CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The critical and important role of Requirement Engineering (RE) in software development has already been highlighted by many researchers [29][30]. The process of requirement engineering encompasses, Requirement Elicitation, Analysis, Specification, Validation and Management of requirements [32]. Major problem areas discussed in literature concerning RE are acquisition, conception and unpredictability of requirements [34]. The Requirement Elicitation is the first and the most important phase of requirement engineering which deals with the gathering of requirements [35][36]. Most of the research work in requirement engineering is focused on the requirement representation and modeling but not how these requirements are to be well collected and documented [33]. RE is first and important step in the software development similarly requirement elicitation is also the first and important step in requirement engineering [31]. This research work is focused on requirement elicitation activities, tools and techniques. This chapter presents literature review which involved a thorough review and critical analysis of existing theory on and around the area of requirement elicitation for software systems .The primary objective of this review was to establish a preliminary set of approach guidelines and tool features, and to provide a theoretical foundation for the research.

2.2 Software Engineering

Software products have significantly changed the facets of our work and daily life. The software products are seen around us, in the working place, office, car, the Internet, etc. Software development has already become one of the largest industries in the world. Software engineering as a discipline emerged in the late 1960s and became increasingly
important during the past 40 years. The overarching objective of software engineering is to bring sound engineering disciplines to the development of software products to improve their quality.

2.2.1 Definitions

Software Engineering has been defined differently depending on the perspectives of the software industry, such as science or engineering. Most literature gives credit to the definition given by in the report on the NATO conference [40]:

“Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.”

This definition gives a general picture of software engineering.

The definition given in the IEEE Standard Glossary [37] of Software Engineering Terminology is:

“Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, the application of engineering to software.”

This definition emphasizes the nature of software engineering in SDLC as engineering discipline. A more comprehensive definition of software engineering is given by D.M. Berry [38] which describes its different facets:

“Software engineering is that form of engineering that applies a systematic, disciplined, quantifiable approach, the principles of computer science, design, engineering, management, mathematics, psychology, sociology and other disciplines as necessary and sometimes just plain invention, to creating, developing, operating and maintaining cost-effective, reliably correct, high quality solutions to software problems.”

Additional definitions can be found in various other literature such as [39][41][42][43].
2.3 Requirement Engineering

2.3.1 Overview

The terms requirement and engineering were first tied together by [44] in the development of SREM (Software Requirement Engineering Method). Requirement Engineering RE was initially applicable to various categories of information systems, and hence was oriented towards organizational and application issues. Since the word engineering has been attached to requirements, RE research efforts have endeavored to incorporate the engineering approach to traditional systems analysis. Originally, RE activities are only related to requirement analysis and requirement specification which is the core activity of SDLC. In the 1990s, it became widely accepted that RE is a key process in the software lifecycle and the scope of requirement engineering has extended far beyond the traditional system analysis. RE has already become an established discipline and includes a variety of skills, processes, methods, techniques and tools.

2.3.2 Definition

The concept of requirement is fundamental in requirement engineering. A “requirement” is defined as [37]:

(1) “A condition or capability needed by a user to solve a problem or achieve an objective”.

(2) “A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents”.

Generally speaking, the concept of “requirements” discussed in the RE domain refers to [45].

“What a system should do rather than how it should do it”.

Having defined requirements, two definitions for requirement engineering are given.
According to [46],

“Requirement engineering is a systematic process of developing requirements through an iterative, co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats and checking the accuracy of the understanding gained.”

This definition gives a very intuitive description of RE and is highly related to representational, social, and cognitive issues.

On the other hand, the definition of RE given by [47] is considered more comprehensive.

“Requirement engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families.”

This definition reflects the nature of RE as a multidimensional discipline. It suggests that requirement engineering is not only related to technical issues and problems but also to managerial, organizational, economic, and social issues. Accordingly, RE might not solely be a front end process, but also be part of the later stages of software engineering.

### 2.3.3 The Importance of Requirement Engineering

There is an increasing awareness of the importance of RE. Many researchers emphasized that the RE process is an essential contributor to the overall quality of the software product based on empirical analysis and software industries experiences [48][49]. Interestingly, the CHAOS report [1] published by the Standish group also strongly supports this argument. According to [18], effective RE practices contribute more than 42% towards the success of a project, compared to other factors. Similarly, improper RE practices account for more than 43% of the reasons that projects are not delivered in time or over budget. Moreover, a case study was carried out by Hall et al. in 12 companies at different levels of capability measured by using CMM [50]. The results
show that out of a total of 268 development problems cited, almost 50% were due to poor RE. All these studies suggest that RE plays an important role in software development.

2.3.4 Requirement Engineering Process

According to [46] requirement engineering is a process itself. RE process is seen as a collection of well-defined activities, techniques, and transformations that people use to develop system requirements, maintain the requirements specification and associated artifacts. Numerous RE process models have been developed in the past two decades [45][51][52]. Each RE process model focuses on different aspects of RE and has different granularity.

[53] proposed a Viewpoints model, called VORD, which includes requirements discovering, requirements analysis, requirements validation, and requirements documentation activities. Later, proposed a more general RE process model in [54], which includes the following, five main phases: requirement elicitation, requirements analysis and negotiation, requirements specification, requirements verification and validation, and requirements management. Each phase can be considered as a process itself that includes a set of activities which implement certain functions.

- Requirement **Elicitation** is concerned with the collection, capture, discovery, and development of requirements from a variety of sources including human stakeholders.
- Requirement **Analysis** focuses on examining, understanding, and modeling the elicited requirements, and checking them for quality in terms of correctness, completeness, clarity, and consistency.
- Requirement **Specification** is the act of recording and documenting the requirements in a way that can be used by the stakeholders, and especially the developers who will design and construct the system.
- Requirement **Validation** is the process of confirming the quality of the requirements, and ensuring that they actually represent the wants and needs of the stakeholders.
• Requirement Management is performed throughout RE process and includes activities such as change control, version control, requirements status tracking, and requirements tracing.

It is worth mentioning that each RE activity has its own merits. It is beyond the scope of this research to discuss all existing requirement engineering activities. A more comprehensive survey of existing requirement engineering activities can be found in [51].

