven observations, therefore the degree of freedom is six.

Table 5.9 Results between E0 and E1		
F-Test Two-Sample for Variances		
	Variable 1	***Variable 2***
Mean	0.025559	0.032224
Variance	0.000308	0.000348
Observations	7	7
Df	6	6
F	0.887194	
F Critical one-tail	4.2839	

Null hypothesis (H0): There is no significant difference between Experiment-0 and Experiment 1.

H0: $\mu 1-\mu 2 = 0$

Alternate hypothesis (Ha): There is significant difference between Experiment-0 and Experiment 1.

Ha: $\mu 1-\mu 2 \neq 0$

Table 5.9 has done between E0 and E1 and the calculated F-value is0.887194 which is less than the value of F-critical at 95%, hence the null hypothesis is accepted and alternate hypothesis rejected.

Table 5.10 Results between E0 and E2		
F-Test Two-Sample for Variances		
	Variable 1	***Variable 2***
Mean	0.025559	0.104416
Variance	0.000308	0.00042
Observations	7	7

Df	6	6
F	0.734777	
F Critical one-tail	4.2839	

Null hypothesis (H0): There is no significant difference between Experiment-0 and Experiment 2.

H0: $\mu 1-\mu 2 = 0$

Alternate hypothesis (Ha): There is significant difference between Experiment-0 and Experiment 2.

Ha: $\mu 1-\mu 2 \neq 0$

Table 5.10 has done between E0 and E2 and the calculated F-value is0.734777 which is less than the value of F-critical at 95%, hence the null hypothesis is accepted and alternate hypothesis rejected.

Table 5.11 Results between E0 and E3		
F-Test Two-Sample for Variances		
	Variable 1	***Variable 2***
Mean	0.025559	0.059316
Variance	0.000308	0.000408
Observations	7	7
Df	6	6
F	0.755975	
F Critical one-tail	4.2839	

Null hypothesis (H0): There is no significant difference between Experiment-0 and Experiment 3.

H0: $\mu 1-\mu 2 = 0$

Alternate hypothesis (Ha): There is significant difference between Experiment-0 and Experiment 3.

Ha: μ1-μ2 ≠ 0

Table 5.11has done between E0 and E3 and the calculated F-value is0.755975 which is less than the value of F-critical at 95%, hence the null hypothesis is accepted and alternate hypothesis rejected.

Table 5.12 Results between E0 and E4		
F-Test Two-Sample for Variances		
	Variable 1	***Variable 2***
Mean	0.025559	0.059316
Variance	0.000308	0.000408
Observations	7	7
Df	6	6
F	0.755975	
F Critical one-tail	4.2839	

Null hypothesis (H0): There is no significant difference between Experiment-0 and Experiment 4.

H0: μ1-μ2 = 0

Alternate hypothesis (Ha): There is significant difference between Experiment-0 and Experiment 4.

Ha: μ1-μ2 ≠ 0

Table 5.12has done between E0 and E4 and the calculated F-value is0.755975 which is less than the value of F-critical at 95%, hence the null hypothesis is accepted and alternate hypothesis rejected.

Table 5.13 Results between E0 and E5		
F-Test Two-Sample for Variances		
	Variable 1	***Variable 2***
Mean	0.025559	0.098059
Variance	0.000308	0.000467
Observations	7	7
Df	6	6
F	0.660243	
F Critical one-tail	4.2839	

Null hypothesis (H0): There is no significant difference between Experiment-0 and Experiment 5.

H0: $\mu 1 - \mu 2 = 0$

Alternate hypothesis (Ha): There is significant difference between Experiment-0 and Experiment 5.

Ha: $\mu 1 - \mu 2 \neq 0$

Table 5.13has done between E0 and E5 and the calculated F-value is0.660243 which is less than the value of F-critical at 95%, hence the null hypothesis is accepted and alternate hypothesis rejected.

Table 5.14 Results between E0 and E6		
F-Test Two-Sample for Variances		
	Variable 1	***Variable 2***
Mean	0.025559	0.022644
Variance	0.000308	0.000422
Observations	7	7
Df	6	6
F	0.729977	
F Critical one-tail	4.2839	

Null hypothesis (H0): There is no significant difference between Experiment-0 and Experiment 6.

