# Enhanced Framework for Big Data Requirement Elicitation

Aya Hesham[1], Osama E. Emam[2]
Department of Information Systems
Faculty of Computers and Artificial Intelligence
Helwan University
Cairo, Egypt

Marwa Salah[3]
Department of Information Systems
Faculty of Computers and Artificial Intelligence
Helwan University
Faculty of Informatics and Computer Science, British
University in Egypt, Cairo, Egypt

***Abstract*—Requirement engineering is one of the software development life cycle phases; it has been recognized as an important phase for collecting and analyzing a system's goals. However, despite its importance, requirement engineering has several limitations such as incomplete requirements, vague requirements, lack of prioritization, and less user involvement, all of which affect requirement quality. With the emergence of big data technology, the complexity of big data, which is defined by large data volume, high velocity, and large data variety, has gradually increased, affecting the quality of big data software requirements. This study proposes a framework with four sequential phases to improve requirement engineering quality through big data software development. By integrating the proposed framework's phases in which user requirements are collected in a complete vision using traditional requirement elicitation techniques with agile methodology and mind mapping, the collected requirements are displayed via a graphical representation using mind maps to achieve high requirement accuracy with connectivity and modifiability, enabling the accurate prioritization of requirements implemented using agile SCRUM methodology. The proposed framework improves requirement quality in big data software development, which is represented by accuracy, completeness, connectivity, and modifiability to understand the value of the collected requirements and effectively affect the quality of the implementation phase.***

***Keywords*—Requirement engineering; big data requirement; agile methodology; mind mapping**

## I. Introduction

Software requirement engineering represents business needs and goals, including functional and nonfunctional dependency competencies that must be represented and achieved. Several issues in the requirement engineering life cycle have contributed to a high failure rate for software engineering projects, such as the lack of comprehensive requirements, including unclear, incomplete, and inaccurate requirements; requirement conflict, leading to insufficient estimation of social and technological requirements; the lack of customer involvement, leading to customer dissatisfaction; and concurrent changes in requirements [1]. Nowadays, the development of big data software applications is prevalent. According to big data technology market analysis [2], "The big data market size is projected to grow from USD 138.9 billion in 2020 to USD 229.4 billion by 2025, at a compound annual growth rate of 10.6% during the forecast period." With the increasing growth of data, defining the term "big data" has become more challenging when considering big data characteristics such as volume, which refers to the quantity of data generated; velocity, which refers to the speed at which data are generated; and variety, which refers to different types of data generated (e.g., text, documents, video, audio, and images). Consequently, requirement challenges have increased because representing a huge volume of data and determining the exact time that data will be received and how long they will take to arrive are essential factors to consider. Additionally, it is critical to determine the type of data received because each data type has its customization, so considering these characteristics in the requirement election phase became more challenging [3]. Big data characteristics pose severe challenges to achieving software requirement quality standards for security, performance, scalability, privacy, and other quality requirements. Additionally, how to systematically handle quality requirements involving big data characteristics to better understand the requirements of big data software projects is a challenge [4]. A well-planned framework for the requirement elicitation process can mitigate the negative effect of big data software requirement limitations. Mind mapping provides the best practice in requirement election representation on the basis of its graphical concept, i.e., mapping the main ideas together to obtain the best value, producing an accurate and clear requirement representation while considering big data characteristics and quality attributes, which greatly aids in obtaining well-prioritized requirements. Nowadays, agile SCRUM methodology has gained popularity because of its properties, such as flexibility, which can handle the technical issues mentioned above, as it handles changes in requirements, customer involvement, and satisfaction, besides documenting the requirements in the product backlog [5]. Although agile is the commonly used methodology in big data projects because of its resilience in accepting new and changes in requirements during the implementation process, big data project needs more optimized methodologies to deal with the massive changes of requirements especially in big data project considering its characteristics. So, by the integration with mind mapping, it'll help develop a prioritized product backlog, as agile SCRUM is the initial requirements specification document [6].

In summary, focusing on the entire process of requirement elicitation for big data software projects, not only on better requirements but also on the quality of the elicited requirements, is an excellent way to set the stage for accurate software development.

This paper proposes a framework for requirement elicitation that comprised four phases: a collection phase, mind mapping phase, prioritization phase, and agile phase, all the phases were explained in Section IV. The integration of these phases provides a complete and accurate path for requirement gathering that will help a system analyst team elicit, analyze, and manage requirements for a big data software project. The framework serves the requirement elicitation and analysis phase in the big data projects development life cycle, and it promotes the achievement of high-quality big data requirements during this stage. It's necessary to place a strong emphasis on achieving quality factors that are highly supportive of the big data characteristics of completeness, correctness, connectedness, and modifiability in order to meet the needs for large volume, velocity, and variety of data.

The following parts are organized into five sections. The background of this research is presented in Section II. In, Section III, the literature review is presented. In Section IV, the methodology is provided. The framework implementation is discussed in Section V. Conclusions are presented in Section VI.

## II. Background

This research is an integration of different parts, including requirement elicitation, big data, agile methodology, and mind mapping, and each part will be introduced in the following sections.

### A. Requirements Elicitation

First, requirement elicitation is considered to be one of the most critical activities in the software development industry [7]. It defines a project scope by gathering requirements, which are considered stakeholders' needs, and merging this information to produce a meaningful and understandable method for developing a system that will meet these needs.

Several empirical studies have discussed requirement engineering challenges in the software development life cycle [8]. These challenges have been traced and collected under two main aspects, which are customer and system aspects, as shown in Fig. 1. The two aspects are interdependent, as challenges from the customer aspect, such as a lack of understanding of users' needs, customer collaboration, and a common language, lead to challenges in the system aspects, such as requirement changes and updates and a lack of accurate documentation, requirement quality, and requirement prioritization.

