RESEARCH & INNOVATION DIVISION
IJIST

Improving Software Requirements Elicitation in Agile Environment

Hassan Ali Shah[1], Syeda Hamna Jillani[1], Ghulam Mustafa[1]*, Muhammad Rizwan Rashid Rana [1]

[1]University Institute of Information Technology (UIIT), PMAS Arid Agriculture University, Rawalpindi 46000, Pakistan.

*** Correspondence.** gmustafa@uaar.edu.pk

Citation | Shah. A. H, Jillani. S. H, Mustafa. G, Rana. M. R. R, "Improving Software Requirements Elicitation in Agile Environment", IJIST, Vol. 07 Issue. 01 pp 159-176, January 2025

Received | Dec 19, 2024 **Revised |** Jan 16, 2025 **Accepted |** Jan 18, 2025 **Published |** Jan 20, 2025.

Requirement elicitation plays an important role during the software development life cycle. The selection of an improper requirement elicitation method affect the quality of developed software. Agile methodologies are popular in the industry and follow an incremental approach to developing software. Agile methodologies value customer needs, interaction among teams, interaction with customers and change management. Researchers proposed methods for requirement elicitation in agile software development. This research aims to investigate the issues faced during requirement elicitation in agile software development. We identified the method that motivates the requirements elicitation in agile software development to meet our objectives. Based on various factors including introductory overview, publications, trends and values, we identified strengths and limitations. Based on the identified limitations, we proposed new requirement elicitation method useful in agile software development. To evaluate the results, two teams of equal expertise were given the same project to develop. One of the teams developed using the proposed framework and the other one did without using the proposed framework. Both of them were given the survey they filled out and gave their input on the requirement elicitation parameters and the results were compared and validated using a t-test and reliability analysis. The results of this research were proved promising specially in agile environments.

Keywords. Requirement; Software Development; Agile; Elicitation; Software Industry

Introduction.

Requirement elicitation is a critical step in the software project management process. It plays a key role in ensuring that the final product is developed as per the user's expectations. Correct requirement elicitation also ensures that the project is completed within the defined scope, time and budget. However, requirement elicitation is a complex process and if not done correctly, it can lead to project failure. This is why there is a constant need to improve the methodologies used for requirement elicitation in any type of Software Development Life Cycle (SDLC) model [1]. Therefore, this research aims to identify the existing challenges in the field of requirement elicitation and develop a new methodology to address those challenges. The objective is to develop a tool that can help both technical and non-technical stakeholders gather and prioritize the requirements effectively, ultimately leading to the successful completion of software projects [2].

Software Requirement Engineering (SRE) is the process of refining and defining software requirements to ensure that they are complete, consistent and unambiguous. This involves gathering, analyzing, documenting, validating and managing the requirements throughout the software development process. The SRE process involves communication with the stakeholders to understand their needs and then translating them into technical terms that the developers can use [3]. The requirements are then broken down into smaller units and prioritized based on their criticality and dependencies.

SRE also involves managing changes to the requirements. As the project progresses, requirements may change due to various reasons such as changing business needs, new technical requirements or stakeholder feedback [4][5]. SRE helps in managing these changes by assessing the impact of the changes, modifying the existing requirements or adding new requirements and making sure that all the stakeholders are informed about the changes. The main goal of SRE is to ensure that the software product meets the needs and expectations of the stakeholders while conforming to the technical constraints and specifications. By using SRE, the developers can have a clear understanding of the requirements, minimize the risks associated with the development process and ensure the quality of the software product [6].

In agile methodology, requirements are elicited through various techniques such as interviews, brainstorming, surveys and observation. In addition, agile methodology emphasizes the collaboration of the development team and stakeholders to ensure the accuracy and completeness of the requirements. This means that the stakeholders are involved in the development process throughout the project and their feedback is taken into consideration at each iteration. The requirements are also documented in user stories, which are concise and easy-to-understand descriptions of a feature or functionality from the end user's perspective. User stories are often written on index cards or post it notes and are prioritized in a product backlog [7]. The product backlog is a dynamic document that is continuously updated as the requirements evolve throughout the project. The use of user stories and product backlog allows for flexibility and adaptability to changing requirements, which is a key feature of agile methodology. The requirement elicitation stages are shown in Figure 1.

Requirement elicitation in agile methodology comprises of two levels, one is abstract level requirement elicitation that helps in dividing the project into multiple features that are needed to be developed. In the second stage, the requirements of each feature are elicited in detail and then it is developed [8]. But the requirements must be clear and concise because if the requirements are not clear or correct then the product developed cannot be a quality product. There are multiple ways to elicit the requirements in agile methodology. They are questionnaires, interviews and brainstorming etc. During the requirement elicitation phase, there are multiple challenges are faced by the development team. These challenges make requirement elicitation or requirement refinements.

Requirements often conflict during refining due to compete for resources or have different priorities. Conflicting needs may lead to delays, cost overruns and quality difficulties. Ambiguous criteria are unclear where the requirements lead to misunderstandings between stakeholders and the development team, resulting in a product that doesn't satisfy stakeholders' demands. "The system should be easy to use" is confusing since "easy" means various things to different.

Project requirements should include all stakeholders with a stake in the result. It's hard to plan meetings or requirement elicitation sessions that work for everyone. Especially when stakeholders have busy schedules and conflicting priorities. When stakeholders cannot attend a meeting or requirement elicitation session, their opinions and feedback may not be completely recorded. This is particularly troublesome if these stakeholders have crucial knowledge or unique opinions that might affect the project.

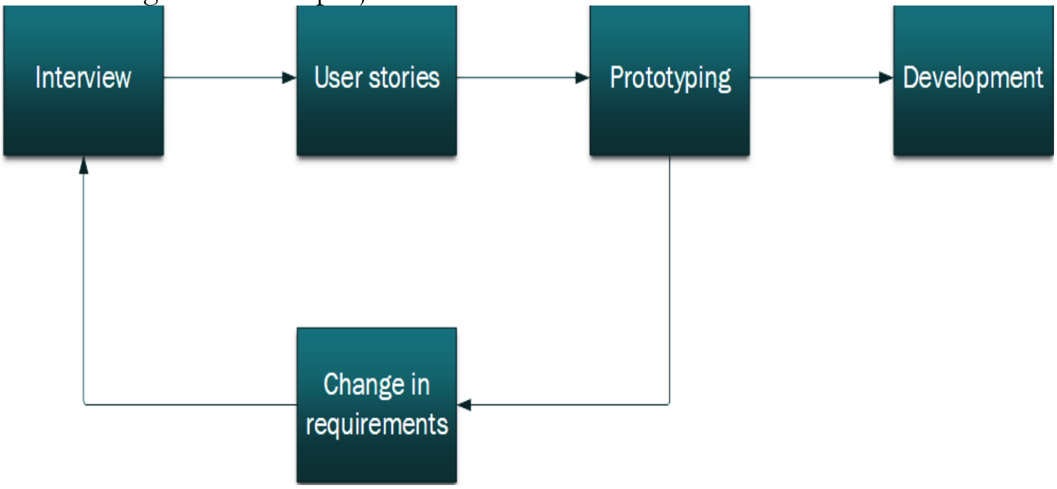

Figure 1. Requirement elicitation stages in agile development

Research Contributions.

SDLC is a critical stage in the development process that involves identifying, analyzing and defining the functional and non-functional requirements for a software application. In the Agile model of software development, this phase is especially important as it helps to establish a shared understanding of the project goals and requirements among the development team and stakeholders.

