

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Proposing a Framework for Collecting Software Project Requirements from Non- technical Customers

Nirjhor Anjum, Md Anwarul Kabir, Kazi Jahanul Islam, Ariful Alam

Head of Engineering, ADN Telecom Limited, Dhaka, Bangladesh

Assistant Professor, AIUB, Dhaka, Bangladesh

Lead Business Analyst, Foster Technologies Limited, Dhaka, Bangladesh

Project Coordinator (Associate), Infinity Flame Software Limited, Sylhet, Bangladesh

ABSTRACT: In software development, various Software Development Life Cycles (SDLCs) such as Agile and Waterfall are used. Each SDLC is suitable for different types of projects depending on the needs and complexity. Although these SDLCs provide a standard process with steps and tasks for software development, they do not clearly describe how to gather requirements from customers. In most cases, the requirements are collected from customers by project managers. However, the customers are usually from non-technical backgrounds, and often, project managers are either technical, semi-technical, or non-technical themselves. This creates a major challenge in collecting proper technical requirements. It becomes difficult for both parties to understand and handle the information accurately. In this paper, a new framework is proposed to assist in collecting technical requirements from non-technical customers. The framework will provide a more effective way to simplify and organize the requirements, making them easier to manage and understand for both technical and non-technical people. To develop this framework, primary data will be collected through questionnaires from several software companies. In addition, case studies will be analyzed to support the findings. Based on the research and analysis, a new requirement gathering framework will be proposed, which aims to improve the requirement collection process and benefit the software development industry.

KEYWORDS: Requirement Collection, Software Development Life Cycle (SDLC), Agile, Waterfall, Non-technical Customers, Requirement Engineering, Framework Design, Software Project Management, Software Requirements.

I. INTRODUCTION

Requirement collection is one of the most important tasks in the Software Development Life Cycle (SDLC). If requirements are not collected properly, the entire software development project can face serious problems. In most projects, requirements are gathered by a project manager. The project manager can be technical, semi-technical, or non-technical depending on their background. Similarly, in many projects, the customers who provide the requirements come from non-technical backgrounds. This situation creates a major challenge in the industry.

When technical requirements must be collected from non-technical customers by a project manager who is not fully technical, the risk of misunderstanding becomes very high. If the requirement collection process is faulty at the beginning, it becomes extremely difficult to build the correct software later. As a result, projects may fail completely or suffer from heavy losses in cost and time.

At present, several SDLC models like Agile and Waterfall are widely used for managing software development. These models offer clear guidelines about development stages and activities. However, they do not provide any specific method or framework for collecting technical requirements from non-technical customers. In real-world projects, this gap often creates confusion and communication problems between customers and development teams. Without a

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

standard process for gathering requirements, it becomes difficult to ensure that the requirements are correct, complete, and understandable for both technical and non-technical sides.

Considering this major problem, the main goal of this paper is to find a better solution. The first aim is to study how requirements are currently collected in the software industry. The second aim is to identify the problems and mistakes that happen during requirement collection. The third and final aim is to propose a new and standard framework that can guide project managers in collecting technical requirements more successfully, even when they are not technical experts.

To achieve this goal, five leading software companies will be selected for data collection. These companies will be from different countries to make the findings more diverse and useful. The selected companies will be from the USA, UK, Sweden, India, and Bangladesh. Primary data will be collected through structured surveys and interviews with people involved in requirement gathering, such as project managers and business analysts. After collecting the data, the information will be analyzed and compared to identify patterns, common issues, and good practices.

Based on the results of the research, a new requirement collection framework will be proposed. This framework will offer a simple and easy-to-follow process for gathering technical requirements from non-technical customers. The new approach will help to reduce misunderstandings, minimize project failures, and improve the overall success rate of software development projects.

II. OBJECTIVE OF THE RESEARCH

In the majority of software projects, project managers are either semi-technical or non-technical. Only in rare cases are they fully technical experts. Due to this situation, project managers often find it difficult to collect technical requirements from customers. At the same time, most customers providing the requirements are also from non-technical backgrounds. As a result, collecting technical requirements becomes a complex task when a non-technical or semi-technical project manager has to gather information from a non-technical customer.