The first step in eliciting requirements from a customer is to understand what the customer wants because interactions with customers typically occur in a natural language, which makes it difficult to obtain complete and clear requirements, and the second step is to engage with the customer to gain a better understanding of the elicited requirement; however, because of the lack of customer collaboration, understanding customer needs and requirement clarification is hampered [9]. Changing requirements during the development process causes loss of system requirement objectives, increases cost and time, and necessitates a continuous update to the stored requirement documentation. Additionally, storing requirements in the form of user stories in a system backlog is not an appropriate solution to fully understand the system requirement and track updates on it according to the agile development life cycle [10]. Prioritizing requirements according to specifics factors such as business value, risk, importance, cost, dependency, and constraints becomes more challenging in a complex system [8]. The above challenges significantly affect requirement quality, especially when dealing with complex projects such as big data projects, as we have discussed in this study. Hence, the main objective of this study is to improve requirement quality, thereby improving the functionality and services provided to end-users.

### B. Requirement Engineering and Big Data Projects

With the ubiquity of emerging technologies such as big data and the Internet of Things, the processes of software engineering, such as requirement gathering, design, implementation, and testing, must be evolved and improved to apply these new technologies [11].

Specifying the requirement engineering processes must be improved in the context of big data projects, as the process complexity has increased because of the existence of several dynamic components, such as distributed networks, databases, business intelligence layers, middleware, and computation node, making requirement election in this complex distributed environment extremely difficult [12]. Additionally, data scientists and software engineers face difficulty when measuring and determining the maximum value of big data.

The main big data characteristics include volume, velocity, and variety. Volume is the size of data, which are generated from different data sources. Velocity is the data speed. Variety is the data type, including structured, semi-structured, and unstructured data, such as text, image, audio, and video. Recent research has claimed that the above-mentioned big data characteristics are the most effective factors that affect the quality of big data projects because it is extremely difficult to provide system requirements with high quality standards, such as performance, security, accuracy, availability, and reliability [13].

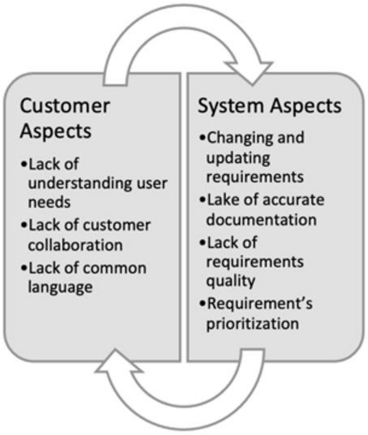
Customer Aspects
•Lack of understanding user needs
•Lack of customer collaboration
•Lack of common language
System Aspects
•Changing and updating requirements
•Lake of accurate documentation
•Lack of requirements quality
•Requirement's prioritization

Fig. 1. Requirement Engineering Challenges.

### C. Agile and Big Data Projects

A Project Management Institute Report stated that more than 70% of companies have adopted an agile approach, concluding that agile projects are 28% more effective than traditional ones [14]. The requirement elicitation for big data software applications using agile methodology is not easy and straightforward. A detailed requirement is required in the software development life cycle, but agile focuses on less documentation, neglecting quality requirements, difficulties with communication in distributed teams, and quick processing, which can lead to skipping necessary user requirements [5]. Customer satisfaction is the only evidence that requirements are complete and comprehensive, and agile methodology provides many ways to keep the customer involved. However, more customer involvement without any knowledge regarding nonfunctional requirements and experience in the project field lowers the likelihood of project success [15]. Requirement prioritization is a critical issue that can increase cost and time estimation [16].

Different agile methodologies include SCRUM, Kanban, Extreme Programming, Agile Unified Process, and Adaptive Software Development. This research focuses on SCRUM because of its advantages [14].

SCRUM follows agile development process principles as its highest priority is to gain customer satisfaction through all development stages [17]. SCRUM uses short iterations called sprints, which occurs every 2 or 4 weeks. New requirements are developed until the project is completed in each sprint, as shown in Fig. 2. The figure shows the workflow of SCRUM methodology, starting with the product backlog, which includes the prioritized requirement list that has been estimated and added according to the business value of a product owner; then, the team adds a few requirements to the sprint backlog and decides how to implement them. Afterward, the development team starts to implement those requirements through sprints, which occur every 2–4 weeks. They also meet in a daily SCRUM meeting to assess the progress of the project. At the end of a sprint, the developed requirements could be delivered to stakeholders. In the next sprint, the same process, starting again from selecting some requirements from the product backlog to delivering the requirements to stakeholders.

In SCRUM methodology, changes in requirements or technologies are always welcomed at any stage of the development process; however, concurrent changes in requirements negatively affect the entire cycle of software development [14] [17], as well as requirement prioritization, which may increase cost and time estimation [18]. In this case, integrating mind mapping and SCRUM methodology is the most effective, as it helps in obtaining understandable and detailed prioritized requirements, which will significantly and positively affect the software development cycle, as demonstrated in the following section.

### D. Mind Mapping

Mind mapping is a technique representing a system's main ideas hierarchically, which helps a team organize, visualize, and generate new ideas, considering the entire aspects of the system [19].

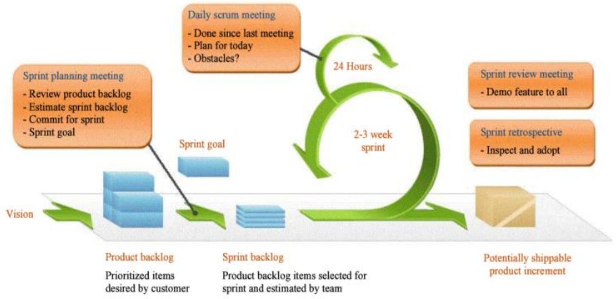

Fig. 2. SCRUM Methodology [35].