However, several challenges can arise during the requirements elicitation phase in Agile software development. These challenges can include difficulties in prioritizing requirements, managing changing requirements and ensuring that all stakeholders are engaged and have the opportunity to provide input. The goal of this research is to identify the latest challenges in the requirements elicitation phase of SDLC in the Agile model of software development and to provide a tool that can help software development teams overcome these challenges. By conducting a thorough analysis of the challenges faced by development teams in this phase, the research aims to identify best practices and effective strategies for addressing these challenges.

The tool developed through this research was designed to support development teams in the requirements elicitation phase by providing a structured framework for identifying, prioritizing and managing requirements. The tool was based on the latest research in Agile software development and was designed to be flexible and adaptable to the specific needs and requirements of each development team. Overall, the research aims to help software development teams overcome the challenges in the requirements elicitation phase of Agile software development and to ensure that the final product meets the needs and expectations of all stakeholders involved in the development process.

Research Objectives and Novelty Statement.

The research objectives are as under.

- Identify Challenges in Agile Requirements Elicitation. Conduct a comprehensive analysis to identify the latest challenges faced during the requirements elicitation phase in the SDLC within Agile software development.
- Develop a Structured Tool. Design and develop a tool to support software development teams by providing a structured framework for identifying, prioritizing and managing both functional and non-functional requirements in the Agile model.
- Enhance Collaboration and Stakeholder Engagement. Establish strategies to improve stakeholder engagement and collaboration during the requirements elicitation phase, ensuring effective communication and shared understanding of project goals.

Literature Review.

In this section, the focus is on discussing the background of the research and the existing literature related to the topic. The work that has been done previously was highlighted, along with the research gap that exists in the current literature. Altaf et al. reviewed agile methodology's effects on Global Software Development (GSD) [9]. In GSD, where the development team and stakeholders are from different countries, cultures and time zones, the agile methodology is gaining popularity and Programming XP is the most widely adopted agile development methodology. However, the language gap in requirement elicitation is a challenge for agile techniques. Stakeholders and developers don't have to speak English. Another technique proposed that enables the requirement analyst to apply imagination to obtain requirements [8]. The requirement analyst is not limited by any project management technique. He may utilize any way to collect stakeholder needs and ensure they are final and unambiguous. This method helps the development team effectively collect requirements. This method also prevents the stakeholder and requirement analyst from doing anything complicated or uncomfortable.

Applying the standard approaches, the functional requirements are somehow elicited clearly but the elicitation of non-functional requirements remains a problem. There is a study that focuses on the elicitation of non-functional requirements in agile methodologies [10]. This research proposes a model for eliciting the nonfunctional requirements by employing a Natural Language Processing (NLP) approach that assess the functional requirements and will generate the verifiable nonfunctional requirements this approach reduced the project failure ratio to about 2% to 6%. This approach helped the development team in analyzing and implementing the nonfunctional requirements as well because generally the functional requirements are given more attention and the nonfunctional requirements are left in the dark and are not discussed because the stakeholders do not have enough technical knowledge to discuss them [11]. These authors discussed the ways how to select the suitable requirement elicitation approach for the project [12]. Before starting the requirement engineering process, it is very crucial to identify which requirement elicitation approach was suitable for the project. That was easy to conduct and will not be disturbing the stakeholders. This research concludes that the questionnaire is not suitable for any type of project as no matter how clear the questions are the ambiguity in requirements remains. The interview is a better approach but it is not possible in many scenarios where there is a time gap, language difference or the number of stakeholders is large. Brainstorming is a great approach but it is practically impossible as it requires lots of time of stakeholders and lots of resources of the development team. Storyboarding and prototyping are the only approaches that are suitable for both small-scale and large-scale projects. Combining them removes any chance of ambiguity in the requirements.

There is another systematic literature review that is based on data-driven requirement elicitation approaches [13]. This study is focusing on automatic requirement elicitation approaches. This research concludes that the dynamic sources of data for automatic requirements elicitation are social media and app store reviews. The data collected from these sources is analyzed individually and is converted to a document that has the requirements. This study shows a systematic literature review of requirements elicitation of the apps running on

Blockchain [14]. This study shows that requirement elicitation in decentralized applications is a comparatively difficult task than the centralized apps. This research gathered data from user reviews of decentralized apps and highlighted the importance of software engineering aspects of Blockchain applications. This study concludes that the decentralized apps have an average rating of 3.6 stars and the reviews are majorly about faulty functionality of the application. This means that the organizations need to spend more time and resources on the requirements elicitation phase of SDLC.

Another research focuses on gathering and integrating data from various sources while ensuring interoperability [15]. It concludes that clustering algorithms positively influence data collection and the elicitation process, affecting professionals' ratings for items within the same cluster. Additionally, the study notes a significant, albeit lesser, impact of part-of-speech (POS) tagging on these ratings. Later another study presents RoboREIT, an interactive robotic tutor designed to improve training in requirements elicitation interviews [16]. By simulating a stakeholder role, RoboREIT allows users to practice interview techniques and receive feedback, addressing the challenge of scalability in traditional training. An exploratory user study was conducted to evaluate its effectiveness and applicability in this training context. Some existing studies are shown in Table 1.

Table 1. Tabular Analysis of Literature Review

Research	Contribution	Limitation
Altaf et al. (2019) [9]	Made communication easy and reduced the language barrier	Requirement ambiguity is still a problem.
Saeeda et al. (2020) [17]	Divided requirement elicitation in three phases	Ambiguous requirements were still there
Aldave et al. (2019) [8]	Ambiguity from the requirements was removed	Required a lot of cost to implement
Kumar et al. (2022) [10]	Non-functional requirements alongside functional requirements were also preferred.	Communication and collaboration within the team was not satisfactory.
Iqbal & Shah 2021) [18]	Requirement prioritization was improved	Dependency of requirements was not considered
Ishaq et al. (2021) [19]	An incremental model was used for the requirement elicitation	Dependent modules and their flow were not discussed
Bouraga et al. (2021) [14]	Found that the application running on a blockchain does not have clear requirements	Did not consider the complexity of blockchain and decentralization
(Lim et al., 2021) [13]	Dynamic sources of data were used to remove ambiguity and conflicts	Requirement analysis was not performed

This research introduces a novel, flexible tool specifically designed for the requirements elicitation phase in Agile software development. Unlike existing tools, it incorporates the latest research and best practices to address key challenges such as requirement prioritization, handling dynamic changes and ensuring stakeholder engagement. The tool's adaptability to diverse team needs and project scopes marks a significant advancement in Agile methodologies, promoting enhanced collaboration, better-defined requirements and improved alignment with stakeholder expectations. By focusing on a structured and systematic approach, this research provides a unique contribution to the Agile software development domain, bridging gaps in requirements elicitation practices.

Material and Methods.

The proposed framework begins by opening projects for auction within a project repository, where development teams can submit bids. Once bids are received, the project owner evaluates them and selects the most reliable and suitable proposal. Upon confirmation, a requirements analyst contacts the project owner to initiate the requirements-gathering process.

In this framework, development teams are considered technical stakeholders, while project owners, end-users, and financiers are classified as non-technical stakeholders. The development teams conduct interviews with the non-technical stakeholders, facilitate brainstorming sessions, and employ user stories to gather clear and unambiguous requirements. The requirements document undergoes multiple revisions and is validated by the non-technical stakeholders to minimize ambiguity.

Once the requirements are finalized, tasks are assigned, and a scrum master is appointed to the team. The schedule for sprints and other tasks is defined, and the product development process commences. The team works through the tasks, with the scrum master ensuring effective and efficient progress. Upon completion, the product is delivered to the stakeholders for review.

