

# Comparative Study of Information Gathering Methods and Tools

Sa'adatu Abdulkadir<sup>1</sup> and Francesca Nonyelum Ogwueleka<sup>2</sup>

<sup>1,2</sup>*Department of Computer Science, Nigerian Defence Academy, Kaduna, Nigeria*

**Abstract**— Software development life cycle depends on the software requirements where researchers and practitioners alike have long recognised and established that requirements elicitation is the most essential phase of the requirement engineering process in any software development project. There is a need to collect necessary information about the requirements and the environment of a system-to-be using the appropriate methods and tools during the requirements elicitation stage of any software development life cycle which needs to be accomplished within a certain time and budget. This involves the use of system analysts who are responsible for collecting the information and the stakeholders who provide the necessary information. There are various requirement elicitation techniques and tools available for correct and complete requirement gathering. This paper reviewed various requirements elicitation techniques from the stakeholders' point of view based on their strengths and weaknesses. The paper also summarises the comparison of requirement elicitation techniques and software requirement tools using various parameters and functionalities.

**Keywords**— Software Engineering, Software Requirements, Requirement Elicitations, Software Development Life Cycle.

## I. INTRODUCTION

Information gathering helps an individual and organizations to undertake complex tasks that would generally be particularly hard to carry out if not out-rightly impossible without the assistance of information gathered. Literally, information gathering is a fundamental skill, which is necessary for humans so as to undertake basic human actions such as eating, sleeping, and working. For example, for one to eat, there is a need to know if the food is edible or not and also for one to sleep, there is a need to determine the cosiness of the sleeping place.

Nowadays, computer and software applications are highly utilised and they have an impact in the management of businesses and other specialized organizations, such as scientific, military, academic, and so forth [1]. Information gathering is an advanced skill, which requires the training and education of staff with emphasis on the rules and methods of gathering information from sources that are of advanced level than ordinary sources. The more information gathered, the more and better mode of dealing with the current problem and this also broaden the minds of those who will use the data obtained to solve current organisational problems. As long as

the facts and figures are used as the basics, new ideas will come more easily.

Requirements gathering or information gathering for any software development project is the most crucial stage of any software development life cycle. In order to facilitate successful software delivery, there is a need to have software quality requirements irrespective of the software methodologies used either the agile or traditional approach. In other words, determining software requirements is the foundation for a successful project and this is a difficult task because it involves natural language to relate with the stakeholders who may specify partial and vague requirements [2]. Some researchers in [3] have reported some substantial evidence in their research literature of quality software requirements. Wiegers (as cited in [3]) argued that "If you do not get the requirements right, it does not matter how well you execute the rest of the project." The authors in [5] (as cited in [4]) stated, "Requirements are critical for defining, estimating and managing any project". Reference [6] stated that good requirements should have the following quality criteria: necessary, atomic, independent, feasible (realistic, possible), understandable, correct, clear (concise, terse, simple, precise), verifiable (testable), unambiguous, and implementation-free (abstract) and having these qualities will ensure the success of any software developed.

Requirements elicitation, also called information gathering, is the process of compiling the requirements of a system or requirement gathering from users and customers known as stakeholders. In addition, [7] stated that requirements elicitation is all about learning and understanding the needs of stakeholders, after which these needs are conveyed to the system developers. Without proper understanding or knowledge of what the stakeholders' requirements, projects cannot be developed based on what the stakeholders' desire [4] and the software project is bound to fail. Reference [8] (as cited in [3]), over the epoch of twenty years, assessed 10 software project failures and three of these failed projects have been due to poor requirements analysis. The author also noted that poor requirements occupied the eighth position in his list of the top 10 blunders made by software developers [8]. Some of the other sources of failures may additionally include: inadequate client involvement, vague and ambiguous requirements, un-prioritized requirements (may lead the software developers to squander time in low prioritized

activities), building functionalities that no one uses, confusion over “Requirements” due to stakeholders’ requirement description or due to verbal communication that leads to confusion, among others.

This paper is structured into five sections. This section provides the introduction of requirement elicitation and in section II is the literature review. Section III, which is divided into two main sections where one section is on the requirement elicitation techniques and its comparison and the other section briefly mention some software tools that support requirement elicitation and its comparison. Section IV discusses the findings of the study and the last section presents the conclusion.