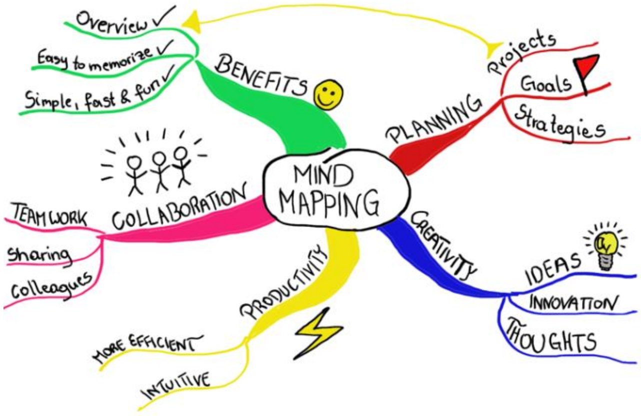

Fig. 3. Mind Mapping Diagram [20].

Mind mapping is performed by placing the most relevant concept in the center of a diagram and connecting it to other concepts, as shown in Fig. 3. This figure is an example of how to represent information in mind maps using multiple layers, and these layers are divided according to the project's requirements.

Mind mapping aids capturing requirements in multiple layers. When collecting data, this results in high quality and accurate requirements by involving stakeholders in the requirement engineering process. This aids in gaining a better and deeper understanding of the overall system's requirements [19]. Mind mapping is a well-known graphical representation concept that can be used on paper or any other tools to most appropriately and accurately represent a system's goal, obtaining complete requirements with their objective value and assisting in obtaining highly prioritized requirements to be implemented [18]. Mind mapping, with its advantages, helps in enhancing a system's requirement quality, which significantly and positively affects the development process and its success. Additionally, integrating mind mapping with SCRUM methodology significantly improves the overall quality of the product backlog.

## III. Related Work

Several studies have been conducted on big data requirement elicitation, as it is a new topic that has attracted researchers' attention in recent years. However, little success has been accomplished in this area, which includes the phases of the requirement engineering process, requirement types, application domains, requirement engineering research challenges, and solutions suggested by requirement engineering research in the context of big data applications [10]. The authors in [21] proposed an artifact model that can capture the main requirement elicitation components and their relationship with the development of big data software applications. In [22], a model was proposed that can engage software engineers and data scientists to discover software requirement processes with their business values for big data software using a use case diagram.

The main objective of this study is to obtain high quality requirements. The authors in [4] presented an approach that specifies quality requirements in the context of big data systems by considering requirement engineering challenges in big data projects. The main idea is to intersect big data characteristics with quality attributes and then identify the system's quality requirements on the basis of that intersection. This proves that big data quality characteristics are mapped to quality requirement specifications. There have been some research papers published on the use of an agile methodology in big data system development to mitigate the challenges of big data system developments. The authors in [23] studied the possibility of applying an agile methodology to big data projects. They gathered information by interviewing experts in big data projects from various organizations. Data are analyzed to determine which agile manifesto concepts can be used to manage big data projects. They have recommended using an agile approach to big data management. In [24], the authors proposed an architecture-centric agile big data analytics development methodology. This architecture enables stakeholders to collaborate effectively to evaluate the importance of the proposition for the system in development and to concentrate on more critical tasks such as value validation. In [25], the authors claimed that in big data analytics using agile methodology, there are three phases. The planning phase is the phase in which the system's stakeholders, goals, and requirements are identified and documented by the product owner in a user story that is prioritized on the basis of independent features. The development phase is the phase in which data are collected according to users' needs, which are then analyzed, and requirements are developed to discover the system's goals and objectives. The closure phase occurs when all requirements are implemented and tested. In [26], the authors studied and analyzed agile methodologies to determine the best practice of business intelligence in big data. They introduced an agile framework that addresses big data's effect on business intelligence using two layers. The first layer comprises five steps (discovery, design, development, deployment, and value delivery) to achieve business goals. The second layer consists of six steps (scope, data acquisition/discovery, analysis, visualization, validation, and deployment) for data analysis and deployment. The two layers are combined to ensure the framework's implementation and management. In [14], a method was proposed for enhancing the quality of the requirement gathering process by combining I*organizational models with standard agile SCRUM methodology; however, because of the complexities of big data projects, it will necessitate additional research.

There are several techniques for requirement elicitation, and the mind mapping technique is the most appropriate for completely and effectively representing the collected requirements and it is effective in large-scale and big data systems [15]. The authors in [27] concluded that the requirement engineering processes (elicitation, analysis, specification, validation, and requirement management) are considered in the development of mind maps and revealed that only functional requirements are considered during the development of mind mapping maps in requirement engineering, with no evidence to consider nonfunctional requirements. In [19], the authors proved the contribution of the mind map development to the software agile methodology results in a high-quality derived product backlog, so mind mapping was recommended for setting up a proper product backlog with agile development methodology such as SCRUM.

According to the above studies, to enhance the quality of big data system's requirements, the best practice for achieving high quality requirements is to integrate requirement gathering election techniques with mind mapping and SCRUM methodology.

## IV. Methodology

In the following section, the proposed framework has been introduced in detail with its integrated sequentially phases. The framework is divided into four main sequential phases, as shown in Fig. 4.

This figure depicts the proposed framework flow, starting with the requirement collection phase, which helps in gathering requirements from stakeholders to gain complete and understandable requirements reflecting the project's scope, aim, and objectives, as described in Section A. The output of this phase is a valuable input to the next phase, which is the mind mapping phase, where all collected requirements will be graphically represented in mind maps, and based on mind mapping and its representation, it will be an added value to obtain the best practice in mapping the collected requirements under the appropriate quality attribute and big data characteristics, as described in detail in section B. In this phase, all requirements will be identified, which will help in identifying functional and nonfunctional requirements for the desired big data project. The output of this phase will be input into the prioritization phase. In this phase, requirements are prioritized on the basis of the business needs and other aspects that will be introduced in section C. Finally, prioritized requirements will be added to the SCRUM product backlog, as shown in Section D.