Overall, this framework provides a structured approach to product development, emphasizing collaboration between technical and non-technical stakeholders. By leveraging user stories and conducting iterative reviews of the requirements document, the framework minimizes ambiguity and ensures the final product aligns with stakeholder needs. The use of sprints and a scrum master further enhances efficiency, ensuring the project is completed on time and within budget. The proposed framework is illustrated in Figure 2.

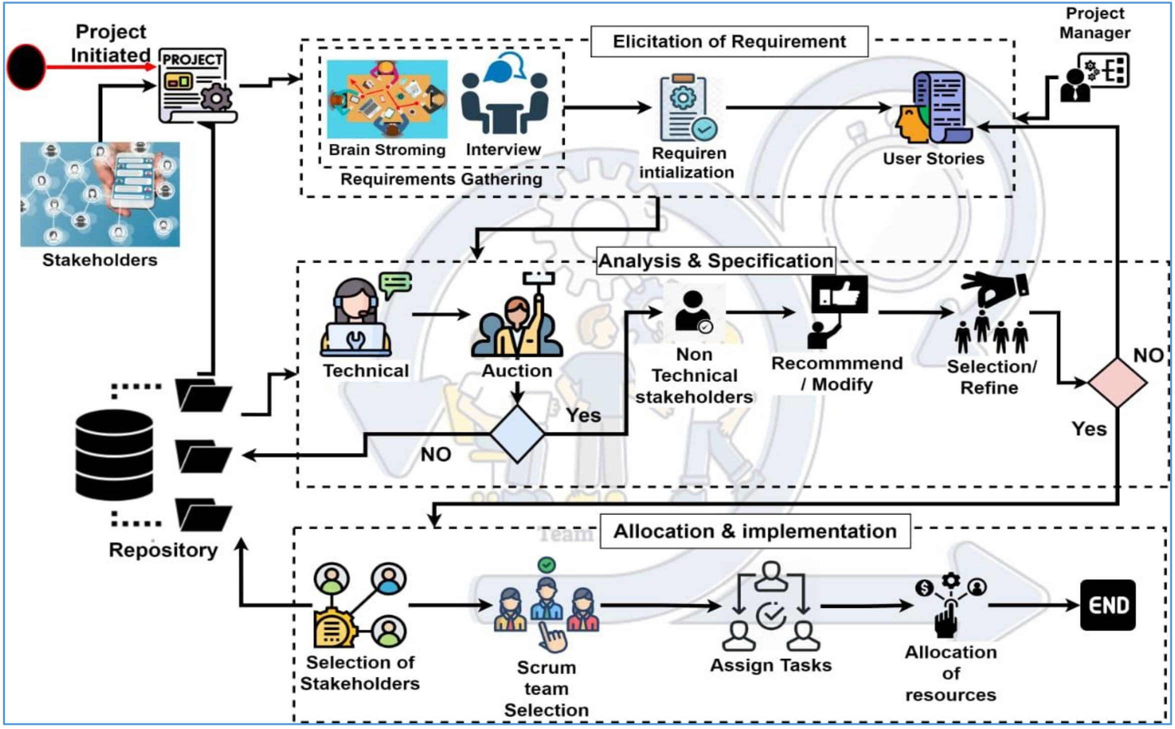

Figure 2. Proposed framework

To facilitate the implementation of the proposed framework in a Global Software Development (GSD) scenario, an Android-based tool has been developed. This tool is designed to enhance communication between companies, their customers, and clients, while also improving collaboration within the organization and with non-technical stakeholders. It serves as a comprehensive platform for communication, collaboration, and project management, enabling companies to manage their projects more efficiently and effectively.

The tool allows project owners to open projects for auction, enabling development teams to submit bids. It also supports task assignment, sprint scheduling, and progress tracking, ensuring seamless project execution. By leveraging this tool, companies can adopt the proposed framework in a more streamlined and efficient manner. It fosters effective communication and collaboration between technical and non-technical stakeholders, ensuring that the final product aligns with stakeholder requirements. Moreover, the tool is particularly well-suited for the Global Software Development scenario, as it enables teams to collaborate effectively across different geographic locations and time zones. By providing a centralized platform for project management and stakeholder interaction, the tool helps bridge the gap between distributed teams, ensuring smooth coordination and timely delivery of projects.

Requirement Elicitation.

The requirement elicitation process is a critical phase in any project. In this framework, three methodologies are integrated and employed to gather requirements from non-technical stakeholders effectively [20]. These methodologies include interviews, brainstorming and user stories and the positive aspects of each approach are integrated to ensure that the final set of requirements is accurate, complete and unambiguous. The project manager plays a key role in this process, initially interviewing the non-technical stakeholders to gain an understanding of their needs and requirements. These requirements are then documented in a requirements document, which is used as a basis for the development team to start the brainstorming sessions. During the brainstorming sessions, technical stakeholders utilize their expertise to identify any ambiguous or unclear requirements in the document. Once such issues are flagged, the non-technical stakeholders are consulted to clarify and refine the requirements, making necessary adjustments to the document. To further eliminate ambiguity and enhance clarity, user stories and storyboards are developed based on the finalized requirements document [21]. These user stories and storyboards help ensure that the requirements are clearly defined and unambiguous, providing a clear and concise understanding of stakeholders' expectations. The entire requirement elicitation process, including the three methodologies used and their integration, is illustrated in Figure 3. By combining these methodologies, the project team can ensure that the final set of requirements accurately reflects the needs and expectations of non-technical stakeholders, resulting in a high-quality product. The requirement elicitation process involve stakeholders not only from the project's immediate domain but also from adjacent industries, enabling cross-domain insights and innovative solutions.

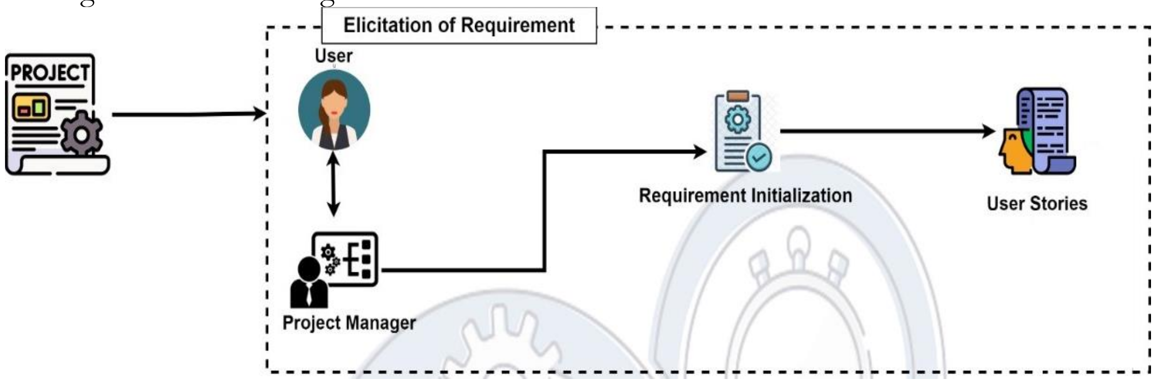

Figure 3. Requirement Elicitation

Requirement Analysis and Specification

As demonstrated in Figure 4, after the requirements are elicited and it has been ensured that there is no further ambiguity left all the requirements are clear and the requirements analysts start to analyze the requirements from a technical perspective. Understanding how the database was structured how many classes was used how the classes interact with each other to make sure that everything is working.