## II. LITERATURE REVIEW

Requirements problems are broadly mentioned to affect and reduce the quality of software and the effectiveness of the software development process [9]. It is estimated that correcting the requirements late can cost up to 10 - 200 times as much as correcting the errors during the requirements [3]. There has been a failure in many software projects because requirements were poorly specified and researchers made observations that many analysts with limited knowledge of the problem domain result in requirements problems, poor quality, and cost overruns. It is important to note that, requirements engineering is a particularly critical stage of the software process or project and thus the first and foremost step in software development. Requirements engineering as defined in [10] is “the process of understanding and defining what services are required from the system and identifying the constraints on the system’s operation and development”. Requirements engineering is a repeatable and systematic method such that in every stage of the requirements engineering lifecycle, the requirements are analysed and evaluated so as to discover the consistency and completeness of the requirements [11] as cited in [1].

A typical requirement engineering process normally consists of Elicitation, Analysis, Specification, Validation/Verification and Management, which are arranged in order of execution and performed in an interleaved manner as shown in Figure 1.



Fig. 1 Requirements Engineering Phases

Requirements elicitation is the procedure of gathering the requirements where the technical professionals such as software developers and system engineers, in an organization work together in a close relationship with the users of the system and the customers to determine more about the problem to be solved and to bridge the gap between the stakeholders and the developers. References [11] and [12] (as cited in [3]) have stated, “Requirements elicitation is the first step of the process, through which a project team determines the organizational needs and wants that need to be addressed with the aid of the project effort”. It is imperative to note that the requirements elicitation process emphasizes on “what” ought to be expected to accomplish by the anticipated system irrespective of “how” it is to be accomplished [3]. In [13] requirements elicitation is defined as “the process of identifying software or system requirements from various sources through interviews, workshops, workflow and task analysis, document analysis, and other mechanisms”. Good requirement elicitation process consists of object setting, background knowledge possession, knowledge organization and finally stakeholder requirements collection [11].

Requirements elicitation is performed using a number of techniques that are dependent on the condition in which they are applied. This is due to the fact that different stakeholders have distinct ways to store, recognize and express their knowledge and understanding of the problem and a single technique is not sufficient to obtain requirements from different stakeholders. Loucopoulos & Karalostas (as cited in [3]) have stated that “the importance of requirements elicitation cannot easily be overestimated; when you have to solve somebody else’s problem the first thing you have to do is to find out more about it”.

## III. COMPARATIVE STUDY

This section, which is the comparative study, is sub-divided into two main sections. The first section briefly discusses the requirement elicitation techniques with comparisons. The other section explains the software tools that support elicitation techniques mentioned.

### A. Requirement Elicitation Techniques

Requirement elicitation techniques are used to collect requirements from the users in order to assist the developers to understand the requirements of users and stakeholders. Requirements development is a thorough interaction procedure between stakeholders and analysts. The elicitation techniques are categorized into four main groups. This is in accordance to the means of communication with stakeholders. The groups are conversational, observational, analytic and synthetic. Each elicitation requirement group presents a unique communication model between analysts and stakeholders, and thereby reflects the nature of a method. Understanding the method category or groups helps analysts/engineers understand various elicitation methods,

which advises them to select an appropriate set of the method(s) for requirements elicitation. Every requirement elicitation technique has its own strengths and weaknesses and each phase mostly uses two or more techniques, known as a hybrid, at a time for gathering requirements.

1) *Traditional Techniques:* This technique has been earlier used since the beginning of software engineering process and is also the most commonly used technique for requirements elicitation, and therefore termed as traditional technique. This technique includes Interviews, Questionnaires/Surveys, Document analysis, and Introspection.

- **Interviews:** Interviews are the first and most important requirement elicitation technique in software development life cycle where the analysts and the engineers of the requirements engineering process deliberate with the various stakeholders so as to understand the requirements of the system and the objective they have to accomplish within the system [13][1].
- **Surveys:** Surveys are used to acquire and analyse information from a variety of people and at different locations that results in huge data and these are used for collecting requirement for general-purpose software. Surveys, once planned accurately and shrewdly, the data gathered will be of good quality and these are inexpensive [14].
- **Questionnaires:** Questionnaires are easy and typically carried out to competently acquire information from a vast number of people at a cheaper cost in less time. The questionnaire needs to be well designed so as to get explicit and reliable information from the stakeholders about themselves and the organisational environment [14].
- **Document Analysis:** Document Analysis, sometimes referred to as Domain Analysis is [14] when a present system needs to be replaced with a new system by gathering and examining suitable requirements and documentation such as instruction manuals, templates, and others from the previous system.
- **Introspection:** Introspection is used to start requirements elicitation process. This can then be used together with other techniques when the analyst effectively acquires requirement based on previous experience and familiarity with domain and business process, which is useful for the user or other stakeholders [7].

2) *Contextual Techniques:* These are techniques that gather requirements at the office of the end user/stakeholders. This technique is sometimes known as Observational

technique by some researchers. Requirements are elicited at the working environment where the new system will be used. The requirement elicitation techniques in this category are Observation/Social Analysis and Ethnography.