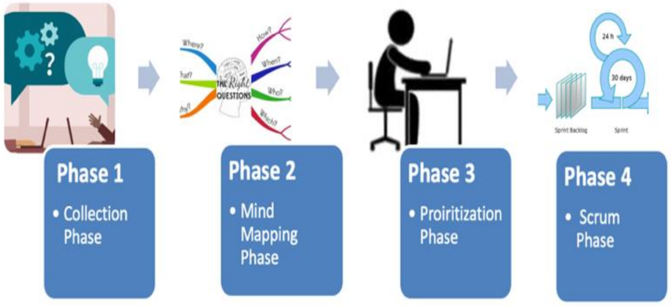

Fig. 4. Proposed Framework Phases.

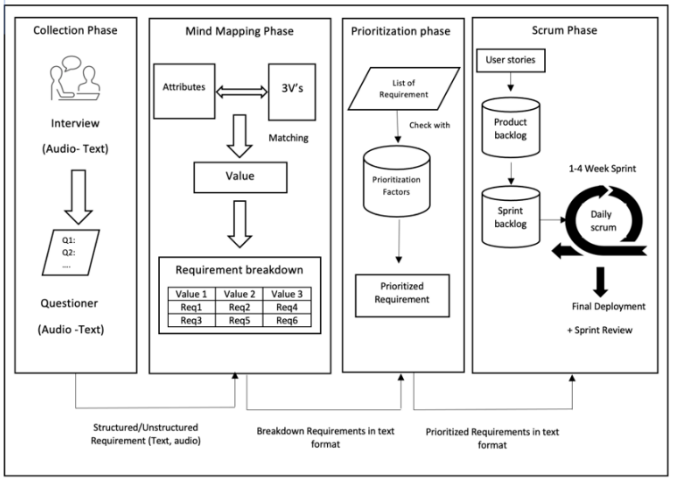

Fig. 5. Framework Methodology.

In the following sections, detailed steps at each phase in the proposed framework are described, as shown in Fig. 5. The steps at each phase are briefly described, and it is demonstrated how beneficial the output of each phase is to the quality of the next phase.

### A. Collection Phase

In the collection phase, all system requirements are collected and specified. Requirement gathering is divided into two steps, each using one of two main requirement elicitation techniques, which are the interview and questioner techniques. The interview technique is extremely effective and useful for gaining full and precise information regarding system requirements, and through the feedback, errors in requirements can be easily found and explained. The interview with stakeholders is based on questions and answers and open discussions, so during the interview, organizational goals, needs, and objectives are identified. Additionally, system users' needs and constraints are identified. Thus, well-defined and organized requirements are identified during the interview, and the interview is recorded and typed in a natural language format to be accessible after the interview. Results from the interview are verified, analyzed, and broken down into clear and understandable points and questions. The requirements are converted into a questionnaire form, then it will be shared with the stakeholders in the second requirement collection step, which is the questioner technique. Presenting all requirements in the form of a questionnaire will gain a better understanding and confirmation of the identified requirements. The outputs of the collection phase are fed into the mind mapping phase, which helps in visualizing complete requirements with high quality.

### B. Mind Mapping Phase

In the mind mapping phase, the most precise requirements are categorized and represented to achieve high quality requirements for big data systems. With the benefits of mind mapping graphical representation and considering big data characteristics, the mind mapping representation is considered the best practice for achieving quality requirements for big data systems. Hence, the objective of mind mapping is to represent the collected requirements under a permutation of big data characteristics, such as volume, velocity, and variety, and quality attributes, such as performance, security, accuracy, availability, and reliability, to achieve the requirement value, as shown in Fig. 6.

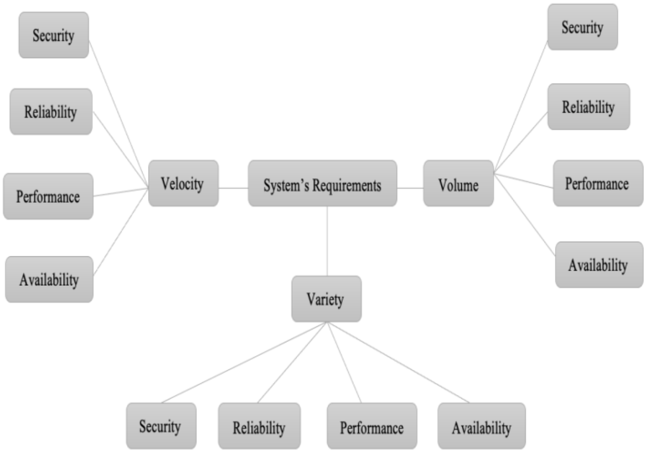

Fig. 6. Mind Mapping Phase Representation.

The figure depicts the division level of mind mapping branches. The first branch level will represent big data characteristics. The second branch level will represent quality attributes. The third branch level will represent all collected requirements from the collection phase. Through two steps, all collected requirements are broken down. In the first step, each big data characteristic will be matched with quality attributes on the basis of the output of requirements of the collection phase. Thus, big data characteristics can be matched with one or more quality attributes according to the big data quality requirement description and goals. In the second step, the collected requirements are matched under each permutation, as discussed in the first step, and shown in the example below.

"System users initiate 10,000 transactions per minute for normal system operations, and these transactions are processed in an average latency of two seconds."

Here, in the above example, the requirement will be added to velocity and performance branches, and the related requirements will be added to these branches as well.

In this phase, all requirements are presented and described without any missing, duplicate, or incomplete information. However, even when new requirements or changes in requirements are added, they will be easily added to the appropriate branch. This will also help in deciding when and how the new requirement will be implemented. Mind mapping has a good visualization effect for the collected information, as it provides users with an in-depth insight to avoid any missing details, and all functional and nonfunctional requirements will be defined.