2.4 Requirement Elicitation.

Requirement elicitation is the first phase of RE. It can therefore be argued that RE process starts from requirement elicitation, and consequently a prerequisite for all the other major development activities. Requirement elicitation is an important activity of RE because requirements gathered from this phase will signify whether the developed software product will work properly or not [59]. Moreover, software project’s failure or success is mainly dependent on the quality of gathered requirements gathered because it is the base for the software product to be established. Effective requirement elicitation considerably enhances the quality of requirements and consequently quality software product will be developed. Therefore there is significant need to understand and apply elicitation techniques properly for successful requirement engineering phase [60].

The researchers suggest that “Requirement Elicitation is the process of collecting meaningful requirements from end users is an early and major goal of any and all software engineering processes” [55]. Despite this, the term ‘requirement elicitation’ is relatively new even untouched as compared to other requirement engineering activities. “Requirement Elicitation is still often referred to as requirements collection, capture, acquisition, determination, gathering, identification, invention, development, discovery, and fact-finding”. For one reason or another these other terms have been insufficient to represent real activities, and subsequently it is now well documented and accepted that requirements are elicited rather than just captured or collected [56]. Currently there is very little uniformity in requirement engineering research and practice concerning a standard definition for requirement elicitation.
“Requirement elicitation is the practice of obtaining the requirements of a system from users, customers and other stakeholders. The practice is also sometimes referred to as requirements gathering”.

Hickey and Davis [20] define requirement elicitation as

“Learning, uncovering, extracting, surfacing, and/or discovering needs of customers, users, and other potential stakeholders”.

An alternative definition of requirement elicitation is

“ The process of identifying software or system requirements from various sources through interviews, workshops, workflow and task analysis, document analysis, and other mechanisms” [57].

The definition of requirement elicitation is further elaborated as

“Requirement elicitation is all about learning and understanding the needs, desires, and expectations of users and customers, with the ultimate aim of communicating these to all the stakeholders and especially the system developers”.

For the most part, however, requirement elicitation is dedicated to uncovering, extracting, acquiring, and elaborating the needs of stakeholders.

“Requirement elicitation refers to this process as trawling for requirements” [58].

To highlight the fact that through effective requirement elicitation process more quality requirements are gathered. The requirements are elicited in such a way that relevant requirement of the project are gathered. As discussed, requirement elicitation is closely interrelated to the other requirement engineering activities, and in particular the phase of analysis.

2.4.1 Analysis of various surveys

The failure of the software projects has been concerns of the software industry from many years. The ultimate aim of software engineering is to develop such software
systems that are up to the mark set by the customer’s side in terms of needs, schedule and budget. Keeping in view these aims of software development, success rate of such developments is not very encouraging. Many surveys have been done to investigate causes of projects failure and their statistics. According to the famous Standish reports [16][18], success rate of software project is only 28%. A major contributing factor in such a low rate of success is said to be unclear and imprecise requirements [25]. Another survey also pointed out that only 12.7% out of 1027 software projects were successful and top reason for the failure was “unclear objectives and requirements” [156]. These reports indicate that a major contributing factor in such a low rate of success of software project is unclear and imprecise requirements. In [28] researcher discovered that “accurately capturing system requirements is the major factor in the failure of 90% of large software projects”. The work was reflection of [26] according to whom “poor requirements management can be attributed to 71% percent of software projects that fail; greater than bad technology, missed deadlines, and change management issues. The cost of this failure is enormous”.

Requirement Elicitation is all about determining the needs of stakeholders and discovering what the stakeholders want. It is one of the most critical and important activity in requirement engineering. Researchers also show that 70% of the systems errors are due to improper requirements and remaining 30% is due to the design faults. It is a reasonably well documented fact that requirement elicitation has a big impact on final product quality [1][27]. After detailed study it is evident that most of the surveys conducted by various researchers gave common reasons pertaining to requirement engineering out of many other reasons as the major factors for the failed and the challenged projects which are summarized in the table below.
Year | Report (Chaos report) | Requirements as reason of failures of projects
--- | --- | ---
1994 | Standish report | User Involvement-15.9%, Clear Statement of Requirements 13.0%, Realistic Expectations 8.2%, Clear Vision & Objectives 2.9%
1995 | Standish report | 13.1% due to incomplete requirements, 12.4% due to Lack of User Involvement, 9.9% due to Unrealistic Expectations, 8.7% due to Changing Requirements & Specifications, 7.5% didn't needed Any Longer
2000 | Standish report | User involvement, Clear business objectives, Minimized scope, Firm basic requirements,
2001 | Standish report | 16% as User involvement, 12% as Clear business objectives, 10% as Minimized scope 6% as Firm basic requirements
2007 | Standish report | User Involvement 20%, Clear Business Objectives 15%
2011 | Standish report | User Involvement, Clear Business Objectives, Agile Business requirement
2013 | Standish report | 15% User Involvement, 6% Clear Business Objectives, 10% Agile Business requirement

Table 1.1: Showing major causes of the software projects failures along with their ranks [27]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Successful</td>
<td>29</td>
<td>35</td>
<td>32</td>
<td>37</td>
<td>39</td>
</tr>
<tr>
<td>Failed</td>
<td>18</td>
<td>19</td>
<td>24</td>
<td>21</td>
<td>18</td>
</tr>
<tr>
<td>Challenged</td>
<td>53</td>
<td>46</td>
<td>44</td>
<td>42</td>
<td>43</td>
</tr>
</tbody>
</table>

Table 1.2: Project Resolution Result from Chaos Research [1][16][18][30]
From the above analysis as discussed in the table 1.1 and 1.2 it is evident that poor requirement engineering especially requirement elicitation, is the most prominent factor for the failure of the software projects. The survey shows that in all the failed projects the 50% reasons is the poor RE, where as the rest of 50% is due to other reasons. So capturing requirement accurately in requirement elicitation is the major challenge that most of the researchers faced. Requirement Elicitation plays a vital role in capturing requirement during requirement engineering. Requirement elicitation is recognized as one of the most critical, knowledge-intensive activities of software development; poor execution of elicitation will almost guarantee that the final project is a complete failure. Since project failures are so rampant, it is quite likely that improving how the industry performs elicitation could have a dramatic effect on the success record of the industry.