This paper focuses on studying how requirements are currently collected in the software industry. It aims to find out the common mistakes, gaps, and weaknesses in the existing requirement collection practices. By analyzing these practices, the paper aims to offer a better and more organized way to collect requirements.

The main objective of the research is to propose a new framework for collecting technical requirements from non-technical customers. The framework will provide clear steps that project managers can follow easily. It will help project managers collect requirements more accurately and clearly, even if they do not have strong technical skills.

By following the proposed framework, it will become easier for software companies to reduce misunderstandings, complete projects successfully, and deliver products that meet customer expectations. The ultimate goal is to create a more reliable, structured, and effective requirement collection process that can be used widely in the software industry.

III. BACKGROUND STUDY

The software industry has developed rapidly over the last few decades across different countries including the United States, United Kingdom, Sweden, India, and Bangladesh. Compared to traditional industries like manufacturing and construction, software development is still considered a young industry. In the United States and United Kingdom, the early growth of the software industry began in the 1960s and 1970s, supported by advancements in computing technology (Cusumano, 2012). Sweden saw significant software development growth during the 1980s, particularly with the expansion of the telecommunications sector (Ågerfalk & Fitzgerald, 2008). India's software industry gained momentum during the 1990s following economic liberalization, while Bangladesh started developing its software sector in the 1980s with more organized growth during the late 1990s (Begum, Khan, Hafiz, Islam, & Shoyaib, 2009).

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Today, the software industries in these five countries contribute significantly to their national economies. In the United States and United Kingdom, software companies provide solutions across a wide range of sectors such as healthcare, finance, education, and retail. In Sweden, the software industry is particularly strong in fintech, automation, and telecommunications (Ågerfalk & Fitzgerald, 2008). India has become one of the global leaders in IT services and outsourcing, while Bangladesh's software industry has been growing steadily, both for offshore and domestic markets.

Software companies across these regions offer various services. Some companies mainly work on the customization and configuration of existing software solutions, while others develop completely new software products or web-based applications. At present, a large portion of the software projects are based on web and cloud technologies. Commonly used technologies include PHP, ASP.NET with C#, Python frameworks, and Java-based technologies like JSP and J2EE (Begum et al., 2009).

Requirement collection is a critical foundation for successful software development projects. Properly collected requirements help with accurate project estimation, detailed analysis, effective planning, and resource allocation. Studies have shown that the success of a software project is closely tied to the quality of its requirements (Bano & Zowghi, 2015). Even though initial requirements are often well documented at the start of a project, changes during the development life cycle are common and create management challenges (Tiwana & Keil, 2004).

The difficulty of managing changing requirements has been widely reported across industries. As technology and business environments become more complex, projects face increased risks when requirement collection is not handled carefully. Poorly collected requirements often lead to scope creep, increased costs, delays, and even project failure (Bano & Zowghi, 2015). Research highlights that one of the primary causes of software project failure is miscommunication during the early stages of requirement collection (Demirel & Das, 2018).

Several strategies have been proposed to deal with requirement management challenges. Techniques like Joint Application Development (JAD), prototyping, and the establishment of formal Change Control Boards have been suggested to control requirement changes (Jones, 1995). Early baselining of requirements is also recommended to reduce confusion and improve project control (Wieggers, 1999). In recent years, many companies in the United States, United Kingdom, Sweden, India, and Bangladesh have adopted Agile or iterative models to handle changing requirements more flexibly (Rajlich, 2006).

While significant progress has been made in managing changing requirements during the development phases, less attention has been given to improving the process of collecting requirements from the beginning, especially when dealing with non-technical customers. Customers often have a clear understanding of their business needs but struggle to express them in technical language. Project managers, especially those who are non-technical or semi-technical, may also find it difficult to translate these needs into technical specifications accurately.

This communication gap between non-technical customers and technical teams is a major challenge that can lead to misunderstandings, incomplete requirements, and confusion during development. If technical requirements are not collected properly at the beginning, the chances of project failure increase significantly (Bano & Zowghi, 2015; Demirel & Das, 2018).

The main focus of this research paper is to propose a structured framework to improve the process of collecting technical requirements from non-technical customers. Although earlier research has emphasized techniques for managing changing requirements, there is a need for a clear and practical method that can guide project managers from the very beginning of the project.