### C. Prioritization Phase

In the prioritization phase, after the requirements are defined and mapped in the mind mapping diagram, the requirements are prioritized using MoSCoW technique. Based on the MoSCoW technique advantages which are handling a large number of requirements and it is the ease of use and scalable [28]. MoSCoW technique helps in identifying which requirement is mandatory and which is out of scope. By its feature, all the gathered requirements from the mind mapping phase will be assigned under four priority categorize; M is a must-have category in which all the mandatory requirements are assigned so any missing in these requirements will cause a system failure, S is a should-have category in which all the high priority requirements are assigned that can't be postponed, C is a could-have category in which all second priority requirements are assigned, and W is a wont-have category in which assigned requirements will not be implemented in the current development phase and will be implemented in the future.

All the requirements are categorized based on the required evaluation criteria such as business value, which represents the importance of the requirement and how it will affect the organization’s needs; profits, which will affect system efficiency if not considered as a priority or necessitate further changes after implementation, thereby affecting the estimation time and cost; importance, that is, implementing the important requirements first always leads to customer satisfaction; cost, that is, selecting requirements according to its importance based on the budget, as budget is one of the system constraints; and dependency, that is, some requirements depend on others, so it is critical to avoid the disorganization of the dependencies, and constraints, which is if developers should research new technologies this will affect the project cost and time [29]. This categorization will highly affect in the next phase in which it will help product owner to select the most effective features to be implemented.

### *D. SCRUM Phase*

In the SCRUM phase, prioritized requirements are added to the product backlog, which is used for documenting the entire system’s requirements as user stories. The advantage of the prioritization phase is that it helps the product backlog obtain well-identified and prioritized requirements, which helps in making a good decision on which requirements will be added first to the product backlog to efficiently implement the collected requirements using SCRUM methodology.

## V. Implementation

An evaluation study has been conducted on the proposed framework through a set of big data software companies in different application domains to assess the quality of the requirements that followed the proposed framework phases. Conducting a set of surveys for the project’s key roles of system analysts team lead, developer team lead, and quality team leads assessing the output’s credibility and effectiveness by analyzing survey results. In this section, the evaluation study has been conducted in three steps, i.e., planning, action, and output.

### *A. Planning*

The planning step aims to identify how to conduct the evaluation study using sequential steps to obtain an effective result for the proposed framework. The sequential steps begin by identifying the business limitations that will be improved within the proposed framework, followed by preparing the survey questions to which participants will respond according to their technical viewpoint, and finally, identifying the appropriate participant to assist in conducting the evaluation study.

*1) Identify business limitations:* In the first sequential step, the main objective of the framework is to mitigate these limitations and obtain the system’s requirements with high quality. Thus, requirement elicitation limitations have been investigated and observed through the literature review, and the quality factors that have to be addressed are selected and categorized into two parts. The first part includes accuracy and completeness in which the proposed framework aims to achieve requirements with fewer missing details, and conflict requirements, and considering big data characteristics, namely, variety, volume, and velocity, in the requirement specifications because specifying all the related requirements will improve the development cycle and testing cycle as well. The second part includes connectivity and modifiability in which the proposed framework aims to accept any new requirement or any change in the requirement smoothly continuously, and all requirements are connected to generate more understandable and linked requirements, which prioritize the requirements in an effective way that will aid in the development cycle.

The requirement elicitation limitations have been classified under each quality factor part, as shown in the next section.

*2) Prepare survey questions:* Survey questions mainly evaluate the requirements that passed all phases of the proposed framework using four quality factors: requirement accuracy, requirement completeness, requirement connectivity, and requirement modifiability. These factors are related to the common limitation in the requirement elicitation process. The requirement accuracy goal is to obtain requirements that demonstrate the extent to which data accurately characterize the real project and accurately represent all of its elements and aspects. The requirement completeness goal is to obtain complete requirements that contain all essential information, including constraints and conditions, that will help implement the requirements that meet the project’s needs. The requirement connectivity goal is to obtain requirements that are linked together, and each word, definition, characteristic, and element is specified and linked as an entire set. The requirement modifiability goal is to obtain a requirement hierarchy that enables the creation of any new requirement or change in the requirement to be applied completely and consistently while also avoiding any duplication or redundancy in requirements.

The survey questions are divided into two main sections. The first section includes seven questions to evaluate the current company’s framework used by the company. Thus, participants will respond to the survey questions, which indicate how far these limitations are a problem in the followed framework. The second section includes nine questions to evaluate the proposed framework. Hence, participants will respond to the survey questions, which show how far these limitations are still a problem. The survey questions contain multiple-choice responses and one open-ended question. The multiple-choice responses range from 0 to 5, where 0 means strongly disagree and 5 means strongly agree. The open-ended question is general feedback on the proposed framework. A briefly described document has been prepared about the proposed framework, which participants use as a manual guide whenever they need more details while answering the survey questions.

*3) Identify participants:* Identifying participants from big data software projects with their different roles, including system analyst leads, developer leads, and quality leads, as they are the key persons of any project, and their participation in the research based on their technical expertise will have a significant impact on assessing the proposed framework. In this regard, we contacted approximately 20 big data software companies. A request will be sent to all participants to respond to the designed survey's questions according to their technical experience. Responses will be analyzed and converted into a statistical representation showing how far the limitations are a problem before and after applying the proposed framework.

### B. Action

The survey questions and descriptive documents were distributed among the three main roles at the 20 selected companies. Direct communications have been established with the three roles to provide them with additional clarifications regarding the proposed framework and to assist them in implementing all its phases.

### C. Output

Fifteen companies agreed to participate in the study, but some of them did not accept announcing their affiliation because of company constraints. Two other companies declined to participate because of company constraints, and the other three companies did not respond. The responses have been analyzed on the basis of two sections, which are responses on the current company's framework, and responses after applying the proposed framework.

*1) Participant responses based on the current framework:* The participants' responses have been analyzed and represented, as shown below in Table I. The first row lists the seven questions. Q1: How far was the conflict in requirements a problem in your project? Q2: How far was the missing requirement a problem in your project? Q3: How far was the data variety of big data requirements identified using your current project? Q4: How far did the requirement specifications identify data with huge volume and velocity (speed of the received data)? Q5: How complete and consistent was the change in requirements? Q6: How far are the collected requirements linked together and understandable? Q7: How effective was prioritizing the requirements in the implementation phase?