2.4.2 Existing Models of Requirement Elicitation

Despite the large array of possible contexts described above researchers proposed a no of process models for requirement elicitation over the years [58][61][62][63][64].These models are based on different variations and core set of basic activities. For the most part these models provide us generic roadmap of the process with sufficient flexibility to accommodate the key contextual differences of individual projects. The inability of these models to provide effective guidelines is a result of the wide variety of tasks that may be performed during requirement elicitation, and these sequences of those activities being dependent on specific project situations.

There are varieties of issues and problems faced by the elicitor and the number of tools, techniques and approaches available to be used makes the task of developing a useful and useable process model more complicated. Furthermore in most of the cases requirement elicitation is performed incrementally over various sessions, iteratively to increase levels of detail, and at least partially in parallel with other SDLC activities. In order to identify and elaborate the problem of existing process models, it is important to determine the core components of the requirement elicitation process to guide the development of more flexible models. [64] states that the requirement elicitation involves the understanding of i) the problem domain to be solved, ii) the business domain of the organization, iii) the way in which the system is to be worked iv) the application domain of the software system. [62] suggests that a good requirement
elicitation process will include i) Object setting, ii) Background knowledge Acquisition iii) Knowledge organization iv) Stakeholder requirements gathering”.

[57] suggests that requirement elicitation gathering plan should be developed, which should include requirement elicitation objectives, requirement elicitation strategies and processes, requirement elicitation products efforts, schedule and resource estimates, and elicitation risks. Although these all address the base of a requirement elicitation process, they provide a list of high-level areas of required investigation. In response to this situation, a number of models have been designed to provide more specific and effective guidelines for the process of requirement elicitation. [20] proposed a unified model for the requirement elicitation process using mathematical concepts. This provides researchers with a more structured and complete knowledge of the process, it offered little in the way of hands on guidance for elicitors.

[65] proposed a step-by-step methodology using the ISO 9126 quality characteristics as a guiding component for the elicitor to capture human, social, and organizational factors that can enhance the quality of the developed software project. Although this largely did not offer advantage or integrate much of the existing requirement elicitation research to date, it did, however, provide researchers with a more tangible and detailed set of guidelines for performing requirement elicitation process yet at a fairly high level of abstraction. One of the reasons for this gap is requirement elicitation process guidelines as discussed by various researchers, is the large number of situational variables and components that can affect how the process should and can be performed.

One of the more obvious of these is the type of SDLC process model that is implemented, and which the requirement engineering and therefore the requirement elicitation activities must support. The type of SDLC model select, such as the Waterfall, Spiral, Evolutionary, or Incremental models each have their own respective model for requirement engineering which in turn affects the specific requirement elicitation process. Therefore, what is needed to improve our understanding requirement elicitation process is a more detailed investigation into the common and underlying activities of typical requirement elicitation process. To this end and to present our own overview of the requirement elicitation process a framework is needed for effective requirement elicitation process that supports elicitor to understand and select methods
for requirement elicitation process also addresses core problem areas of requirement elicitation. In this thesis a framework is proposed for effective requirement elicitation.

### 2.4.3 Activities of Requirement Elicitation

The core activities of Requirement Elicitation are divided into following categories

- Understanding the Domains
- Identifying the Sources
- Manage Stakeholders
- Selecting the Elicitation Methods
- Eliciting the Requirements
- Organizing the Requirement

#### 2.4.3.1 Understanding the Domains

The process of requirement elicitation can have a variety of starting points such as a problem, an opportunity, or an idea; however, more typically it is a problem of domain that must be solved [66]. This may result in dissatisfaction with the current situation and the development of a new situation, or an opportunity to improve the situation in terms of time and cost. Typically the process begins with an informal and incomplete high-level statement for the software project [67]. This may be expressed by a set of fundamental goals, functions, and constraints for the target system, or as a description of the problems to be solved. In order to develop this description, project stakeholders and other requirement sources are identified and used to elicit information about the environment in which the product will be situated.

These preliminary results form the basis of further investigation and revision of gathered information in a typically iterative and incremental manner. It is important at the beginning of the requirement elicitation process to investigate and examine methodically and in detail the real-world situation in which the product will ultimately reside, sometimes known as problem domain, in addition to the type of product being developed, known as the application domain [68]. These can be collectively referred to as the operating environment of the software system, and needs to be thoroughly studied and explored in addition to the technical, political, organizational, and social aspects
concerned with the project and the system, as well as any rules & constraints that may need to be enforced.

A recent study [69] found that “software projects with a moderate structure or well defined structure of the problem domain usually performed less iterations” of requirement elicitation process. An analysis of both the problem domain and application domains involves better understanding of different elements of the entire system. This includes not only the type of project to be performed and the type of system to be developed, but also the size and kind of the organization, and the sort of business it conducts [70]. Evidently, it is important to analyze the potentially large and diverse number of contextual factors related to the domains, which affects the process of requirement elicitation [71].

2.4.3.2 Identifying the Sources

In most cases software system requirements are elicited from, a number of sources in various contexts [36]. The current systems and processes already in place represents main source for eliciting requirements, particularly when the project involves replaces an existing system [72]. Stakeholders are more specifically the eventual end-users of the software system; represent the most obvious source of requirements gathering. One of the important and first steps of requirement elicitation is to identify and analyze all the relevant stakeholders [58]. This includes identification of their individual goals, ideas, motives, and what potentially influence the project [74]. All parties involved and affected, directly or indirectly, by analyzing, developing and implementation of the target system, considered as stakeholders [73]. Therefore stakeholders include the client, customers, analyst, elicitor, management, end users, developers, and other project members for lists of key project stakeholders[58][75][77].

Different types of user can also be identified depending on several factors, including, frequency, use of the system, and experience with software systems, features, tasks, access privileges or security levels associated with. It is critical for successful requirement elicitation that all the target system stakeholders are involved in the project process [57]. Although the level of identified stakeholders have an impact on the project and it depends ultimately on various factors like, availability, cooperation, and
ownership. Most of the researchers suggest that user involvement is critical and important to the success of the requirement elicitation process [76].