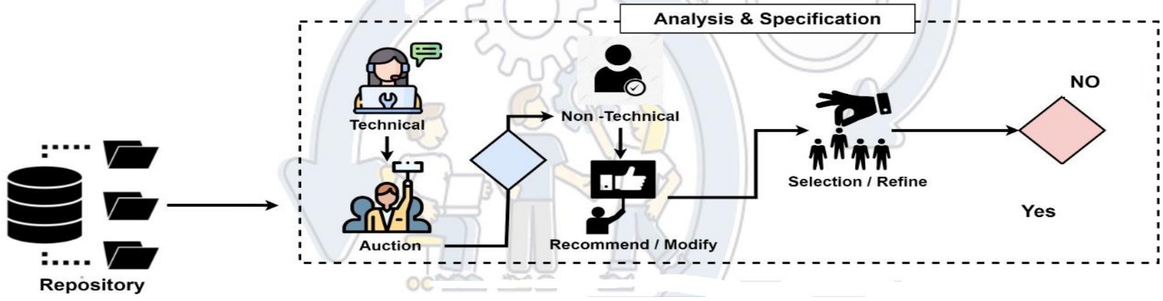

Figure 4. Analysis and Specification

Once the requirements are analyzed, an ambiguity check is applied again to check if some requirement is ambiguous If an ambiguity is found then non-technical stakeholders was consulted and the ambiguity was removed. If in case of ambiguity, the requirements are then again sent to the project manager and then consulted with the non-technical stakeholders, Then the user stories are used again for clarity in the requirements. The user stories are the best way to remove ambiguity as the behavior of the user is clarified using this approach. After that, the dependent requirements are checked. Some dependent requirements are found that might affect the sequence of events while development might also affect the requirement prioritization, the non-tech stakeholders are informed [21]. In case some requirement is found to be contradicting with some other requirements. The non-technical stakeholders are then consulted about the conflicting requirements and the conflict is resolved. Once all the confusions are resolved and the requirements are refined and prioritized, the refined requirements are documented and given to the project manager.

Allocation and Implementation.

Once the requirements are documented the project manager reviews the requirements and selects the scrum master for it. The scrum master then selects the relevant people with relevant expertise. The Sprint schedule is designed and the implementation of the projects starts, as shown in Figure 5.

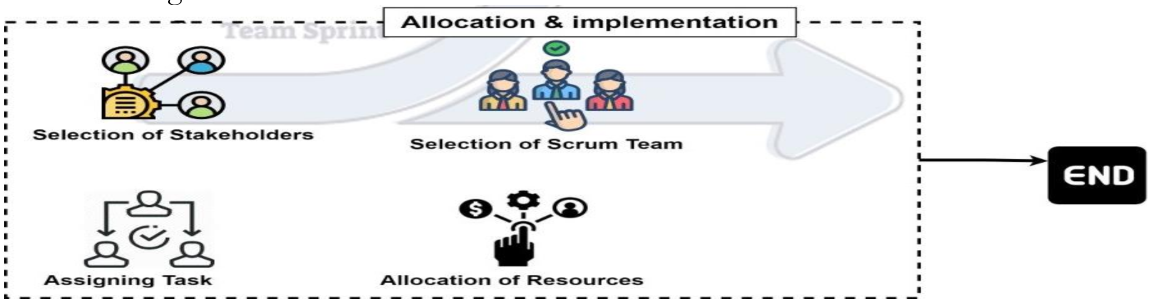

Figure 5. Allocation and implementation

If the developer gets confused about some requirement while development, the scrum master is consulted, The Scrum master give the answer but if the answer is not known to the scrum master, then the project manager is consulted and eventually, if none of the technical stakeholders clears the confusion, the non-technical stakeholders are consulted. After the development, the product is delivered to non-technical stakeholders.

Android Based Tool

The proposed methodology is is new requirement elicitation tool that help the development team to elicit the requirement unambiguously. Figure 6 shows the steps being followed by the tool.

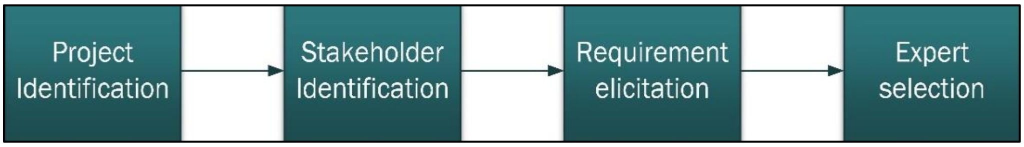

Figure 6. Steps followed in Android-based tool

Identification of Project.

This step is taken by the developers where they analyze their expertise and then select the project according to their expertise and experience. Here every project has some criteria based on which it is selected. For example, the criteria include the domain and nature of the project whether it is a mobile app web app or requires both to be developed. What are the system constraints and how the system was able to meet those constraints? These checks and constraints are demonstrated in Figure 7.

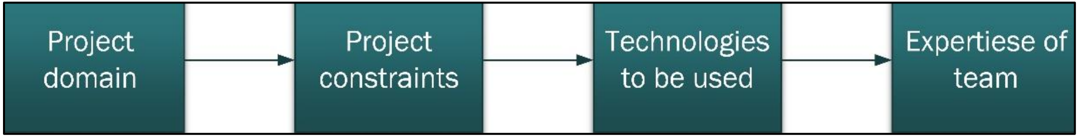

Figure 7. Steps followed in project selection

Identification of Stakeholders.

Who was the end user of the product, and who was the owners of the product? Who provide the requirements for the product? The answers to these questions help to identify the stakeholders of the project. Stakeholders are categorized into two groups. technical and non-technical stakeholders.

Technical Stakeholders.

These include the development team (developers, testers, requirement engineers, and project managers) or any stakeholders with technical knowledge or a technical background. In short, anyone who understands the technical aspects of the project is considered a technical stakeholder. They can also participate in the requirement elicitation phase to clarify any ambiguities. Their involvement helps them better understand the project and ensures that requirements are developed with minimal ambiguity.

Non-Technical Stakeholders.

These are individuals who are not familiar with technical aspects but are connected to the project in some capacity. They can include the product owners and end users. The owners may only have an idea they believe succeed in the market and seek its development as quickly as possible. Despite their lack of technical knowledge, the information they provide is crucial to the project's success. They supply the requirements that guide the development of the final product. Their role is just as vital as that of technical stakeholders since the technical team develop the product based on their vision. This stakeholder division is also illustrated in Figure 8.

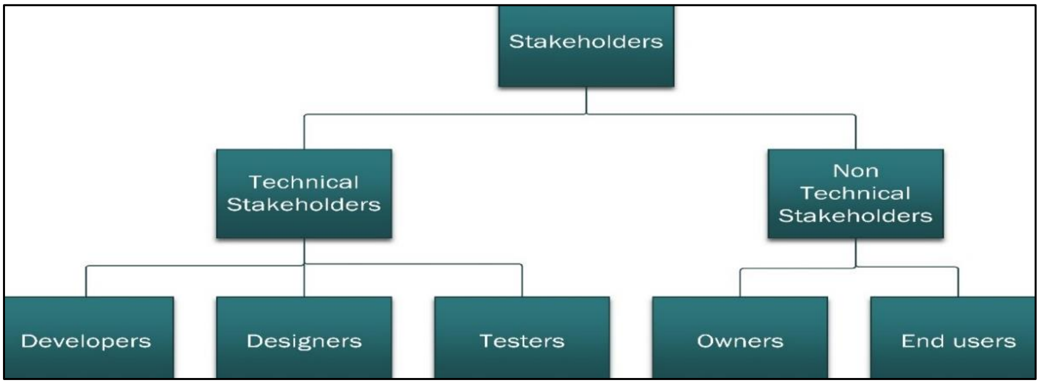

Figure 8. Stakeholders categorization

Requirement Elicitation.