This proposed framework will be based on primary research conducted across leading software companies from the United States, United Kingdom, Sweden, India, and Bangladesh. By studying the industry practices of these five countries, the framework aims to be internationally applicable. The goal is to help bridge the communication gap between customers and project teams, improve requirement quality, and contribute to better project outcomes.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

IV. RESEARCH METHODOLOGY

This research paper is based on a combination of three primary research methods: the Survey Method, the Detailed Interview Method, and the Case Study Method. These methods were selected carefully to analyze real-world software development projects, study their software development life cycles, and find the best practices for requirement collection, especially from non-technical customers.

At the initial stage of the research, a structured set of questions was prepared. These questions were designed to cover different phases of the Software Development Life Cycle (SDLC), including requirement gathering, system design, development, testing, and maintenance. The questionnaire aimed to identify how requirements are collected, documented, and managed across different types of projects.

To ensure the quality of the collected data, five experienced project managers were selected from software companies operating in the United States, United Kingdom, Sweden, India, and Bangladesh. These project managers were chosen because they had direct involvement with requirement collection activities in their respective organizations. Their valuable feedback on the questionnaire helped to uncover critical problems related to requirement collection practices. From their responses, it was clearly identified that requirement collection is one of the major challenges in software development projects.

However, it was found that the survey method alone was not sufficient to fully understand the complexity of the problems related to requirement collection. Therefore, a more detailed approach was taken through the Interview Method. In this stage, in-depth interviews were conducted with the same project managers who participated in the survey. During these interviews, the participants were encouraged to share detailed experiences, examples, and strategies they used to manage requirement collection challenges in their projects. The interviews provided deeper insights into the causes of requirement collection issues and also highlighted the methods and practices that project managers applied to address those issues. Still, it was realized that more practical examples were needed to strengthen the study.

Next, the Case Study Method was applied. Five successful but challenging software projects were selected from leading software companies operating across the five target countries. These projects were chosen from different categories, such as e-commerce, healthcare, education technology, financial technology, and enterprise solutions, and they involved different software development models such as Agile, Waterfall, and Hybrid approaches. The projects also involved clients from different cultural and business environments to ensure that the study would not be limited to a specific type of development or location.

The selected case studies focused particularly on projects that faced significant problems during the requirement collection phase, even though the companies managing them were well-established and experienced. By analyzing these real-world case studies, the research was able to find patterns, sources, and types of requirement collection problems that were common across different industries and countries.

The combination of survey responses, interview findings, and case study analysis helped this research to develop a comprehensive understanding of the challenges involved in collecting technical requirements from non-technical customers. It allowed the research to identify the root causes, scopes, and impacts of the problems. Most importantly, it guided the research in formulating a structured and practical framework to improve the requirement collection process.

Although the software industries in the United States, United Kingdom, Sweden, India, and Bangladesh vary in size and maturity, the challenges related to requirement collection from non-technical customers were found to be remarkably similar. Therefore, even with a small but focused sample size of five project managers and five projects, the data collected were considered sufficient to build a reliable and effective requirement collection framework.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