2.4.3.3 Manage Stakeholders

Stakeholders Management is an important activity of requirement elicitation. To effectively manage the stakeholders is the key to effective requirement elicitation process.

Stakeholders analysis

Requirement elicitation involves a wide range of people involvement. These people include customers or clients who pay for the system, users who interact with the system to get their work done, software developers who design, construct, and maintain the software product, and policy makers who impose rules and regulations on the development and operation of the software product known as stakeholders [73][78]. They are from diverse platforms, expertise, interests, goals and personalities [47]. The identification and prioritization of stakeholders that can influence or be influenced by the software project is known by the term stakeholder analysis. These stakeholders are the source of requirements during requirement elicitation [77][78]. Early practices in stakeholder analysis focused mainly on paying customers. Soon it was realized that a lot of focus is required for stakeholders analysis [79]. Also, in addition to users who will benefit from the product, people who will lose from it should also be considered, as they may oppose the product [77][80]. As such, modern requirement elicitation broadens its scope to involve and analyze stakeholders.

Stakeholders Identification

The task of identifying key stakeholders is far from straightforward [73]. Information about stakeholders is not readily available and it is difficult to obtain a complete list of stakeholders [77]. Omitting stakeholders is reported as the most common problem in software product [80]. The majority of elicitors face problems in finding the right stakeholders with adequate time, interest, and knowledge for the project [58]. Elicitors
often omit key stakeholders and the omission significantly impacts overall success of project. The project had to be cancelled when the omission was discovered as the auditor’s requirements were not met by the project. [80] proposed context-free questions to identify information about stakeholders, such as: customer, team, problem does the product solve, who are the right people to ask these questions. [83] Suggested identifying stakeholders by considering those who interact directly or indirectly with the system and those who have interests in the system but do not interact directly with it. The output of these approaches is a list of key stakeholders of the project [80][83]. [84] proposed a more systematic approach to stakeholder identification that involves interviewing stakeholders to identify key stakeholders. The approach starts by identifying stakeholder and their roles. The individuals taking up these stakeholder roles are then interviewed to identify the initial list of stakeholders. The interview uncovers new stakeholders, who are interviewed in turn, and their perceptions integrated into the stakeholder list.[84] applied their approach to identify key stakeholders of information systems in the management domain projects. The interview method is thorough. The underlying principle of this proposed approach is that stakeholders cannot be viewed in isolation. The interrelations between stakeholders are used to identify stakeholders who are otherwise likely to be overlooked. Some recent approaches for stakeholder identification provide a checklist of key stakeholders. [77][85] proposed the Onion model which contains customizable slots of generic stakeholder roles in different layers, similar to the layers in an onion . Roles that directly interact with the system are close to the centre roles that are indirectly involved are further away from the centre. The model is used by asking stakeholders what their roles are and populating the model down to the contact details of the people filling each role.
2.4.3.4 Selecting the Methods

Once the sources of requirements and the key stakeholders have been identified, the actual task of eliciting the core requirements begins with the selection of elicitation techniques from a variety of tools and techniques. Although some researchers suggest that just one elicitation technique is sufficient, it is generally accepted that single requirement elicitation technique cannot possibly be suitable for effective requirement elicitation process [20]. In fact the quality of the requirements is greatly influenced by the techniques selected and employed during requirement elicitation.

Requirement elicitation is such a complex process involving various activities that it is typically best performed by using a variety of the available tools and techniques. By using a combination of elicitation technique, many of the issues commonly associated with requirement elicitation process can be minimized. The relative strengths and weaknesses of these techniques determine their use in a particular project situation. The choice of technique to be used is therefore strongly related to the context of the project and their situation [62]. Despite it often being a critical factor of the success of the elicitation process, technique selection has received relatively little attention [72], however, some of the researchers suggest guidelines for selection of requirement elicitation techniques.[86] in their ACRE framework provide descriptions, preconditions for use, perceived strengths and weaknesses, and useful references for twelve elicitation techniques.[20] also analyzed selection investigated elicitation technique selection at length, and state that a particular elicitation technique may be selected for any combination of various reasons. Both of these efforts have attempted to develop a guideline for selection of requirement elicitation technique.

A sentiment shared by researchers [86][20] is that selection of elicitation technique is not fixed, during requirement elicitation process combination of techniques need to be selected and used, and the selection of the next technique will be dependent on the results of previous used techniques. Therefore technique selection itself is iterative process. [20] suggests that right technique to apply in a given situation must be a function of what requirements already known to us and what requirements are still need to known; after all, different techniques are good at uncovering different kinds of
requirements. Selection of requirement elicitation technique depends on the experience and expertise of the elicitor for specific problem and situation.

2.4.3.5 Eliciting the Requirements

During the actual task of eliciting the requirements, it is important to establish the level of scope for the system and investigate in detail the needs and stakeholders requirements, typically over a number of sessions using various requirement elicitation techniques. Although the process is called requirement elicitation, in reality more than just the requirements for the software system are elicited.

The requirement elicitation process is also concerned with gathering other types of requirements of the organization, the people, and the organization environment including system goals, business rules, work processes, assumptions, constraints and implementation details. [87] suggest that it is important to elicit other types of information during requirement elicitation process in order to complete the system knowledge and situation, including a description of the present work in the domain, a list of the present requirement engineering problems, goals, critical issues, ideas and risk, among others. This is consistent with the perspective of concept modeling, where requirement are elicited about the processes, data, and behavior of the target project [88], and also supported by the traditional structured analysis and design view of software project [71].

Determining the priority, source, and particularly the rationale of the requirements and goals are also very important components of the requirement elicitation process [89]. This supports the view that requirement elicitation is the process to discover of not only what is required, but also why it is required. Therefore during requirement elicitation process, it is essential to elicit more than just the requirements of software system.

2.4.3.6 Organizing the Information

Once the requirements of a system has been elicited from all the available sources, it then needs to be organized and integrated in such a way as can be later accessed, analyzed, validated, negotiated and prioritized.
This process involves merging the gathered requirements from the different sources, categorizing them, and storing them in a standardized template. Typically the deliverable from requirement elicitation phase, and eventually the entire requirement engineering process, is a document or a list of candidate requirements containing different types of information [20]. The output of the requirement elicitation process however is also ultimately dependent on the intended audience and therefore needs to be in a template understandable to those users.