For the requirement elicitation process, the storyboarding and the prototyping was used. As discussed by several systematic literature reviews discussed in section 2 these methodologies have been proven optimal to gather the requirements very efficiently, effectively and unambiguously. The requirements were gathered by as many stakeholders as possible. Then the requirements were refined, the duplicates were removed and the requirements was made ready to give to the experts. This whole process is demonstrated in Figure 9.

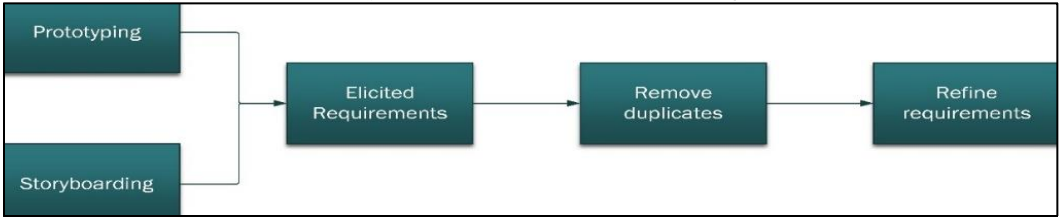

Figure 9. Requirement elicitation process

Expert Selection.

The process of selecting the right experts for a project is a critical step in ensuring its success. The experts chosen by the project manager was responsible for the development of the product and therefore their selection should be done with great care. A wrong selection of experts can lead to delays, cost overruns and even project failure. Therefore, the project manager must choose the experts wisely, ensuring that they have the necessary technical expertise and experience to complete the project successfully [22]. The experts who participate in the development of the products include developers, designers and testers. The project manager selects these experts based on several variables, including their experience, the agile model they have worked in and their domain of expertise. Once the experts have been selected, the project manager provide them with the required documents, including a list of tasks and the duration of sprints. This helps the development team understand what is expected of them and ensure that the project is completed on time and within budget. Figure 10 provides a visual representation of this process. It is important to note that the selection of experts is just one step in the product development process [23]. To ensure a successful outcome, the project manager must also ensure that the team members work together effectively, communicate regularly and have access to the resources they need to complete the project. By carefully selecting the right experts and providing them with the support they need, the project manager can increase the likelihood of project success. Experts was selected from varied industries and professional backgrounds to ensure a comprehensive understanding of the project requirements. This approach help address challenges that may not be apparent within a single domain

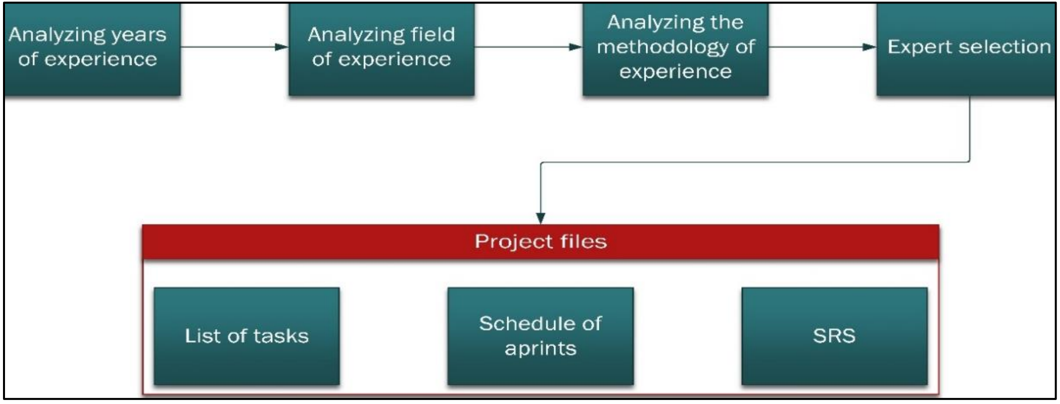
Analyzing years of experience
Analyzing field of experience
Analyzing the methodology of experience
Expert selection
Project files
List of tasks
Schedule of aprints
SRS

Figure 10. Expert selection process

Results And Discussion.

The experiment was designed to evaluate the effectiveness of the proposed framework in terms of the quality of the requirements elicitation process and the efficiency of the development team. The experiment was conducted using a sample of software development teams, which were randomly assigned to either the experimental group or the control group. The experimental group was provided with the proposed framework for requirements elicitation in the Agile model, while the control group used the traditional requirements elicitation approach. The performance of both groups was evaluated using various metrics, including the number of requirements elicited, the time taken to elicit the requirements, the quality of the requirements and the overall satisfaction of the development team.

The results of the experiment indicated that the proposed framework for requirements elicitation in the Agile model was significantly more effective and efficient than the traditional approach. The experimental group elicited a higher number of requirements in less time and with a higher level of quality than the control group. Additionally, the development team using the proposed framework reported higher levels of satisfaction with the requirements elicitation process. In conclusion, the experiment demonstrated that the proposed framework for requirements elicitation in the Agile model is a promising approach to address the gap in the requirements elicitation phase of the software development life cycle. The results of the experiment support the hypothesis that the proposed framework can improve the quality and efficiency of the requirements elicitation process in the Agile model of software development.

Experiments.

To test and evaluate the performance of the proposed framework an extensive experiment is designed. Two teams of 50 members each were chosen with equal expertise in them. Expertise was judged using the experience of the members and the number of different projects they have completed. The same project was assigned to both of them one of them used the proposed framework and the other one used traditional agile methodology. Then after the development process was complete both of them were surveyed and the results were compared. The validation of results is performed using some statistical analysis such as Reliability analysis and T-test analysis.

Parameters.

The parameters chosen for the evaluation of the proposed framework are as follows.

1. Correct project selection
2. Correct technical and non-technical stakeholder classification
3. Requirement clarity
4. Ambiguity
5. Allocation of tasks
6. Communication
7. Collaboration
8. Sprinting

These are the common attributes in the traditional agile way and the proposed framework. A survey was conducted to collect the data.

Demographic Information.

The two teams were selected for one project, Both of them had similar expertise levels and an equal number of professionals in their respective fields. The breakdown of the team members is

1. 1 Project Manager
2. 4 Requirement analysts
3. 20 developers (different areas)
4. 10 SQA engineers
5. 8 designers
6. 4 development team leads (for their respective area)
7. 2 SQA team leads
8. 1 designing team lead

Each member of both teams was carefully selected based on their expertise and experience in their respective fields. The teams consisted of individuals with an equal number of years of experience and completed projects, ensuring that each member brought valuable knowledge and skills to the table. Once the project was finished, each team member was provided with a survey to assess their satisfaction with the project and their overall experience working on the team. The survey results were carefully recorded and analyzed, providing valuable insights into areas where the team excelled and areas where they could improve in future projects.

Hypothesis.

A hypothesis is the expected outcome of this research. Hypothesis also means the prediction of results made by the researcher before conducting the experiments. Researchers then work to prove their hypothesis.

The null hypothesis assumes that there is no relation between the parameters and if there is then it is by chance. For this research, there are no dependencies between the parameters so, a null hypothesis is used. An alternate hypothesis is the opposite of the null hypothesis.

- Null Hypothesis. The proposed framework has improved the process of requirement elicitation and has reduced the challenges.
- Alternate Hypothesis. The proposed framework does not improve the process of requirement elicitation and has reduced the challenges.