H0: $\mu 1-\mu 2 = 0$

Alternate hypothesis (Ha): There is significant difference between Experiment-0 and Experiment -6.

Ha: $\mu 1-\mu 2 \neq 0$

Table 5.14has done between E0 and E6 and the calculated F-value is0.729977 which is less than the value of F-critical at 95%, hence the null hypothesis is accepted and alternate hypothesis rejected.

Table 5.15 Results between E0 and E7

F-Test Two-Sample for Variances

	Variable 1	*Variable 2*
Mean	0.025559	0.022644
Variance	0.000308	0.000422
Observations	7	7
Df	6	6
F	0.729977	
F Critical one-tail	4.2839	

Null hypothesis (H0): There is no significant difference between Experiment-0 and Experiment 7.

H0: $\mu 1-\mu 2 = 0$

Alternate hypothesis (Ha): There is significant difference between Experiment-0 and Experiment -7.

Ha: $\mu 1-\mu 2 \neq 0$

Table 5.15 has done between E0 and E7 and the calculated F-value is0.729977which is less than the value of F-critical at 95%, hence the null hypothesis is accepted and alternate hypothesis rejected.

5.4 Summary

In this chapter, confidentiality, effectiveness, integrity, Authentication, Access Control, Authorization has been justified and verified as a key factor to security, addressed to secure requirement elicitation process. At the validation stage, the security requirements elicitation is performed at seven security factors: confidentiality, effectiveness, integrity, authentication, access Control, authorization. The developed framework to secure the elicitation process and correlated to security factors with weight and perspective rank through AHP and ANP process. This assessment is validated by f test. The validation is empirical and theoretical both. This validation establishes the fact that proposed work has given the framework that satisfactory in security perspective.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The issues of the requirement elicitation get complicated when different stakeholders are working on a large scale process of software development. For fulfilling the work of requirement elicitation it is important to have knowledge of different techniques used for requirement elicitation. Choosing the suitable tool for communication with various stakeholders is a very important step in requirement engineering. The quality software development ultimately depends upon the effective requirement engineering process at the same time effective requirement elicitation is another important parameter for requirement gathering/ collection.

In this research, an extensive literature review was done to identify the significant security attributes affecting the secure software and then a hierarchical structure of attributes was proposed. Next, the opinion of various experts are collected on the six security attributes and among them three high priority factors are i.e., effectiveness, confidentiality, and authentication. The experts are from the software field and academia. Then the weightage of these factors have been calculated through fuzzy based AHP. Effectiveness came out to be the most critical factor. To ensure the development of a secure software product, developers must consider effectiveness of security factor.

In this research we have tried to develop the requirement elicitation process for effective requirement extraction, in which all the phase of requirement elicitation process, i.e. establishes objective, understand the background, organize knowledge, collect requirement are the main process of the requirement elicitation for developing of effective software. In this

era security requirement elicitation is a booming area for research. Factors contributing in requirement elicitation techniques are proven to be the very important entities for a secure software development process. This thesis described about security requirement elicitation techniques with respect to software development life cycle. We use FANP to make decisions taking into consideration several factors that influence such choices. We evaluate the weights affected by the clusters on the techniques of security requirement elicitation as shown in previous chapters. Hence, according to the research conducted based on Fuzzy ANP, questionnaire and interview are found to be the most prioritized techniques among all. This priority wise listing of techniques is helpful in deciding the most significant techniques among the multitude of techniques in requirement elicitation process. Further, this prioritization will also help in providing a guideline to developers for successful implementation of security requirement elicitation in software development process.

6.1 Concluding Remarks

- This work talks about the process of requirement elicitation. Additionally, this thesis also explains various techniques used for requirement elicitation, during the process of software development
- This research also depicts challenges and issues in elicitation of requirements. These issues are seen in requirement elicitation and have shown several times a key reason of system failure.
- This thesis describes the problem classifications in the context of requirement elicitation. There are many other requirements gathering methods which are helpful for one or more issues. In future, researchers may use this work to solve the above mentioned issues.
- This work will pave the way to use artificial intelligence for requirement gathering process during software development.