- **Observation:** Observation, known as Social Analysis is a widely used technique. It requires the analyst to watch the stakeholder's working place first-hand and take notes on the activities without interfering in their work so as to understand and interpret their actions [14].
  - **Ethnography:** Ethnography requires the analyst to participate in the activities of stakeholders with different background actively or passively to collect data for the system being developed [14]. This technique can be used where the new system is the solution to problems with procedures and processes of the previous system
- 3) *Collaborative/Group Techniques:* Group elicitation technique is where a certain group of stakeholders use their individual know-how on a specific issue and agree upon a set of decisions. It generally involves professionals from different subjects who have been given equal powers to provide their ideas regarding a certain system. These techniques include Prototyping, Joint Application Development (JAD), Brainstorming, Use Cases/Scenarios, Requirements Workshop and Group Work/Focus Group. These group techniques are said to be better because ideas generated from all the stakeholders with distinctive backgrounds, greater skills and knowledge is involved in making decisions.
- **Prototyping:** Prototyping is when the stakeholders are unaware of the solutions available. As long as the requirements are uncertain, a dummy version of the product is used so as to obtain requirements and this is refined frequently according to stakeholder's feedback [15].
  - **Joint Application Development (JAD):** Joint Application Development (JAD) helps in gathering information from stakeholders and making decisions within a very short time through conversation when the need arises. This technique is a well-structured approach where the roles and actions of the participants and also the goals of the system are pre-defined [15][14].
  - **Brainstorming:** This is defined in [16] as an "informal discussion where free expression of ideas is given to every participant for a new kind of system to be developed, which promotes participants that usually belong to different stakeholder group to generate as many ideas as possible". The ideas generated are normally recorded where inappropriate ideas are eradicated and the suitable ones are

prioritized. Brainstorming helps in finding new ideas to solve previous problems.

- **Use Cases/Scenarios:** Use cases/scenarios represent stakeholders' interaction with the new system, which is a real world example of how a system will be used [16]. Once the initial requirements are collected, use cases or scenarios can be used and this involves a complete explanation of all the processes in a diagrammatic form, such as starting state, flow of events, parallel activities, and end state. Use cases/scenarios are only useful when a system needs to be described from the user or layman's viewpoint using a simple natural language.
  - **Requirements Workshops:** Requirement workshops are used to discover new requirements of a system where stakeholders have structured and well-planned small meetings so as to discuss, improve and endorse their ideas. This kind of workshop is generally suitable for large and complex projects that should be led by an expert facilitator for its success [14].
  - **Focus Group:** A focus group, sometimes called group work which consists of a group between four to nine professionals from distinct backgrounds and with diverse skills discuss in a free form and concerns about the features of a system that will be created [14]. A focus group helps to pinpoint user's expectations from the system by indicating what is important and also what to expect from the system. This often brings out spontaneous answers and ideas that are more precise, beneficial and significant.
- 4) *Cognitive Techniques:* These allow examining and gathering information up to the level of human thinking. It involves understanding the problem comprehensively according to stakeholder's point of view. The elicitation techniques in this category are Card Sorting, Laddering, Repertory Grids, Protocols Analysis and Class Responsibility Collaboration.
- **Card Sorting:** Card-sorting technique requires that the requirement specified is well understood by the analyst and stakeholders. In this technique, the names of domain entities are written on cards and stakeholders are required to arrange them based on their knowledge and understanding which helps in grouping and association of user requirements. Card sorting is usually conducted as a specific activity in the early design phase of a project, but this can also be used during a product evaluation so as to determine if usability issues are due to problems with grouping or group labels [14].
  - **Laddering:** Laddering is a form of planned interview in which limited sets of standard questions arranged in hierarchical order, are questioned to the

stakeholders. This technique depends on the domain knowledge of the stakeholders in order to be a success [15][14].

- **Repertory Grid:** Repertory Grid, known as Repository grid is an elicitation technique of developing a grid in a matrix form that saves the requirements involved by developing attributes and assigning values to a set of domain entities by asking the stakeholders. It is a good technique to provide the variations such as similarities and differences between different information domains [15][14].
- **Protocol Analysis:** Protocol analysis is a technique of gathering all stakeholders and analysts so as to discuss the requirements of the system [14]. The user/stakeholder is observed while he is occupied with work and he clarifies his thoughts and opinions regarding that at the same time. Problem solving at an individual level will be understood easily. Protocol analysis additionally gives the necessary actions to be taken so as to accomplish user requirements with reasoning.
- **Class Responsibility Collaboration:** Class Responsibility Collaboration (CRC) [16] includes a collection of cards representing a system where each individual card is divided into *Class*, *Responsibility* and *Collaboration*. The cards are organised according to their functionality and relationships with other classes. CRC represents software requirements where each card is a particular class. Each class is allotted tasks and responsibilities to process user requirements as shown in Figure 2.