To evaluate the effectiveness of the proposed framework, a one-way Analysis of Variance (ANOVA) was performed on the collected data. The ANOVA test was used to assess whether there were statistically significant differences in key variables such as project selection, requirement clarity and stakeholder collaboration. The independent variable in this analysis was the framework used, while the dependent variables were the measured project efficiency indicators. Assumptions of normality and homogeneity of variance were checked before conducting the ANOVA test to ensure the reliability of the results. This statistical approach provided an objective assessment of the framework's impact on requirement elicitation and validation. Additionally, ANOVA was utilized to compare the engagement levels of technical and non-technical stakeholders at various project stages. This analysis helped determine if significant variations existed in stakeholder participation during requirement elicitation, ambiguity resolution and final requirement documentation. Post-hoc tests were conducted to pinpoint the specific project stages where engagement levels differed. The insights gained from this analysis allowed for the identification of key phases where increased stakeholder involvement could further enhance requirement accuracy and overall project success.

Data Collection.

For data collection online tools are used i.e. Google Forms. The single survey is generated and given to both teams ' turn so that it can be differentiated that the first 50 responses are coming from the team using traditional agile and the next 50 responses are coming from the team using the proposed framework.

The experiment was conducted physically and the team using the traditional agile methodologies was not told that the other team was using the proposed framework to retain the anonymity of the teams. Both teams collected the requirements very extensively and then worked in a good manner and delivered the product within a given deadline. After receiving the product, a survey was sent to both teams and the data was received.

Results.

The data collected was statistically analyzed. For data analysis two tests were executed, reliability analysis and T-test analysis.

Reliability Analysis.

This analysis means that the values taken are reliable or not it takes 10 consistent readings to find the average variance between the variables and find the validity of the hypothesis. Its values lie between 1 and -1. This Analysis is performed on the input from both teams.

This analysis is the test that the variables are co-related to each other and can have some internal consistency among them. This most common reliability analysis is Cronbach's alpha. Cronbach's alpha checks the internal consistency of the data and gives the value from 0 to 1. The higher the value is higher the reliability. There are chances that the value might be negative.

Negative value means there is negative covariance in the data and negative covariance means that the data is not reliable. It is globally accepted that the good reliability value for the data to be considered reliable is 0.7 and above. The mathematical representation of Cronbach's Alpha is given by equation 1.

Cronbach's alpha ranges from 0 to 1, with higher values indicating greater internal consistency or reliability. A Cronbach's alpha value of 1 indicates that all the items in the scale or questionnaire are perfectly consistent with one another and measure the same underlying construct. A Cronbach's alpha value of 0 indicates that there is no relationship between the items in the scale or questionnaire.

$$\alpha = \frac{N\bar{c}}{v+(N+1} \qquad (3)$$

Cronbach's alpha has some limitations. For example, it assumes that the items in the scale or questionnaire measure a single underlying construct and it may not be suitable for scales or questionnaires with multiple dimensions. Additionally, it can be affected by the length and complexity of the scale or questionnaire, as well as the sample size and the characteristics of the population being studied. Therefore, it's important to interpret the results of Cronbach's alpha in light of these factors and the specific context of the research. The results obtained after applying the Cronbach's alpha on the data received from the team that used the proposed framework were promising, the Cronbach's alpha value received was 0.761 Which means that the data is highly reliable enough to proceed with the experiment.

The results obtained after applying the Cronbach's alpha on the data received from the team that did not use the proposed framework were promising, the Cronbach's alpha value received was 0.752 Which means that the data is reliable enough to proceed with the experiment. The results obtained after applying the Cronbach's alpha on combining the data from both teams were also promising, the Cronbach's alpha value received was 0.775 Which means that the data is reliable enough to proceed with the experiment. These results are combined and summarized in Table 2. This shows that the internal consistency of the data is good enough to be considered reliable data. This means that the data obtained can be used for experimentation.

Table 2. Reliability Analysis of data

Data	With proposed framework	Without proposed framework	Combined
Cronbach's Alpha's Value	0.761	0.752	0.775

As the data from both teams was reliable enough to proceed with the experiment The next analysis was conducted to compare the data of both teams to observe which team performed better. The reliability analysis is a statistical methodology that is used to observe the internal consistency of the data. The higher internal consistency means higher reliability which is why it is globally considered that if the value of Cronbach's alpha is higher than 0.7 the data is considered to be reliable. But if the value of Cronbach's alpha is lower than 0.7 then the data is not good enough to experiment on. This also means the survey must be conducted again. Another factor that comes in is the negative value. The negative value means that there is a negative average co-variance. This means that there must be something wrong with the data. But in the case of this research the values of Cronbach's alpha are up to the mark and are higher than 0.7 This means that the collected data is good to use for an experiment.

T Test Analysis.

T Test analysis is used to compare the mean of two datasets. In the case of this research, the null hypothesis states that the mean of the data from the team that used the Proposed framework is higher than the data from the team that did not use the proposed framework. To compare the mean of the two datasets this analysis uses equation 1 for computation.

$$t = \frac{m-\mu}{s/\sqrt{n}} \qquad (2)$$

The T-test is used when the sample size is small and the standard deviation is unknown, making it difficult to determine if the means of the two groups are significantly different from each other. The t-test takes into account the sample size, the mean and the standard deviation of each group to calculate a t-value, which is then used to calculate the probability that the two groups are significantly different. The T-test assumes that the data is normally distributed and it is important to check for normality before performing the test. The test also assumes that the samples are independent and there is no relationship between them. If the two samples are dependent, a paired t-test is used instead.

This analysis compares the statistical data of two classes. This analysis creates a standard value for each variable after observing the values of the variables and then tells which class has the lesser standard error. In this case, the responses from both teams were compared and the results for each variable are given in Table 3. In summary, t-test analysis is a statistical method used to compare the means of two groups of data to determine if they are significantly different from each other. It is a commonly used method in hypothesis testing, which helps researchers to make evidence-based conclusions about the differences between the two groups.

Table 3. T-Test Analysis results

Variable Name	Standard error with a proposed framework	Standard error without proposed framework
Project Selection	0.071	0.086
Tech/non-tech classification	0.075	0.112
Requirement Clarity	0.073	0.104
Ambiguity	0.072	0.105
Allocation of tasks	0.071	0.091
Communication	0.077	0.121
Collaboration	0.081	0.286
Sprinting	0.071	0.076

After observing the results, it can be seen that in every variable the team that used the proposed framework performed better than the team that did not use the proposed framework. To further assess the applicability of the proposed framework, it was evaluated across multiple project types, including mobile applications and enterprise solutions. The experiment revealed that the framework is adaptable and scalable to varying project scopes. For mobile applications, the framework enabled faster elicitation of feature-specific requirements tailored to end-user needs, ensuring clarity in rapidly evolving iterations. In enterprise solutions, the framework proved effective in managing the complexity of integrating diverse technical and non-technical stakeholder inputs, improving task allocation and communication in larger teams. The adaptability of the framework across these projects underscores its utility in addressing requirements elicitation challenges in both small-scale agile applications and large-scale enterprise systems, highlighting its versatile application to diverse software development environments. Graphically the results are shown in Figures 11, 12 and 13. This certainly proves the null hypothesis of this research