In general a successful requirement elicitation process will produce a quality requirements document that reflects a common vision and provides a better understandability of the problems, needs, constraints, related to the project development. Therefore it is important to achieve the right level of detail for the information of requirements. There are a number of well known problems with using natural language for requirements representation, but using a more formal notation early in the project, risks of making the requirements impossible to understand for the stakeholders [90].

The success of software project is heavily dependent on the quality of the requirements used for software project development, and in turn on the process used to elicit them. It is also important that requirements are organized and represented in such a way so that it can be tested to determine their quality [87]. Ultimately the how and when to end the process is really a judgment call by the elicitor that, takes courage, and should be based on the project situation, and should be agreed to by all the stakeholders [66]. [91] in his work ‘Just Enough Requirements Management’ has addressed this specific issue directly.

2.5 Problems of Requirement Elicitation

Problems of requirement elicitation can be categorized into:

- Problem of scope
- Problem of understanding
- Problem of volatility
2.5.1 Problem of scope

Elicitation techniques need to be broad enough to establish boundary conditions for the target system, yet still should focus on the creation of requirements as opposed to design activities. Avoiding contextual issues can lead to requirements which are incomplete, not verifiable, unnecessary, and unusable. In brief the problem of scope is expressed as:

- The boundary of the system is ill-defined
- Unnecessary design information may be given

2.5.2 Problem of understanding

Users have incomplete understanding of their needs, capabilities and limitations of system and poor knowledge of problem domain. Problems of understanding during elicitation can lead to requirements which are ambiguous, incomplete, inconsistent, and even incorrect because they do not address the stakeholders true needs. In brief the problem of scope is expressed as:

- Problems of understanding
- Users have incomplete understanding of their needs
- Users have poor understanding of computer capabilities and limitations
- Analysts have poor knowledge of problem domain
- User and analyst speak different languages
- Ease of omitting obvious information
- Conflicting views of different users
- Requirements are often vague and untestable.

2.5.3 Problem of volatility

The changing nature of requirements is known as volatility. During the time it takes to develop a system the user’s needs may mature because of project domain knowledge brought on by the development activities, or they may shift to a new set of needs because of unforeseen organizational or environmental pressures. If such changes are not accommodated, the original requirements set will become incomplete, inconsistent with the new situation, and potentially unusable because they capture information that
has since become obsolete. One primary cause of requirements volatility is that “user needs evolve over time”. Another cause of requirements volatility is that the requirements are the product of the contributions of many individuals, and these individuals often have conflicting needs and goals.

2.6 Requirement Elicitation Techniques

As the name implies, a requirement elicitation technique is a technique used by the elicitor to elicit requirements from stakeholders and other sources. More generally a ‘technique’ is a process doing something or a practical method applied to some specific task [92]. A technique is intended to provide a way and guidance for both the elicitor and stakeholders in requirement elicitation, in order to avoid the ‘blank slate syndrome’ common when stakeholders are asked to produce information related to requirement [55]. In reality there are various available techniques from a variety of sources that can be employed for requirement elicitation process. An early survey by [56] examined at a relatively high level only a small number of the more traditional elicitation techniques such as interviewing, observation, and task analysis. In a more recent survey on requirement elicitation [67][20][91], several additional and more current techniques were examined including those based on goals, scenarios, viewpoints, and domain knowledge.

In this chapter some of those elicitation techniques are presented which are more widely used, in order to evaluate their relative strengths and weaknesses in eliciting requirements and addressing the current issues. Although this selection is by no means complete or exhaustive, believe it is believed that representative of the range of available techniques described in the literature and performed in practice today.

The requirement elicitation techniques are categorized into various categories so that the elicitor understands the purpose and use of variety of techniques. Therefore, for the sake of simplicity and understanding of elicitation techniques, based on our categorization and considered to be the most important characteristic of each technique, which also has some relationship with other selected elicitation techniques [78]. This classification has been cited elsewhere in the literature [93], and provides a suitably broad and logical coverage of the available requirement elicitation techniques for the purposes of
evaluation and comparison. [94][95][96][97][98][99] presents different models for selection of requirement elicitation technique.

2.6.1 Traditional Techniques

Traditional requirement elicitation techniques include a broad class of generic elicitation techniques [31][97]. These techniques are those which have been used since the beginnings of software engineering for the purposes of determining the needs and expectation of stakeholders, even before requirement elicitation had been established as a separate area of interest within software development.

2.6.1.1 Interviews

Interviews are probably the most traditional and commonly used technique for requirement elicitation by elicitor [95][100]. Interviews are essentially human based social activities, they are inherently informal and their effectiveness depends greatly on the quality of interaction between the stakeholders.

Interviews provide an efficient way to collect large amounts of data quickly from different stakeholders [158]. The quality of the results from interviews process, such as the usefulness of the information gathered, can vary significantly depending on the skill of the interviewer or elicitor [101]. Interviews are categorized into three types: unstructured, structured, and semi-structured. Unstructured interviews are conversational in nature where the interviewer enforces only limited control over the direction of interactions. Because they do not follow predetermined questions, there is the risk that some topics may be completely neglected. It is also a common problem with unstructured interviews process focus in detail on some specific areas, and not enough in others [102]. Because of their dynamic nature, unstructured interviews require a significant amount of skill for better performance. Structured interviews are conducted using a predetermined set of questions. The success of structured interviews depends on knowing what are the correct questions to ask, when should they be asked, and who should answer them. Templates that provide guidance on structured interviews for requirement elicitation such as Volere [58] can be used to support this technique.
The novice analyst is suitable for evaluation, and they require limited training and require less time [103].

Although dependent on the purpose, in general meaning interviews should start with basic and high-level topics, and it focuses on the specific and relevant item. Because of 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, particularly with respect to various issues and problems. However the problem with interviews for requirement elicitation process is that they can be costly and time consuming in terms of preparation, execution, recording and analysis, and typically it is necessary to interview multiple stakeholders at various steps in order to obtain a set of quality requirements[158].