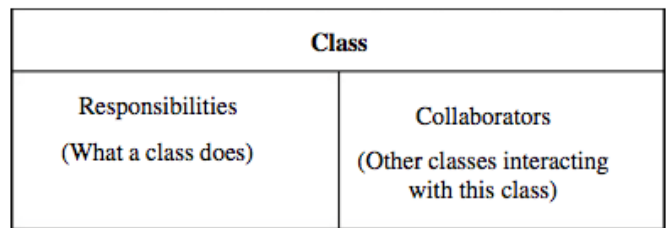


Fig. 2 CRC Card Structure

- 5) *Comparison of Requirement Elicitation Techniques:* This section compares various requirement elicitation techniques with a focus on the strengths and weaknesses of each technique. It can be observed that every elicitation technique in Tables I, II, III and IV has some strengths and weaknesses.

TABLE I  
COMPARISON OF TRADITIONAL ELICITATION TECHNIQUES

Elicitation Techniques	Strengths	Weaknesses
Document Analysis	Helpful during a dearth of stakeholders and users.  Useful historical data is	Takes too much time to find information from a huge amount of

	provided	documentation.
	Useful to frame questions for interviews	Valid information may be out-dated.
	Used for requirement reuse	Periodic update of documentation is required
	Inexpensive technique	Information might be incomplete
Questionnaires/ Surveys	Takes a short time to reach a large number of people.	Difficult to get further clarification of the problem from the user by the analyst.
	Useful when the same set of questions is asked to a vast audience.	Questions can be misconstrued.
	It is cost-effective.	Useful feedback not received.
	Very simplified because questions are either multiple choice, true/false or fill in the blanks.	Question ambiguities may surface.
Introspection	Implementing this technique is practically free.	Difficult for analysts to visualise the environment in which the new system works.
	Easy to implement.	
	It can act as a good first step to begin requirements elicitation	It does not allow discussion with stakeholders. Therefore, it is discouraged to combine with other techniques.
		Analysts and stakeholders need to be knowledgeable about the domain.
Interviews	Good for a complex topic.	Small number of audiences involved.
	Provides rich information that is good at eliciting detailed requirements.	It is hard to interview stakeholders at an appropriate time.
	Ambiguities are elucidated.	Quality of data gathered depends on the talents of the interviewer. The respondents may be affected if the interview is biased.
	No answer distortion by the person being interviewed as he is being vetted in person.	
	An interviewer can analyse emotions or some kind of discomfort triggered to the interviewee from the questions asked.	Information needs to be clarified by having some follow-ups especially when participants may not recall information properly.
	Interviewee's body language is easy to read.	
	Non-responsiveness from the interviewee remains low.	They are effortful and time-consuming.

TABLE II  
COMPARISON OF CONTEXTUAL ELICITATION TECHNIQUES

Elicitation Techniques	Strengths	Weaknesses
Observation/Social Analysis	Authentic and reliable because analysts observe the environment.  Requirements collected through other methods can be useful when confirmed and validated.  Gives an idea of how users will interrelate with the system.  Helpful in timing the duration	Multiple sessions may be required to check all the requirements.  Users can be aloof if during active observation, are interrupted by questions.  It is hard for the analyst to understand why some decisions are made during passive observation.  It is time-consuming.
Ethnography	Takes a short time to uncover certain features of a work place.  Helps fathom how people work an interrelate with each other in an organization.  Need fewer resources to be effective.  Reveals serious events unobserved by other techniques.	A skilled ethnographer is required as there is a lack of a comprehensive guide on how to implement an ethnographic technique effectively.  Unique new features added to the system may not be noticed.  Having a diverse population may lead to non productive of desired results.  Concentrations primarily on end-users.  Can be time consuming.
Joint Application Development (JAD)	The time and cost of requirements elicitation are minimised.	Time and resources wasted if not properly planned.  Needs proficient

TABLE III  
COMPARISON OF COLLABORATIVE/GROUP ELICITATION TECHNIQUES

Elicitation Techniques	Strengths	Weaknesses
Prototyping	It requires the involvement of a user during the development process.  It permits early user feedback for requirements improvement.  It reduces the time and cost of development.  It enables a better understanding of the system by users and analysts.	It allows users to resist changes when they are used to a particular system.  It may increase effort and cost as calculated earlier.  It consumes time for complex systems.