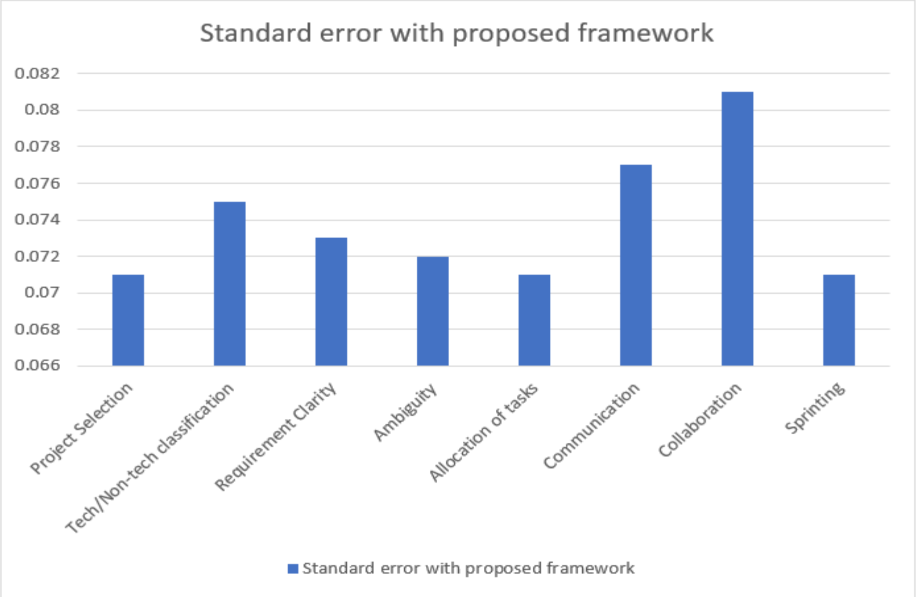

Figure 11. Standard error with a proposed framework

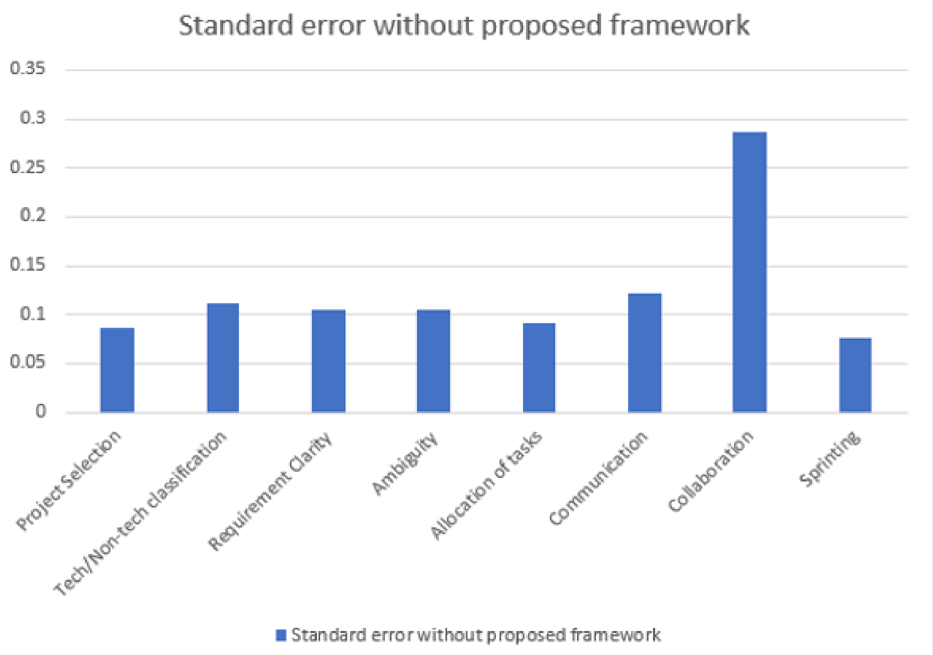

Figure 12. Standard error without proposed framework

The ANOVA results indicate statistically significant differences ($p < 0.05$) across all measured variables, including project selection, requirement clarity and collaboration. The team utilizing the proposed framework demonstrated superior performance in each of these aspects compared to the team following conventional methods. The analysis highlights the framework's effectiveness in streamlining project selection, reducing ambiguities in requirements and enhancing stakeholder collaboration, ultimately leading to improved project outcomes.

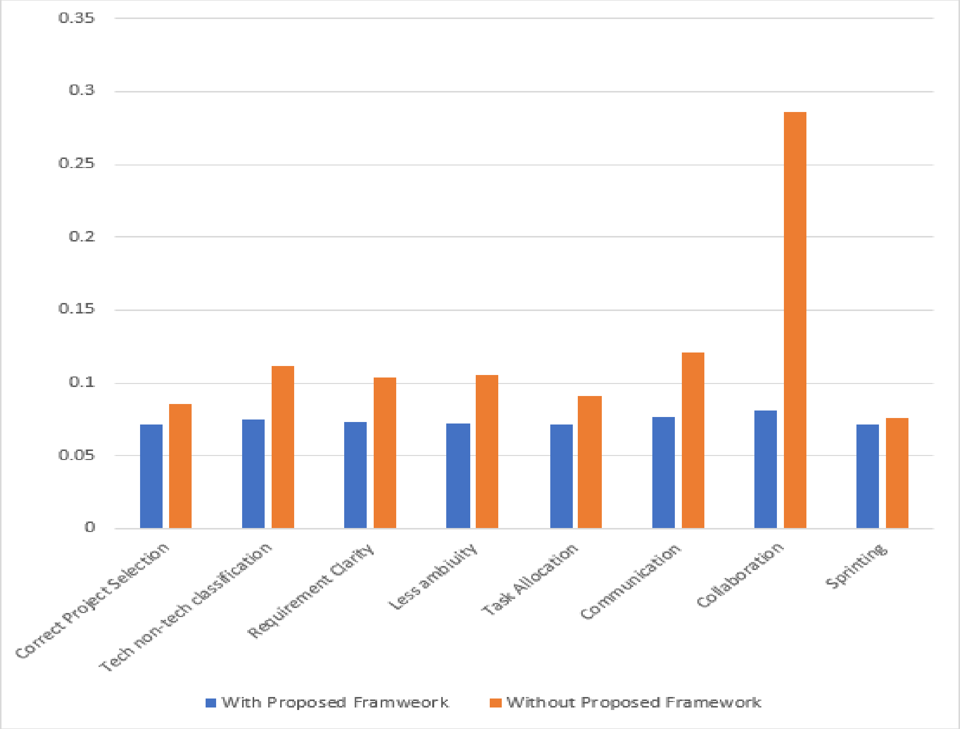

Figure 13. Comparison between standard error

A detailed comparative analysis was conducted to evaluate the impact of the proposed framework on reducing ambiguities and improving stakeholder satisfaction compared to traditional methods. The ambiguity metric was reduced by 30% when using the proposed framework (mean value. 4.2/5) compared to the traditional method [15] (mean value. 3.1/5). Similarly, stakeholder satisfaction improved by 25%, as indicated by survey responses where participants rated their overall experience and clarity higher under the proposed framework.

Conclusion.

Requirement elicitation plays an important role during the software development life cycle. The selection of improper requirement elicitation methods affect the quality of developed software. Agile methodologies are popular in the industry and follow an incremental approach to developing software. Agile methodologies give value to customer's needs, interaction among teams, interaction with customers and change management. Researchers proposed methods for requirement elicitation in agile software development. The objective of this research is to investigate the issues faced during requirement elicitation in agile software development. To meet our objective, we identified the method that motivates the requirements elicitation in agile software development. After identifying the literature systematic literature review was performed. An introductory overview, publications trends and values, strengths and limitations was highlighted. Based on the identified limitations, we proposed a new requirement elicitation method useful in agile software development. A case study-based experiment was performed to evaluate the effectiveness of our proposed approach. For future work, we recommend leveraging machine learning and natural language processing (NLP) techniques to automatically extract, classify and prioritize both functional and non-functional requirements from stakeholder communications, documents and feedback. This automation could enhance the accuracy and efficiency of requirement elicitation in agile software development.