2.6.1.2 Questionnaires

Questionnaires are mainly used during the early stages of requirement engineering and may consist of open-ended and/or closed questions [99]. Questionnaires are to be effective, in terms, concepts, and boundaries of the domain must be well established and understood by the participants and questionnaire designer.

Questions must be focused to avoid gathering large amount of redundant and irrelevant information [100]. They provide an efficient way to collect large amount of information from multiple stakeholders quickly, and can easily be performed remotely. However they are limited in the depth of knowledge they are able to elicit, and in general provide little supporting contextual information. Most importantly, questionnaires lack interactivity, and consequently do not provide the opportunity to explore further on a new topic or expand on fresh ideas as interview do. In the same way they provide no mechanism for the participants to request clarification or correct misunderstandings. Furthermore, the design of questionnaires may be difficult as open-ended questions results are hard to analyze, and the results from closed questions can be easily misinterpreted. Generally questionnaires are considered more useful to determine basic attitudes, beliefs, and characteristics, as informal checklists to ensure fundamental elements are addressed early on, and to establish the foundation for subsequent elicitation activities.
2.6.1.3 Task Analysis

Task analysis employs a top-down approach. In Task Analysis, high-level tasks are decomposed into subtasks and eventually into detailed sequences until all actions and events are described [104][105]. The primary objectives of Task Analysis is to construct a hierarchy of the tasks performed by the system and the users and determine the knowledge used or required to carry them out. Task analysis provides information on the interactions of both the user and the system with respect to the tasks as well as a contextual description of the activities that take place. In most cases considerable effort is required to perform thorough task analysis, and it is important to establish what level of detail is required and when components of the tasks need to be explored further. Despite this task analysis is a useful technique to employ in order to investigate usability problems.

2.6.1.4 Domain Analysis

Examining related documentation and applications in the target domain of the system is very useful and important technique for gathering early requirements and identifying reusable concepts and components. These types of investigations are particularly important when the project involves the replacement or enhancement of an existing legacy system, or when the analyst is not familiar with the organization. Types of documentation that may be useful for eliciting requirements include design documents and instruction manuals for existing systems, and hardcopy forms and files used in the current business processes. Domain Analysis can provide good background information about the organization, the business, and its processes. However the available documentation may contain large amounts of irrelevant information, or worse, may be inconsistent with the real operations. Domain knowledge in the form of detailed descriptions and examples on the other hand can also play an important part in the requirement elicitation process. Approaches based on this type of information are often used in conjunction with, and as the input for, other elicitation techniques. These approaches also provide the opportunity to support reuse specifications and validate new user requirements against other domain instances [100][106]. Problem Frames in particular provide a method for detailed problems examination in order to identify patterns that could provide links to potential solutions [107].
2.6.1.5 Introspection

The technique of introspection requires the analyst to develop requirements based on what he or she believes the users and other stakeholders want and need from the software system [101]. Despite being employed by most analysts to some extent, this technique is mainly used only as a starting point for other requirement elicitation efforts. Introspection is effective when the analyst is not only very familiar with the problem domain and goals of the system, but also expert in the business processes performed by the users. In cases where the elicitor is forced to use this technique more, for example when the users have little or no previous experience with software systems in their work environment, a type of facilitation introspection should take place via other elicitation techniques such as interviews and protocol analysis. Given that when using this technique the elicitor is required to project what the stakeholders think, introspection has the potential to be highly inaccurate [101].

2.6.2 Cognitive Techniques

Cognitive techniques include a series of techniques originally developed for knowledge gathering [78][100]. These techniques aim to elicit requirements by representing and structuring the knowledge of stakeholders in terms of how they see both the problem and solutions domains.

2.6.2.1 Card Sorting

Card sorting requires the stakeholders to sort a series of cards containing the names of domain entities into groups according to their own understanding [96][100]. Furthermore the stakeholder is required to explain the rationale for the way in which the cards are sorted. It is important and effective technique in which all the entities are included in the process. This is possible only if the domain is sufficiently understood by both the elicitor and the participants. If the domain is not well established then group work can be used to identify these entities. This technique is often used more for the categorization and clarification of requirements rather than elicitation. Class Responsibility Collaboration (CRC) cards are a derivative of card sorting that is used also to determine program classes in software code [108]. In card sorting technique
cards are used to assign responsibilities to users and components of the system. Because entities represent such a high level of system abstraction, the information obtained from this technique is limited in its detail.

2.6.2.2 Laddering

In laddering technique, stakeholders are asked a series of short predefined prompting questions known as probes, about one or more concepts related to the target system, and are then required to arrange the answers into an organized set according to their own needs, understanding and preferences [95] [109].

The knowledge, which is often displayed using tree diagrams of interlocking ladders, is then reviewed and modified dynamically as more information is added. Much like card sorting, laddering is mainly used as a way to clarify and understand user requirements and categorize domain entities. A primary assumption when employing laddering is that the knowledge to be elicited can actually be arranged in a hierarchical fashion. For this technique to be effective, the stakeholders must be able to express their understanding of the domain and then arrange it in a logical way.

2.6.2.3 Repertory Grids

Repertory grids involve asking stakeholders to develop attributes and assign values to a set of domain entities [110]. As a result the system is modeled in the form of a matrix by categorizing the elements of the system, detailing the instances of those categories, and assigning variables with corresponding values to each one.

The aim Repertory grid is to identify and represent the similarities and differences between the different domain entities. These represent a level of abstraction unfamiliar to most users. As a result this technique is typically used when eliciting software requirements from domain experts. Although more detailed requirement are gathered than card sorting, and to a lesser degree using laddering technique, repertory grids are somewhat limited in their ability to express specific characteristics of complex requirements.
2.6.3 Group Techniques

Group elicitation techniques aim to foster stakeholder agreement and buy-in, while exploiting team dynamics. They typically require various stakeholders working together in order to generate ideas and specifications for the target system [100].