Brainstorming	The design of the system is hastened.	facilitators.		which leads to results being biased.
	Encourages feedback by users.	Needs lots of preparation and work.	User Scenarios	It is hard to draw useful scenarios.
	More satisfaction by stakeholders.	This technique is expensive.		It is highly unsuitable for all types of projects even if more requirements are captured.
	It enables stakeholders, analysts and other professionals to have good communication.			Not all the processes of the future are completed.
	Visual aids and case tools make the session interactive when used.			
	Inexpensive and minimal resources are needed.	It is inappropriate to resolve major problems.		
	Participants can be amateurs and they each have to take part actively in the process.	If not organized properly can be time-consuming.		
Requirements Workshop	It is clear and easy to implement.	Quantity and quality of ideas are not always equal.		
	Assists in generating new ideas.	Lack of proper stakeholders participation can lead to duplication of ideas.		
	Aids in resolving conflicts.	Extroverts may take over all the session and all the time-sharing their ideas while introverts will be afraid to take the time-sharing their views.	Laddering	Difficult to maintain requirements hierarchy during addition or deletion or user requirement.
	Each participant is permitted to speak and share ideas.			Technique becomes complex when the number of requirements is enormous.
	Takes a shorter time to gather requirements.	It is challenging to get stakeholders and analyst together at the same time and place.		It is a demanding technique.
	Cheaper than having interview gatherings.	A vast number of participants may make the process to slow down.	Card Sorting	Unsuitable for difficult, large and diverse designs.
	Fast response.	Limited number of participants may lead to requirements that are not complete and does not denote the right set of user requirements.		It contains mutable results.
Group Work/Focus Groups	Assists analysts and stakeholders to trust and understand each other.			Useful when the number of cards is restricted.
	Participants work as a team to produce a particular set of requirements.			Only shallow features are included.
	Quality requirements are acquired in a short time.	It is quite tedious to gather all the stakeholders to the same place at the same time due to their demanding schedule.		Limited collaborations and thorough explanations lower the value of the technique.
	For a same number of people, this technique is cheaper than conducting interviews.	Participants may have trust issues and may be hesitant to discuss critical or sensitive matters.	Repository Grids	Lots of effort is needed on the part of experts and analysts.
		Members may get manipulated by assertive people in the meeting		It is consumes too much

TABLE IV  
COMPARISON OF COGNITIVE ELICITATION TECHNIQUES

Elicitation Techniques	Strengths	Weaknesses
Laddering	Easy to understand requirements because of hierarchical nature.  Requirements reuse saves time and cost.  Not suitable for building a new system	Difficult to maintain requirements hierarchy during addition or deletion or user requirement.  Technique becomes complex when the number of requirements is enormous.  It is a demanding technique.
Card Sorting	It is quick and cheap.  People that are geographically remote can access and participate through the internet.  It is an easy trustworthy recognised technique.  Useful in providing good organisation of information to be fed into the information process.  Aids in collecting qualitative real data from the users.	Unsuitable for difficult, large and diverse designs.  It contains mutable results.  Useful when the number of cards is restricted.  Only shallow features are included.  Limited collaborations and thorough explanations lower the value of the technique.
Repository Grids	Distinguishes between dissimilarities and similarities of components unknown to experts.	Lots of effort is needed on the part of experts and analysts.  It is consumes too much

	Reduces expert's prejudice while he acquires an understanding of domain from user's perspective	time.
Class Responsibility Collaboration (CRC)	Replaces use of UML diagrams.	Software requirements are limited
	Classes and collaborations are described using natural languages.	CRC cards are not used by software engineers but by designers.
	Users and experts work as a team to design the model.	Not appropriate for large systems.
	Experts analyse and develop the model ensuring that the right model is built.	Creating cards in bulk is difficult and time consuming.
Protocol Analysis	Techniques can be implemented easily and are cheaper in cost.	Very unsuitable for projects with tight plans due to time consumption.
	It can be used at any phase of project development.	Users need to clarify their actions and decisions every time.
	Analyst's thorough understanding about product domain is usefully gained.	It is suitable for circumstances where an individual can portray in his mind the whole situation of the domain.
	It provides how the system will work in real life	

6) *Evaluation of Requirement Elicitation Technique:* This section describes a comparative evaluation of the requirement elicitation techniques earlier discussed in section III-A based on some parameters in [2], which is an essential part when it comes to the project characterization and also choosing the right elicitation technique. These parameters are:

- Project size is quantified in terms of the number of requirement or functionality in passes. It can be of three types: small (<100), medium (100 - 1000) and large (>1000).
- Project complexity can be calculated by factors such as the number of components involved, number of technology required, management complexity and the diversity of stakeholders. It can be divided into four types: low, medium, high, very high.
- Requirement volatility is defined as the number of changes in requirement over the period of time, which is calculated by analysing the domain of the

problem. It can be of three types: small, medium, and large.