Acknowledgment. No Funding received.

Author's Contribution. Formal analysis, A.H.S, H.S.J, G.M.; methodology M.R.R.R, Manuscript review, M.R.R.R, G.M., supervision, A.H.S; writing—original draft. All authors have carefully reviewed and granted their approval for the final manuscript version to be published.

Conflict of interest. The authors declare no conflicts of interest.

Project details. Nil

References.

[1] M. S. F. Baharom, "Current Practices and Required Knowledge of Non-Functional Requirements Elicitation in Agile Context. Key Results of an Empirical Study," ECTI Trans. Comput. Inf. Technol., vol. 18, no. 3, pp. 356–370, 2024, [Online]. Available. https.//ph01.tci-thaijo.org/index.php/ecticit/issue/view/17508

[2] R. Á. P. F. H. Martins, A. Junior, E. Canedo, R. A. Dias Kosloski, "Design Thinking. Challenges for Software Requirements Elicitation," Information, vol. 10, no. 12, p. 371, 2019, doi. https.//doi.org/10.3390/info10120371.

[3] M. Neumann, Y. Bogdanov, M. Lier and L. Baumann, "The Sars-Cov-2 Pandemic and Agile Methodologies in Software Development. A Multiple Case Study in Germany," Lect. Notes Bus. Inf. Process., vol. 408, pp. 40–58, 2021, doi. 10.1007/978-3-030-67084-9_3.

[4] M. S. S. S. Santos, T. Pimentel, F. G. Rocha, "Using Behavior-Driven Development (BDD) for Non-Functional Requirements," Software, vol. 3, no. 3, pp. 271–283, 2024, doi. https.//doi.org/10.3390/software3030014.

[5] M. M. V. Siddeshwar, S. Alwidian, "A Systematic Review of AI-Enabled Frameworks in Requirements Elicitation," IEEE Access, vol. 12, pp. 154310–154336, 2024, doi. 10.1109/ACCESS.2024.3475293.

[6] M. I. N. Rashid, S. U. Khan, H. U. Khan, "Green-Agile Maturity Model. An Evaluation Framework for Global Software Development Vendors," IEEE Access, vol. 9, pp. 71868–71886, 2021, doi. 10.1109/ACCESS.2021.3079194.

[7] S. O. Alok Mishra, Samia Abdalhamid, Deepti Mishra, "Organizational issues in embracing Agile methods. an empirical assessment," Int. J. Syst. Assur. Eng. Manag., vol. 12, pp. 1420–1433, 2021, doi. https.//doi.org/10.1007/s13198-021-01350-1.

[8] T. A. & M. S. Faiz Akram, "Recommendation systems-based software requirements elicitation process—a systematic literature review," J. Eng. Appl. Sci., vol. 71, p. 29, 2024, [Online]. Available. https.//jeas.springeropen.com/articles/10.1186/s44147-024-00363-4

[9] M. S. M. Hidalgo, F. Yanine, R. Paredes, Jonathan Frez, "What Is the Process? A Metamodel of the Requirements Elicitation Process Derived from a Systematic Literature Review," Processes, vol. 13, no. 1, p. 20, 2025, doi. https.//doi.org/10.3390/pr13010020.

[10] A. Garg, R. Kumar Kaliyar and A. Goswami, "PDRSD-A systematic review on plan-driven SDLC models for software development," 8th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2022, pp. 739–744, 2022, doi. 10.1109/ICACCS54159.2022.9785261.

[11] E. D. Canedo, A. Toffano Seidel Calazans, A. J. Cerqueira, P. H. Teixeira Costa and E. T. Seidel Masson, "Agile Teams' Perception in Privacy Requirements Elicitation. LGPD's compliance in Brazil," Proc. IEEE Int. Conf. Requir. Eng., pp. 58–69, 2021, doi. 10.1109/RE51729.2021.00013.

[12] I. H. H. et.al, "An Approach for Selecting the Suitable Requirement Elicitation Technique," Turkish J. Comput. Math. Educ., vol. 12, no. 3, pp. 2083–2087, Apr. 2021, Accessed. Jan. 23, 2025. [Online]. Available. https.//turcomat.org/index.php/turkbilmat/article/view/1078

[13] A. H. & J. Z. Sachiko Lim, "Data-Driven Requirements Elicitation. A Systematic Literature Review," SN Comput. Sci., vol. 2, p. 16, 2021, doi. https.//doi.org/10.1007/s42979-020-00416-4.

[14] S. Bouraga, C. Burnay, I. Jureta and S. Faulkner, "Requirements Elicitation for Applications Running on a Blockchain. Preliminary Results," Lect. Notes Bus. Inf. Process., vol. 424 LNBIP, pp. 38–46, 2021, doi. 10.1007/978-3-030-79108-7_5.

[15] F. A. Bernanda Devi Yurisca,Dayang N.A. Jawawi, Shahliza Abd Halim, "Natural Language Processing For Requirement Elicitation In University Using Kmeans And Meanshift Algorithm," Baghdad Sci. J., vol. 21, no. 2, 2024, doi. https.//doi.org/10.21123/bsj.2024.9675.

[16] B. Görer and F. B. Aydemir, "RoboREIT. An interactive robotic tutor with instructive feedback component for requirements elicitation interview training," J. Softw. Evol. Process, vol. 36, no. 5, p. e2608, May 2024, doi. 10.1002/SMR.2608.

[17] H. Saeeda, J. Dong, Y. Wang and M. A. Abid, "A proposed framework for improved software requirements elicitation process in SCRUM. Implementation by a real-life Norway-based IT project," J. Softw. Evol. Process, vol. 32, no. 7, p. e2247, Jul. 2020, doi. 10.1002/SMR.2247.

[18] A. S. M. A. Iqbal, "Stakeholder's Evaluation Process for GSD Based Requirements Elicitation Frameworks," Pakistan J. Eng. Technol., vol. 4, no. 1, pp. 175–183, 2021.

[19] K. Ishaq, F. Rosdi, N. A. M. Zin and A. Abid, "Requirements Elicitation for Game-based Language Learning Application," 4th Int. Conf. Innov. Comput. ICIC 2021, 2021, doi. 10.1109/ICIC53490.2021.9693004.

[20] U. F. Muhammad Yaseen, "Requirement Elicitation Model (REM) in the Context of Global Software Development," Int. J. Adv. Appl. Sci., vol. 7, no. 3, 2018, doi. http.//doi.org/10.11591/ijaas.v7.i3.pp303-308.

[21] A. Sinha and P. Das, "Agile Methodology Vs. Traditional Waterfall SDLC. A case study on Quality Assurance process in Software Industry," 2021 5th Int. Conf. Electron. Mater. Eng. Nano-Technology, IEMENTech 2021, 2021, doi. 10.1109/IEMENTECH53263.2021.9614779.

[22] N. Bahurmuz, R. Alnajim, R. Al-Mutairi, Z. Al-Shingiti, F. Saleem and B. Fakieh, "Requirements Elicitation Techniques in Mobile Applications. A Systematic Literature Review," https.//services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJITPM.2021070101, vol. 12, no. 3, pp. 1–18, Jan. 1AD, doi. 10.4018/IJITPM.2021070101.

[23] A. Alzayed and A. Khalfan, "Understanding Top Management Involvement in SDLC Phases," J. Softw., pp. 87–120, May 2022, doi. 10.17706/JSW.17.3.87-120.