- The selection of supplier is a pivotal task for the companies and this selection depends on various criteria. The alternative suppliers should be inspected effectively in order to avoid conflicts in criteria. Various methods are there for this task. The existing methods are TOPSIS, SPECTRE etc. This work has used fuzzy based Analytical Hierarchy Process. This is because in general the preferences of developers depend on various intangible and tangible criteria, and they can be represented easily through Fuzzy set theory.
- For multiple criteria decision-making Fuzzy AHP plays an important role. When Fuzzy AHP used scientific weights that are derived from comparison matrices. The present approaches that are used for determining fuzzy weights are very complex

6.2 Limitations

- Lack of resources to conduct the experiment i.e. we were not able to find any industrial contact to conduct the experiment. Our experiment required software analysts from industry. Response from industries was not positive because software organizations have their own ongoing projects therefore it was not possible to engage someone from industry.
- If we have more time, we should choose more complex method i.e. more steps in the framework. Just using more time with a simple method will not improve the number of problems found. We had to select the method depending on several issues e.g. available time.
- We have few Data points in our experiment. Few data points may affect generalization of results. As mentioned earlier, our results and conclusions are inferred from one system which was randomly selected.
- Our framework is work intensive and we can see from the experiment that its activities need time to execute. This un-addressed issue can be a limitation of the study. Our motive in the study was to find more

number of security requirements than a formal or informal technique. Reducing activities of framework would simply be at the cost of security requirements. Making a less work intensive Framework and testing its applicability becomes out of scope for us in the present study.

6.3 Future Work

- In future, various models like ELECTRE, Fuzzy logic based models can be used to find the solution of the problem used in this work and its results can be compared with this work. Moreover, for problems related to different sources, use of mathematical models can be done.
- In future, this work can be extended to develop parameters that can be used for determination of exact requirements related to security of software.
- This work proposes a LFPP methodology which gives a unique priority vector for any fuzzy pair wise comparison. This methodology will lead to more research and work in the field of Fuzzy AHP and its uses in near future.
- In our framework we have focused on Requirement phase of Software Development Lifecycle. We considered only those activities of CLASP and Secure TROPOS which were related to Requirement elicitation. In future, one can add more activities to the framework that are related to other software development Life cycle Phases. This will help to invoke security in all phases of development lifecycle.
- We have tested our Framework for one software system. For future work one can check the practicality of framework for different systems. In this procedure improvements in our framework can be suggested.

6.4 Take Away from the Research work

- Security requirement elicitation helps focus security in software system. We have mentioned before that cost related to fixing security problem is much more than the cost of security requirement elicitation in earliest phase of software development.
- The flexibility of framework and its capability to permit iterations improves the security of system. Moreover, the framework has defined set of steps to elicit security requirements
- This research was an attempt to provide a flexible framework for security requirement elicitation. We achieved that by mitigating the weaknesses in formal and informal security requirement elicitation techniques by combing them. In the questionnaire, which was filled by the participants of the experiment, the participants agreed that using this framework improved their knowledge about importance of security in software systems.
- During this research we learned and discovered that there are many security requirement elicitation techniques proposed by researchers but almost none of them are used in industry on a large scale.
- Most of the big brand names like IBM and CISCO have their own, customized, set of steps to ensure security in their product. It is worth noting that these "customized set of steps to ensure security" also reflect organizational structure. Academic research in this field is not used in industry at large scale.
- Reason behind not including research outputs in security requirement engineering field is that small organizations do not have resources for that. Big organizations have developed their own customized procedures over time to handle security in their product and it is costly to switch to anything new suddenly.
- Therefore it is unfortunate that academic research in this field is not given attention as it deserves.

- It is clear that not every software development organization gives desired attention to security in software and hacking incidents around the world are the evidence to our argument. That means, the organizations that produce vulnerable software do not have any formal procedures to view security as a major concern in a software system. To make such an organization refer to the academic research in this regard, we suggest creating a demand for secure software development.
- Social awareness about software security is very important because it will create a demand for secure software systems. Since software has a major role to play in our lives today, we suggest that Software vendors, Customers and Researchers must understand the importance of security in software systems. Demand for systems which are secure from customer's and vendor's perspective will automatically force both big and small organizations to refer to academic research in software security, especially security requirements elicitation.
- This may also help to bridge the gap between industrial organization and academics in the context of security requirement engineering.