Then, the first column includes the response scores ranging from 0 to 5, where 0 means strongly disagree and 5 means strongly agree. The second column contains the participants in which SATL denotes the system analyst team lead, DTL denotes the developer team lead, and QTL denotes the quality team lead.

The responses are represented as the total number of responses from each participant in each role for each question.

As shown in Table I, the responses identified the extent of the limitations in the software companies, as conflict and missing requirements are still considered an issue in the current working framework. Most of the participants' responses were between 3 and 5, indicating that they strongly agreed on the missing requirements and the requirement conflict. The requirements with their big data characteristics (volume, velocity, and variety) are unspecified, and this is based on the participants' responses, with the average response ranging from 1 to 3. The efficiency of changing any requirement constantly is still a research gap based on the participant's responses, which are ranged from 1 to 3.

Fig. 7 shows the average responses from all companies' participants' roles based on the company's' current framework.

TABLE I. RESPONSES BASED ON THE CURRENT FRAMEWORK

| Score | Role | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 |
|---|---|---|---|---|---|---|---|---|
| 0 | SATL | | | | | | | |
| | DTL | | | | | 1 | | |
| | QTL | | | 1 | | 2 | 1 | 1 |
| 1 | SATL | | | 4 | 8 | 3 | 1 | 1 |
| | DTL | | | 4 | 2 | 5 | 1 | 4 |
| | QTL | | | 6 | 3 | 8 | 2 | 4 |
| 2 | SATL | 5 | 3 | 6 | 2 | 4 | 4 | 3 |
| | DTL | 1 | 2 | 7 | 8 | 5 | 5 | 4 |
| | QTL | | 2 | 6 | 7 | 3 | 8 | 4 |
| 3 | SATL | 1 | 2 | 5 | 3 | 7 | 4 | 4 |
| | DTL | 4 | 2 | 4 | 4 | 3 | 6 | 7 |
| | QTL | 5 | 3 | 2 | 4 | 2 | 3 | 6 |
| 4 | SATL | 7 | 5 | | 1 | | 5 | 3 |
| | DTL | 7 | 4 | | 1 | 1 | 3 | |
| | QTL | 4 | 5 | | 1 | | 1 | |
| 5 | SATL | 2 | 5 | | 1 | 1 | 1 | 4 |
| | DTL | 3 | 7 | | | | | |
| | QTL | 6 | 5 | | | | | |

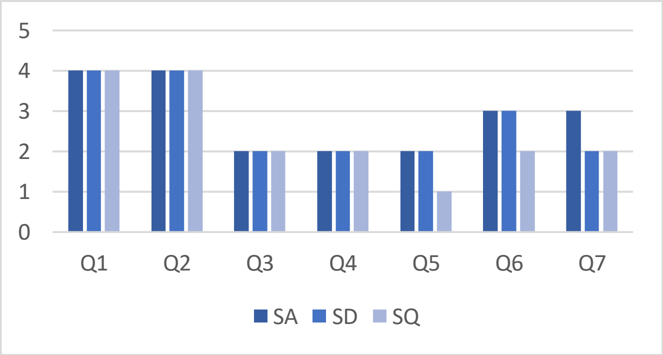

Fig. 7. Average Participant Responses for the Current Framework.

*2) Participant responses after using the proposed framework:* The participants' responses have been analyzed and illustrated, as shown below in Table II. The first row lists the eight questions. Q1: How far was the conflict in requirements still a problem in your project? Q2: How far was the missing requirement still a problem in your project? Q3: How far was the data variety of big data requirements identified within the proposed framework? Q4: How far did the requirements specifications identify the data with huge volume and velocity (speed of the received data) within the proposed framework? Q5: How complete and consistent was the change made in requirement using the proposed framework? Q6: How far were the collected requirements linked together and understandable using the proposed framework? Q7: How far prioritizing the requirements are applied effectively in the implementation phase within the proposed framework? Q8: How far did the elicited requirements specify the required technologies and tools that should be used, and the absolute constraints to the project using the proposed framework? Q9: Open questions on the participant comments and feedback. Then, the first column contains the response scores, which is ranged from 0 to 5, where 0 means strongly disagree and 5 means strongly agree. The second column contains all participants, where SATL denotes the system analyst team lead, DTL denotes the developer team lead, and QTL denotes the quality team lead.

The responses are represented as the total number of responses from each participant in each role for each question.

TABLE II. RESPONSES BASED ON THE PROPOSED FRAMEWORK

| Score | Role | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | SATL | 6 | 3 | 1 | 1 | | | | |
| | DTL | 4 | 9 | | | | | | |
| | QTL | 5 | 3 | | | | | | |
| 1 | SATL | 4 | 7 | 1 | 1 | | | | |
| | DTL | 7 | 5 | 1 | | | | | |
| | QTL | 6 | 7 | | | | | | |
| 2 | SATL | 1 | 2 | 1 | | 2 | | | |
| | DTL | 3 | 1 | | | | | | |
| | QTL | 3 | 4 | | | | | | 1 |
| 3 | SATL | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 2 |
| | DTL | | | | | | | | 1 |
| | QTL | 1 | 1 | | 1 | | | | |
| 4 | SATL | 1 | 1 | 3 | 4 | 9 | 2 | 5 | 6 |
| | DTL | 1 | | 8 | 7 | 6 | 7 | 5 | 8 |
| | QTL | | | 7 | 6 | 8 | 8 | 9 | 7 |
| 5 | SATL | 1 | 1 | 8 | 7 | 3 | 11 | 9 | 7 |
| | DTL | | | 6 | 8 | 9 | 8 | 10 | 6 |
| | QTL | | | 8 | 8 | 7 | 7 | 6 | 7 |

As illustrated in Table II, the responses demonstrated the effectiveness of the sequential phases of the proposed framework on mitigating the software requirement limitation, so conflict and missing details in requirements are identified in the second phase of the proposed framework on the basis of the participants' responses, which ranged from 0 to 3 that strongly disagree with limitation existence and the participants' feedback as well. Using mind mapping, all requirements related to big data characteristics are identified in an understandable and connectable way, allowing for changes in requirements and effective prioritization of requirements.