- Time & cost constraint is defined as the time threshold available for project completion. It can be – highly time rigid, flexibility in time constraint. Cost can be defined as the total budget of the project. If the budget is high then more money can be invested for the requirements engineering process. It can be low, medium, high, and very high.
- Stakeholders' availability is defined as the level of heterogeneous of stakeholders. It can be of three types: low heterogeneous, medium heterogeneous, and high heterogeneous.
- Domain knowledge is defined as the knowledge and understanding of the software project domain. It can be – low, medium, high.

Table V shows the evaluation between the parameters mentioned above and the various requirement elicitation techniques mentioned in section III using the following as metrics: low (L), medium (M), high (H), small (S), large (G), and very high (V). This helps the analysts in making the right decision in choosing the appropriate elicitation technique to use for the software development.

#### B. Requirement Elicitation Tools

There are many tools that have been developed so as to support requirements elicitation that is available in literature and on the Internet, commercially or laboratory version. Each state-of-the-art tool has its own features and limitations and has functionalities that can create, retrieve, update and remove requirements and link requirements to their source [17]. Some of the tools to be discussed here are Rational RequisitePro, Dynamic Object Oriented Requirements System (DOORS), TopTeam Analyst, Objectiver, CaseComplete, Analyst Pro, and Generic Model Approach to Requirements Capture (GMARC).

- **Rational RequisitePro:** The Rational RequisitePro is developed by IBM and is mostly used by software development teams to manage their requirements so as to minimize rework and also to maximize the quality. It supports Microsoft Word for communication and is complemented with a relational database [15][18][17].
- **DOORS:** Dynamic Object Oriented Requirements System (DOORS) was originally developed by Telelogic but was later acquired by IBM Rational in 2008. It is used as document management software which helps to reduce costs, increase efficiency and improve quality by enabling the analyst to optimize requirements communication, collaboration and verification [18][17].

TABLE V  
EVALUATION OF ELICITATION TECHNIQUES

Category	Elicitation Technique	Parameters					
		Domain Knowledge	Project Size	Project Complexity	Requirement Volatility	Time & Cost Constraint	Stakeholders' availability
Traditional	Document Analysis	M	M/G	M/H	M	M/H/V	M/H
	Interviews	M/H	S/M/G	M/H/V	M/H	L	M/H
	Introspection	M	S/M	L/M/H/V	M/H	L/H/V	M/H
	Questionnaires	M	M	H/V	L/M	H/V	M/H
	Surveys	H	G	L/M	M	L	L
Contextual	Observation/Social Analysis	L	S/M/G	H/V	L	L	L
	Ethnography	L/M	M/G	H/V	L	L	L
	Prototyping	M	S	H/V	H	L	M/H
Collaborative/Group	Joint Application Development (JAD)	H	M/G	M	H	M	M/H
	Brainstorming	M	M/G	M/H	L/M	L	H/M
	Requirements workshop	L/M	M/G	M	L	L	L
	Group work/Focus Group	H	S/M/G	H/V	L	L	H
	User Scenarios	M	S/M/G	M/H/V	M/H	M	M/H
	Laddering	M/H	S	M/H	M	L/M	M
Cognitive	Card Sorting	H	M	H	M/H	L/M	M
	Repertory Grid	M/H	S	L/M	L/M	M/H	M
	Class Responsibility Collaboration (CRC)	M/H	M/G	M/H	M/H	H/V	M/H
	Protocol Analysis	M	M	M	M/H	L	M/H

- **Top Team Analyst:** Top Team is a solution for requirements gathering and management that supports use cases and test cases to support the requirements specification process. It has a powerful database for all requirements artifacts such as Screen Prototypes, and Documents [17].
- **Objectiver:** Objectiver was developed by Cediti and designed by practitioners to enable real requirements engineering. It allows stakeholders to take a universal synopsis of the system and all the prototypes representing the system. The tool has a chance to draw diagrams, define system concepts and also the relationship between those concepts. The defined models and its diagrams are explained using a text document, which helps in gathering system requirements [15][18].
- **Case Complete:** This is developed by Serlio Software to manage, share use cases and requirements based on Microsoft Word. Case Complete helps to write use cases and requirements quicker and easier that have excellent compliance with use case standards for a novice or an expert user whether working on a solo system or a part of a diverse team [18].
- **Analyst Pro:** This is a tool for requirements management, tracing and analysis developed by Goda Software Inc. With Analyst Pro, requirements can be traced with any lifecycle software model such as waterfall, RUP, spiral. It also provides integrated configuration management to simplify the development process [18].
- **GMARC:** Generic Model Approach to Requirements Capture (GMARC) tool developed by Computer

System Architects Ltd. integrates a fully developed Requirements Engineering Methodology and provides rapid elicitation of requirements using a generic approach to increase re-usability and encourage standardization across projects [15][18][17].