2.6.3.1 Brainstorming

Brainstorming is a technique where participants from different stakeholder groups engage in informal discussion to rapidly generate as many ideas as possible without focusing on any one in particular, where quantity is paramount and not quality[111]. This is typically followed by a consolidation stage where the number of ideas is narrowed down by removing those ideas that the group immediately identifies or recognizes as inappropriate or unsuitable, and then the remaining ideas are examined and evaluated, refining and combining them until the group is satisfied with the results. It is important when conducting this type of group work to avoid exploring, critiquing or analyzing ideas in detail. All stakeholder present should actively participate with equal worth be creative, and all the ideas generated should be recorded, no matter how unrealistic they may seem. It is not usually the intended purpose of brainstorming sessions to resolve major issues or make key decisions. Brainstorming is often used to develop the preliminary mission statement for the project and target system. One of the advantages in using brainstorming is that it promotes freethinking and expression, and allows the discovery of new, creative, and innovative solutions to existing problems.

2.6.3.2 Requirements Workshop

Requirement workshop is a technique given to a number of different types of group meetings where the emphasis is on developing and discovering requirements for a software system [112]. There are many different types of requirements workshops depending on their purpose and participants. Workshops are a well established, very common and often default technique for effective requirement elicitation [100]. This technique is particularly effective because they involve and commit the stakeholders directly and promote cooperation. In workshop various sessions are organized to understand the needs and requirement of the stakeholders. These types of sessions can
be difficult to organize due to the number of different stakeholders that may be involved in the project. Managing these sessions effectively normally requires a highly trained facilitator with both expertise and experience to ensure that individual personalities do not dominate the discussions. [112] recommends the use of collaborative patterns such as Divide, Conquer, Correct, Collect, as well as walkthroughs and checklists can be used to further enhance requirements workshop.

The key factors in the success of group work are the makeup of participants and the cohesion within the group. Participants in workshop should be motivated and cooperative, have the right mix of skills and knowledge, and should all share a common goal [112]. Workshops are often performed using support materials such as documents, diagrams, and prototypes to promote discussion and feedback. Workshop encourages stakeholders to resolve conflicts and develop solutions themselves, rather than relying on the analyst to drive the process. One of the other obvious advantages of using group workshops is the ability to integrate other elicitation techniques into them, and then to incorporate their combined usage into a defined requirements process [113].

2.6.3.3 Focus Groups

Another variation of requirements group work derived and often used in market analysis and subsequently market-driven systems development is Focus Groups [114]. Focus group is a kind of group interview, with a session lasting not last more than two hours and involving around a dozen people at the most. Typically this technique is used to discuss a specific topic or address a particular problem. In software development, focus groups often use stimulus material such as questionnaires, prototypes, or storyboards to provoke and encourage dialog among the participants. Because the facilitator usually takes a passive role during the session, and the level of structure required is flexible, focus groups allow more natural conversation than more defined group work approaches. Providing that the participants are expert in the area of discussion, focus groups can also enable the elicitation of important, useful, and accurate opinions and perceptions about the target system. Although relatively practical and economical to conduct, focus group is not well suited to finding solutions to complex problems or eliciting hard requirements.
2.6.3.4 Group Meeting

The Group Meeting represents a more structured form of brainstorming. Group Meeting generally begins with the participants generating ideas in writing but anonymously based on a problem stated by the facilitator [100]. Each participant then reads out one idea in turn in a round-robin fashion recorded by the facilitator, until all ideas have been presented. 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. The steps of discussion and voting may continue to take place as necessary several times until a general decision is achieved. This technique is particularly useful in politically or socially sensitive situations as it attempts to balance the influence of all the individual participants, thereby reducing somewhat the negative effects of group dynamics. Group Meeting is the technique in which participants seek a group solution as a process of problem solving rather than by negotiation. Although effective for consensus building, the Nominal Group Technique tends to be limited to a single purpose and single topic meeting, and requires significant preparation and strong facilitation. This can be used as an alternative to the Delphi technique [115] and focus groups, and has been integrated within other types requirement elicitation workshops include JAD [116].

2.6.4 Contextual Techniques

Contextual techniques have been utilized for requirement elicitation as an alternative to both traditional and cognitive techniques[78][95] These techniques focus on requirement gathered directly from the context in which the target system will eventually exist, that is the specific environment in the real world.

2.6.4.1 Ethnography

Ethnography being the study of people in their natural setting is a form of social analysis and involves the elicitor actively or passively participating in the normal activities of the users over an extended period of time gathering information on the process being performed [117].
Ethnography in its various forms is especially useful when addressing contextual factors such as usability, and when investigating collaborative work settings where the understanding of interactions between different users with the system is important. This technique is particularly effective when the need for a new system is a result of existing problems with processes and procedures, and also in identifying social patterns and complex relationships between human stakeholders. Researcher [101] suggested that ethnography should be used throughout the requirement elicitation process to provide the contextual basis for the results of other elicitation techniques.

2.6.4.2 Observation

Observation is one of the more widely and traditional technique used as contextual technique [118]. This technique suggests, the elicitor simply observes the actual execution of existing processes by the users without direct interference. Observation is often used in conjunction with others such as interviews and task analysis. As a general rule, observation is expensive to perform, and requires significant skill and effort on the part of the elicitor to interpret and understand the actions being performed.

The effectiveness of observation can vary as users have a tendency to adjust the way they perform tasks when knowingly being watched. Furthermore, observational studies may need to be carried out over a long period of time in order for the analyst to be able fully understand what is actually taking place. Despite this, observation technique is considered to be a good technique for gathering domain knowledge of current process and associated issues, however this also create a bias towards what is presently being done.

2.6.4.3 Protocol Analysis

Protocol analysis is the technique in which participants perform an activity, task, wish list talking it through aloud, describing the actions being conducted and the thought process behind them [101]. Protocol analysis provide the analyst with specific information on and rationale for the processes the target system must support [102]. In most cases however talking through an operation is not the normal way of performing the task, and as a result may not necessarily represent the true process completely or
correctly. Likewise minor steps performed frequently and repetitively are often taken for granted by the users, and may not be explained and subsequently recorded as part of the process.

2.6.4.4 Prototyping

Providing stakeholders with prototypes of the system to understand the problem and provides support for the investigation of possible solutions is an effective way to gather detailed information and relevant feedback [54]. It is common and advantageous technique to use prototypes in conjunction with other elicitation techniques such as interviews and group work to encourage discussion and debate. Prototypes are typically developed using preliminary requirements or existing product of similar features. This technique is particularly useful when developing human-computer interfaces, or where the stakeholders are unfamiliar with the available products, and there is a great deal of uncertainty about the requirements [95].