Fig. 8 represents the average responses from all companies' participants' roles based on the proposed framework.

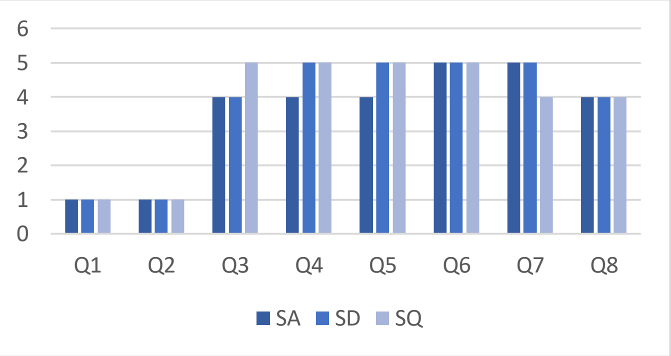

Fig. 8. Average Participant Responses for the Proposed Framework.

The survey questions are formulated to measure the four quality factors, which are divided into two parts:

*3) Part 1: Completeness and accuracy are represented by Q1, Q2, Q3, and Q4:* Survey question responses for the first four questions indicate that the completeness and accuracy of requirements become more realistic and consider the entire main aspects and elements of the project, which primarily applies to the big data project. When comparing the participants' responses before and after applying the proposed framework, requirement completeness and accuracy were improved by more than 50% of the current situation, as shown in Fig. 9.

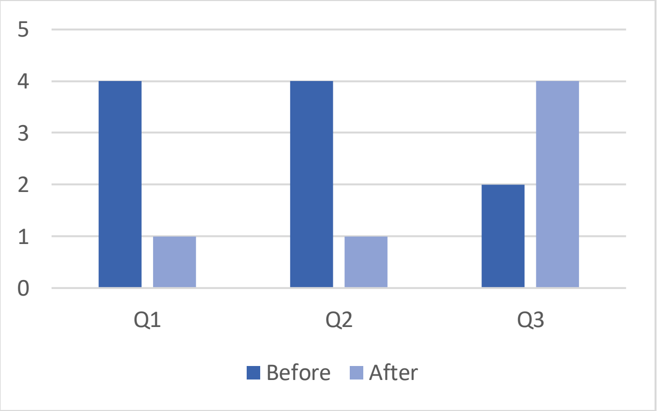

Fig. 9. Comparative Analysis between, before and after applying the Proposed Framework.

*4) Part 2: Connectivity and modifiability are represented by Q4, Q5, Q6, Q7:* Survey question responses for the three questions indicate that the connectivity and modifiability of the requirements help in mapping and linking the requirements together in an understandable and clear scenario; additionally, by identifying big data characteristics, the required technologies and tools that will be used and the absolute constraints applying to the project are specified. When comparing the participants' responses before and after using the proposed framework, the requirement connectivity and modifiability were improved by more than 30% of the current situation, as shown in Fig. 10.

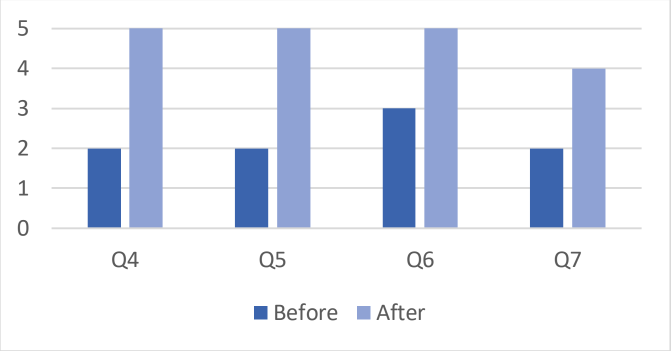

Fig. 10. Comparative Analysis between, before and after using the Proposed Framework.

## VI. Conclusion and Future Work

Requirement engineering is the most crucial stage in the software development life cycle, and more attention should be given to its limitations, such as incomplete, unclear, and conflict in requirements. The complexity of collecting accurate requirements and achieving high quality of the collected requirements becomes more challenging when considering big data characteristics. Representing the huge volume, velocity, and variety of the data in the big data project requirements is the main reason to increase the complexity. An integrated framework with four different phases is proposed; each phase works independently to get the best results for the next phase to improve the requirement engineering process in big data software development. Collection phase is using traditional requirement elicitation techniques to clearly identify all the system requirements from the stakeholder. Mind mapping phase map all the collected requirements under big data characteristics and the quality attributes. The prioritization phase helps to identify requirements under four categories; server, high priority, less priority, and not required to classify which bulk of requirements should be developed first and which should be postponed. SCRUM phase to efficiently implement the big data project requirements. A survey was conducted with industry experts to validate the proposed framework on the basis of their technical background, in which the survey question is assessing the output performance before and after applying the proposed framework. The survey results prove the usefulness of the proposed framework in obtaining high-quality, complete, and detailed requirements for big data projects.

In the future, the proposed framework will be extended and verified to handle different phases in the software engineering process like the design, and the testing phase. Each phase will require different big data characteristics and different big data quality factors need to be achieved.

## References

[1] G Dipti Kumar and P. Alencar, "Software engineering for big data projects: Domains, methodologies and gaps," 2016 IEEE International Conference on Big Data (Big Data), 2016, pp. 2886-2895, doi: 10.1109/BigData.2016.7840938.

[2] The global big data market size to grow at a CAGR of 10.6%", GlobeNewswire News Room, 2021. [Online]. Available: https://www.globenewswire.com/news-release/2020/03/24/2005688/0/en/The-global-big-data-market-size-to-grow-at-a-CAGR-of-10-6.html.