1) **Assessment of Requirement Elicitation Tools:** Each of the requirement tools mentioned in the previous section has its own features and functionalities such as automatic link detection, automatic link creation and change, coverage analysis support, documentation support, graphical representation and tools integration [15]. The functionalities to be assessed are briefly explained as follows:

- **Glossary** describes the different terms used in the project so as to reduce misunderstandings in the requirement management phase.
- **Template** is defined as user requirement in a structured format like a table written in natural language. The development team normally completes templates where its structure is predefined.
- **Tool Integration** incorporates one tool to another that helps project team to manage their requirement stored so as to make them visible and linkable and also to improve the traceability and to reduce rework and to increase the quality.
- **Graphical Representation** is user-friendly where the requirements can be easily recognized and data can be easily compared and this helps to save time and budget. The requirements can be represented using matrices, trees or graphs.
- **Documentation Support** describes the requirement



that is generated by the tool into a standard document for the requirement management. The complete development process is managed with the help of requirement documentation.

- **Checklist** verifies the requirements and ensures that the requirements are complete, accurate and reliable.
- **Scalability** accommodates a large number of user requirements according to the project's needs.

Table VI below shows the analysis of the software requirement tools. The symbol (✓) shows that the functionality exists in the tool whereas the symbol (X) shows that the functionality does not exist in the tool.

#### IV. DISCUSSION

Requirements elicitation is a very crucial phase in any software development life cycle where the requirements have to be accurate, complete and reliable. Although there are many software requirements tools that are available for

use in the elicitation, design and management stages, this paper assesses the requirement elicitation techniques and software requirement tools using various parameters and functionalities. The requirement elicitation techniques were grouped according to communication and were compared using their strengths and weaknesses as shown in Tables I, II, III and IV and also some parameters such as domain knowledge, project size and complexity, requirement volatility, time and cost constraint and stakeholders' availability as shown in Table V. In requirement elicitation phase, two or more techniques can be used at the same time for gathering requirements. Requirement tools used for the assessment are RequisitePro, Objectiver, CaseComplete, AnalystPro, DOORS and GMARC. Table VI shows the assessment of the tools based on Glossary, Templates, Link creation and change, Tool Integration, Documentation Support, Graphical Representation, Checklist and Scalability parameters.

TABLE VI  
ASSESSMENT OF REQUIREMENT TOOLS

Tools	Glossary	Templates	Link Creation & Change	Tool Integration	Documentation Support	Graphical Representation	Checklist	Scalability
Rational	X	✓	X	✓	✓	✓	X	X
RequisitePro	✓	✓	X	✓	✓	✓	✓	X
Objectiver	✓	✓	X	X	✓	✓	X	X
CaseComplete	X	✓	X	✓	✓	✓	X	✓
AnalystPro	X	✓	✓	✓	✓	✓	X	✓
DOORS	X	✓	✓	✓	✓	✓	X	✓
GMARC	X	✓	X	X	✓	✓	X	X
TopTeam Analyst	X	X	X	X	X	X	X	X

#### V. CONCLUSION

In this paper, a review of various requirements elicitation techniques and tools were presented. Comparing the various elicitation techniques determines that every technique has its strengths and weaknesses. Every technique depending on its features is used at a certain stage of the requirements phase and also for eliciting different types of requirements in the software development cycle. Some of these techniques are used at initial stages while some are used at a later stage and some for rapid eliciting requirements and some for rudimentary requirements. It is important for system analysts to be properly acquainted with the details of elicitation techniques so as to appropriately select and implement the right technique(s). It is important to note that although using a certain number of techniques will ensure that requirements are gathered at their maximum, however, using an excessive number of resources will reduce the project's efficiency. Thus there is a need for evaluation of these techniques to check which is more proficient and applicable in eliciting requirements.