There are a number of different methods for prototyping systems such as storyboards, executable, throwaway and evolutionary, with varying levels of effort required. In many cases prototypes this technique is expensive to produce in terms of time and cost. However, an advantage of using this technique is that it encourages stakeholders, and more specifically the users, to play an active role in developing the requirements, as it is easier to discuss an actual tangible system. One of the potential hazards in using this technique for requirement elicitation is that users may become attached to them, and therefore become resistant to alternative solutions from then on. Despite this the technique is extremely helpful when developing new systems for entirely new applications.

2.7 Requirement Elicitation technique selection

The previous sections of the literature review shows that every elicitation technique has some advantage and limitations [101], and even for the most common and natural one of interviews [158], there is insufficient available support for navigating the specifics of a requirement elicitation process, and no guidance for specifying the requirements [119]. [101] in their seminal work on requirement elicitation techniques present a very
good and detailed description of the social problems and limitations of many of the requirement elicitation techniques.

In [120] “The problem is that requirements gathering methods tend to fall into two categories: those which produce rich results but are expensive and those that are less expensive but also less informative”. An important point is that although elicitor may be familiar with several requirement elicitation techniques, no one technique in isolation is able to capture all the product requirements completely [121]. Therefore more than one technique is required to elicit all the actual requirements for complex and large software based systems. There are varieties of requirement elicitation techniques available, and because of this wide range of techniques it is possible to use alternative techniques in many situations, which enable greater flexibility of the process and more choice for the elicitor and stakeholders. It can also been seen that most of these techniques do not originate from the traditional areas of Software Engineering. Techniques for requirement elicitation are mostly derive social sciences, organizational theory, group dynamics, knowledge engineering, and very often from practical experience. As a result, some techniques are effective in eliciting domain knowledge, or user knowledge, or current and future situations.

Most requirement elicitation techniques are informal and involve human to-human interaction. Of all the techniques, group work is particularly effective as it would appear that groups are able to deal with complex tasks such as requirement elicitation process better than individuals because they have a wider range of skills and abilities to draw from. Group work techniques are also beneficial for requirement gathering because they involve the users, commit the customers, promote discussion, collaboration, idea generation, solution finding, and decision making. Another advantage of Requirements Workshops is that they combine with other requirement elicitation techniques. Furthermore these group techniques are naturally very important to the requirements gathering because software development is inherently a group effort [125]. However there is some debate over the relative performance of facilitated workshops versus one-on-one interviews. In a recent study [122] it was found that one-on-one interviews as prescribed by method, [124] were found to be more efficient for small projects, whereas facilitated workshops in accordance with the DSDM method [123] were more efficient.

2.8 Requirement Prioritization

Requirement prioritization is the process to prioritize the requirement according to needs and requirement of stakeholders. After requirements are identified, they also need to be prioritized. A function can always be added and the user interface enhanced. Although some project requirements are critical for the success of the software product, others may merely be adornments [126]. Hence, software requirements should be prioritized so that the ones that are most likely to achieve customer satisfaction can be selected for implementation [127]. A prioritization technique commonly used in practice is the numeral assignment technique [127][128]. In this technique, each requirement is assigned a value representing its perceived importance.

A variation of this technique is called the Planning Game in extreme programming, where customers distribute requirements into three groups: “Those without which the system will not function, those that are less essential but provide significant business value, and those that would be nice to have” [129]. A finer granularity of the scale of perceived importance can range from 1 to 5, where 5 indicates mandatory, 4 indicates very important, 3 indicates rather important, 2 indicates not important, and 1 indicates does not matter [126].

Numeral assignment is straightforward, but [126] found that numerical values are subjective, and the scoring system is often inconsistent as different people make use of different personal scales. Similar to the numerical assignment technique, many existing approaches prioritize requirements from an individual’s perspective. As prioritizations involve a small subset of stakeholders, the results are biased towards the perspective of those involved in the process [130]. The value-oriented prioritization method proposed by [131]. Value-oriented prioritization method constructs a prioritization matrix using core business values and risks [131]. More sophisticated methods combine prioritizations from multiple stakeholders. In the 100-point test or cumulative voting,
each stakeholder is given 100 points that they can distribute as they desire among the requirements [132]. Requirements that are more important to a stakeholder are given more points. Requirements are then prioritized based on the total points allocated to them. Many existing prioritization methods consider requirements to have a flat structure and be independent of one another [133]. However, requirements are often defined at different levels of abstraction [134]. Hierarchical cumulative voting proposed by [135] enables prioritizations to be performed at different levels of a hierarchy.

In the requirements triage method, [99] proposed that stakeholders should be gathered in one location and group voting mechanisms used to prioritize requirements. In the win-win approach proposed by Boehm, stakeholders negotiate to resolve disagreements about candidate requirements [136]. Using the win-win approach, each stakeholder ranks the requirements privately before negotiations start. The requirement prioritization framework proposed by [137] incorporates various aspects of requirement prioritization from existing literature. [138] proposes a framework for requirement prioritization. [139] proposes A simulation-based fuzzy multiplet attribute decision making for prioritizing software requirements.

2.9 Conclusion

Requirement Elicitation is recognized as one of the most critical and important activity of software development [140], poor execution of elicitation will almost guarantee that the final project is a complete failure. It is generally accepted that the quality and success of a software system depends on the quality of the requirements upon which it has been built and how well the final system meets those requirements. Therefore if the requirements do not satisfy the problems the system is intended to address, then the chance for project success is very small. In fact poor execution of elicitation will almost guarantee that the final project is likely to be a failure. Poor requirements will reduce the quality of the software, introduce defects, create costly rework, cause late delivery of the system, and create customer and user dissatisfaction. High quality requirements on the other hand might not guarantee the results from the development of a system, and plays an important role in project success. From the above literature survey it is concluded requirement elicitation process is the most important and critical activity of requirement engineering. It can therefore be seen that requirement elicitation process
requires an extensive skill set combined with experience to be performed well. A lot of research is done in this field but still 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 Requirement Elicitation Framework is proposed in this research study that gives suggestive solution to these problems.