[3] D. Arruda, "Requirements Engineering in the Context of Big Data Applications", ACM SIGSOFT Software Engineering Notes, vol. 43, no. 1, pp. 1-6, 2018. Available: 10.1145/3178315.3178323.

[4] Noorwali, I., Arruda, D. and Madhavji, N., 2016. Understanding quality requirements in the context of big data systems. *Proceedings of the 2nd International Workshop on BIG Data Software Engineering - BIGDSE '16*,.

[5] Aldave, A., Vara, J., Granada, D. and Marcos, E., 2019. Leveraging creativity in requirements elicitation within agile software development: A systematic literature review. Journal of Systems and Software, 157, p.110396.

[6] K.Curcio,T.Navarro,A. Malucelli andS. Reinehr,"Requirements engineer ing: A systematic mapping study in agile software development", Journal of Systems and Software, vol. 139, pp. 32-50, 2018. Available: 10.1016/j.jss.2018.01.036.

[7] Zakari, A., Lawan, A. and Bekaroo, G., 2017. A Hybrid Three-Phased Approach in Requirement Elicitation. Emerging Trends in Electrical, Electronic and Communications Engineering, pp.331-340.

[8] Kasauli, R., Knauss, E., Horkoff, J., Liebel, G. and de Oliveira Neto, F., 2021. Requirements engineering challenges and practices in large-scale agile system development. Journal of Systems and Software, 172, p.110851.

[9] T. Shah and S. V Patel, "A Review of Requirement Engineering Issues and Challenges in Various Software Development Methods", International Journal of Computer Applications, vol. 99, no. 15, pp. 36-45, 2014. Available: 10.5120/17451-8370.

[10] Arruda, D. and Madhavji, N., 2018. State of Requirements Engineering Research in the Context of Big Data Applications. Requirements Engineering: Foundation for Software Quality, pp.307-323.

[11] Al-Jaroodi, J., Hollein, B. and Mohamed, N., 2017. Applying software engineering processes for big data analytics applications development. 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC),

[12] Chung, L., 2016. Big data: A requirements engineering perspective. 2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA),

[13] D. Arruda, "Requirements Engineering in the Context of Big Data Applications", ACM SIGSOFT Software Engineering Notes, vol. 43, no. 1, pp. 1-6, 2018. Available: 10.1145/3178315.3178323.

[14] Kaur, N., Bhuiyan, M., Prasad, P., Haque, F. and Elchouemi, A., 2019. Enhancing Quality in Agile Requirement Elicitation Process. American Journal of Applied Sciences, 16(1), pp.1-10.

[15] F. Shams and H. Bhutta, "Selection of Software Requirements Elicitation Techniques- A Systematic Review", Journal of Multidisciplinary Approaches in Science (JMAS), vol. 1, no. 1, pp. 8-17, 2019.

[16] Bhutta, H., 2019. Requirement Prioritization in Agile Development - A Review. Journal of Multidisciplinary Approaches in Science, (1), pp.18-26.

[17] N. Ramadan and S. Megahed, "Requirements Engineering in SCRUM Framework", International Journal of Computer Applications, vol. 149, no. 8, pp. 24-29, 2016. Available: 10.5120/ijca2016911530.

[18] H. Saeeda, J. Dong, Y. Wang and M. Abid, "A proposed framework for improved software requirements elicitation process in SCRUM: Implementation by a real-life Norway-based IT project", Journal of Software: Evolution and Process, vol. 32, no. 7, 2020. Available: 10.1002/smr.2247.

[19] Mahmud, I. and Veneziano, V., 2011. Mind-mapping: An effective technique to facilitate requirements engineering in agile software development. 14th International Conference on Computer and Information Technology (ICCIT 2011),

[20] Focus. 2021. *15 Creative Mind Map Examples for Students - Focus*. [online] Available at: <https://www.mindmeister.com/blog/students-guide-to-mind-mapping/> [Accessed 12 July 2021].

[21] Arruda, D. and Madhavji, N., 2017. Towards a requirements engineering artefact model in the context of big data software development projects: Research in progress. 2017 IEEE International Conference on Big Data (Big Data),

[22] Altarturi, H., Ng, K., Ninggal, M., Nazri, A. and Ghani, A., 2017. A requirement engineering model for big data software. 2017 IEEE Conference on Big Data and Analytics (ICBDA),

[23] P. Franková, M. Drahošová and P. Balco, "Agile Project Management Approach and its Use in Big Data Management", Procedia Computer Science, vol. 83, pp. 576-583, 2016. Available: 10.1016/j.procs.2016.04.272.

[24] Chen, H., Kazman, R. and Haziyev, S., 2016. Agile Big Data Analytics Development: An Architecture-Centric Approach. 2016 49th Hawaii International Conference on System Sciences (HICSS),

[25] Dharmapal, S. and Sikamani, K., 2016. Big data analytics using agile model. 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT),

[26] Larson, D. and Chang, V., 2016. A review and future direction of agile, business intelligence, analytics and data science. *International Journal of Information Management*, 36(5), pp.700-710.

[27] Quispe Vilchez, E. and Pow-Sang Portillo, J., 2019. Mind Maps in Requirements Engineering: A Systematic Mapping. *Design, User Experience, and Usability. Design Philosophy and Theory*, pp.335-350.

[28] P. Achimugu, A. Selamat and R. Ibrahim, "A Clustering Based Technique for Large Scale Prioritization during Requirements Elicitation", Advances in Intelligent Systems and Computing, pp. 623-632, 2014. Available: 10.1007/978-3-319-07692-8_59.

[29] Saher, N., Baharom, F. and Romli, R., 2020. Guideline for the Selection of Requirement Prioritization Techniques in Agile Software Development: An Empirical Research. International Journal of Recent Technology and Engineering, 8(5), pp.3381-3388.