#### REFERENCES

- [1] S. G. Gunda, "Requirements Engineering: Elicitation Techniques," Department of Technology, Mathematics and Computer Science, University West, Sweden, 2008. Retrieved from <http://www.diva-portal.org/smash/get/diva2:215169/fulltext01>.
- [2] S. Tiwari and S. S. Rathore, "A Methodology for the Selection of Requirement Elicitation Techniques," *arXiv:1709.08481 [cs]*, Sep. 2017. Retrieved on 10<sup>th</sup> June 2018, from <https://arxiv.org/pdf/1709.08481.pdf>.
- [3] O. I. Al Mrayat, N. M. Nowawi and N. Basir, "Requirements Elicitation Techniques: Comparative Study," *International Journal of Recent Development in Engineering and Technology*, vol. 1, no. 3, pp. 1-10, 2013.
- [4] B. Davey and K. R. Parker, "Requirements Elicitation Problems: A Literature Analysis," *Issues in Informing Science and Information Technology*, vol. 12, no. 1, pp. 71-82, 2015. Retrieved from <http://iisit.org/Vol12/IISITv12p071-082Davey1929.pdf>.
- [5] R. Raisinghani, R. Riggen, and K. Ryan, "Adopting an Agile Methodology Requirements-Gathering and Delivery". PricewaterhouseCoopers, 2014. Retrieved on 7<sup>th</sup> June 2018 from [https://www.pwc.com/en\\_US/us/insurance/publications/assets/pw-c-adopting-agile-methodology.pdf](https://www.pwc.com/en_US/us/insurance/publications/assets/pw-c-adopting-agile-methodology.pdf).
- [6] A. Srivastava, "Top 6 characteristics of Good Requirements and SRS". [Online]. Available: <https://www.linkedin.com/pulse/top-6-characteristics-good-requirements-srs-abhishek-srivastava>. [Accessed: 17-Jun-2018].
- [7] D. Zowghi and C. Coulin, "Requirements Elicitation: A Survey of Techniques, Approaches, and Tools", in *Engineering and*

- Managing Software Requirements*, A. Aurum and C. Wohlin, Eds. Berlin/Heidelberg: Springer-Verlag, 2005, pp. 19–46. Retrieved on 31<sup>st</sup> May 2018, from <https://web.eecs.umich.edu/~weimerw/481/readings/requirements.pdf>
- [8] R. R. Nelson, “IT Project Management: Infamous Failures, Classic Mistakes, and Best Practices”, *MIS Quarterly Executive*, vol. 6, no. 2, pp. 67–78, 2007.
- [9] B. Solemon, S. Sahibuddin and A. Abd Ghani, “Requirements Engineering Problem s in 63 Software Companies in Malaysia,” *International Symposium on Information Technology*, vol. 4, pp. 1-6, 2008.
- [10] I. Sommerville, *Software Engineering*, 9<sup>th</sup> ed., Boston, Massachusetts: Addison Wesley, p. 790.
- [11] G. Kotonya and I. Somerville, *Requirements Engineering: Processes and Techniques*. England: John Wiley & Sons Ltd, 1998.
- [12] B. Nuseibeh and S. Easterbrook, “Requirements Engineering: A Roadmap”, 2000, pp. 35–46.
- [13] M. Tariq, S. Farhan, H. Tauseef and M. A. Fahiem, “A Comparative Analysis of Elicitation Techniques for Design of Smart Requirements using Situational Characteristics,” *International Journal of Multidisciplinary Sciences and Engineering*,” vol. 6, no. 8, pp30-38, 2015. Retrieved on 17<sup>th</sup> June 2018 from <http://www.ijmse.org/Volume6/Issue8/paper6.pdf>
- [14] S. Tiwari, S. S. Rathore, and A. Gupta, “Selecting Requirement Elicitation Techniques for Software Projects”, 2012, pp. 1–10. DOI (Boukhari and Siddiqui 2010): 10.1109/CONSEG.2012.6349486.
- [15] M. A. Abbasi, J. Jabeen, Y. Hafeez, D.-B. Batool, and N. Fareen, “Assessment of Requirement Elicitation Tools and Techniques by Various Parameters”, *Software Engineering*, pp. 7-11. DOI: 10.11648/j.se.20150302.11.
- [16] M. Yousuf and M. Asger, “Comparison of Various Requirements Elicitation Techniques,” *International Journal of Computer Applications*, vol.116, no. 4, pp. 8-15, 2015. Retrieved on 17<sup>th</sup> June 2018 from [http://www.academia.edu/12160700/Comparison\\_of\\_Various\\_Requirements\\_Elicitation\\_Techniques](http://www.academia.edu/12160700/Comparison_of_Various_Requirements_Elicitation_Techniques)
- [17] M. Shahid, S. Ibrahim and M. N. Mahrin, “An Evaluation of Requirements Management and Traceability Tools,” *International Journal of Computer and Information Engineering*, vol. 5, no. 6, pp. 627-632, 2011. Retrieved on 10<sup>th</sup> July 2018 from <https://waset.org/publications/15009/an-evaluation-of-requirements-management-and-traceability-tools>
- [18] M. U. Bokhari and S. T. Siddiqui, “A Comparative Study of Software Requirements Tools for Secure Software Development,” *BVICAM's International Journal of Information Technology*, vol. 2, no. 2, pp. 207-216, 2010