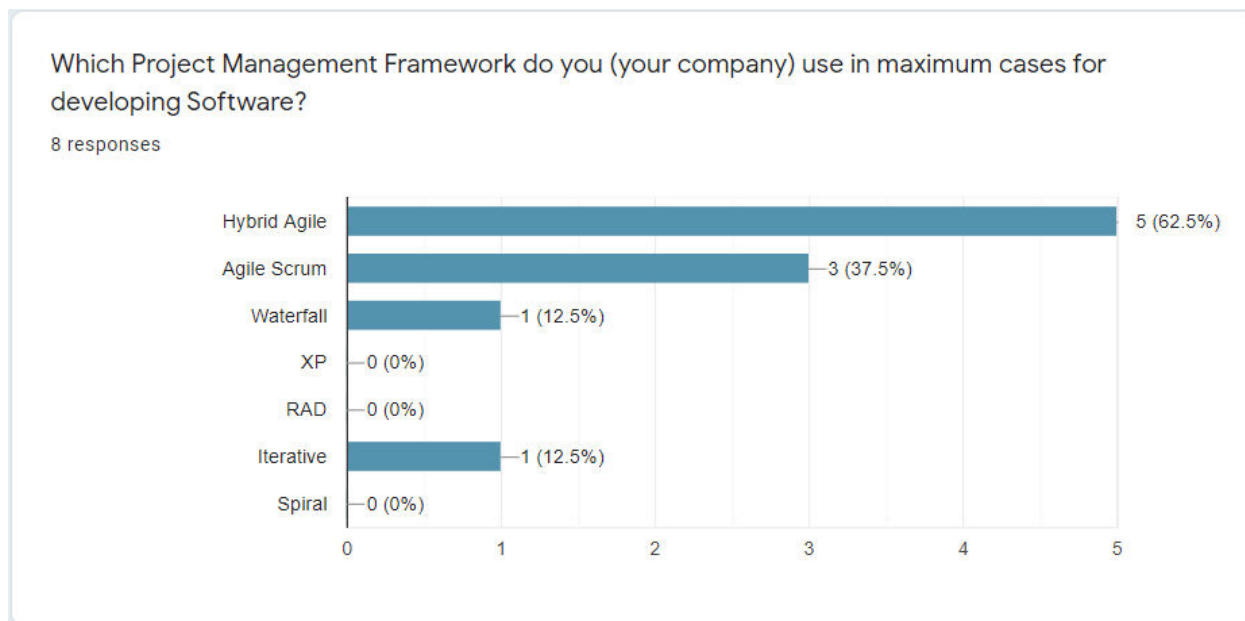
The ultimate objective of using these three research methods was to ensure that the proposed framework would be based on real-world practices, lessons learned from experienced professionals, and practical project experiences from a variety of software development environments. The goal was to propose a solution that would not only be theoretically strong but also practically applicable to software companies across different regions.

V. EMPIRICAL STUDY AND PROBLEM IDENTIFICATION

To propose a Framework for Requirement Collection, at first fixed questionnaire based Survey Method is followed. From the Survey Method the following findings are found:

5.1. SURVEY METHOD FINDINGS:

Finding 1: It is found that, maximum software companies use different type of Agile Methodology for managing the development of their software projects.



International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

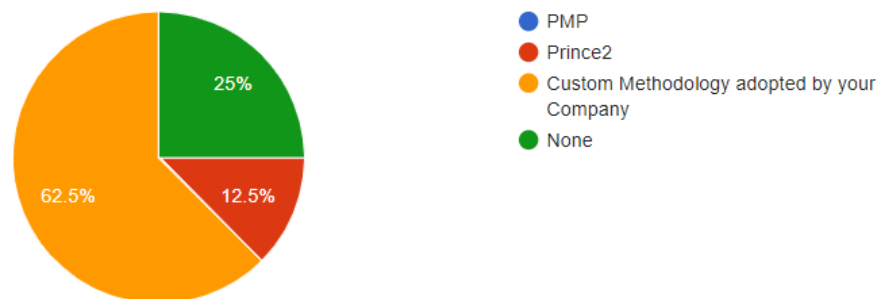
Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Finding 2: It is found that, in Bangladesh maximum companies do not follow any specific Project Management Methodology/Standard.

Which Project Management Methodology/Standard do you (your company follow) for Software Development?

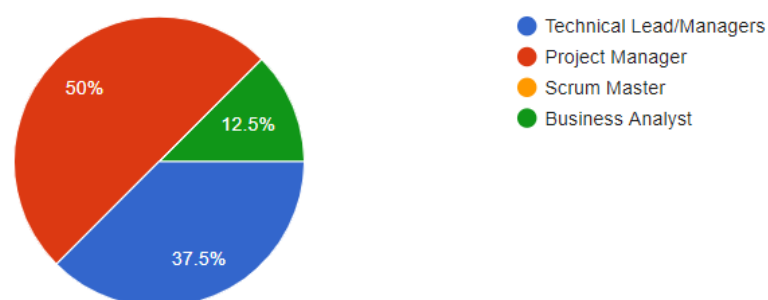
8 responses



Finding 3: It is found that, generally Technical Lead or Project Managers manage the software projects.

Who manages the Software Projects in your Company?

8 responses



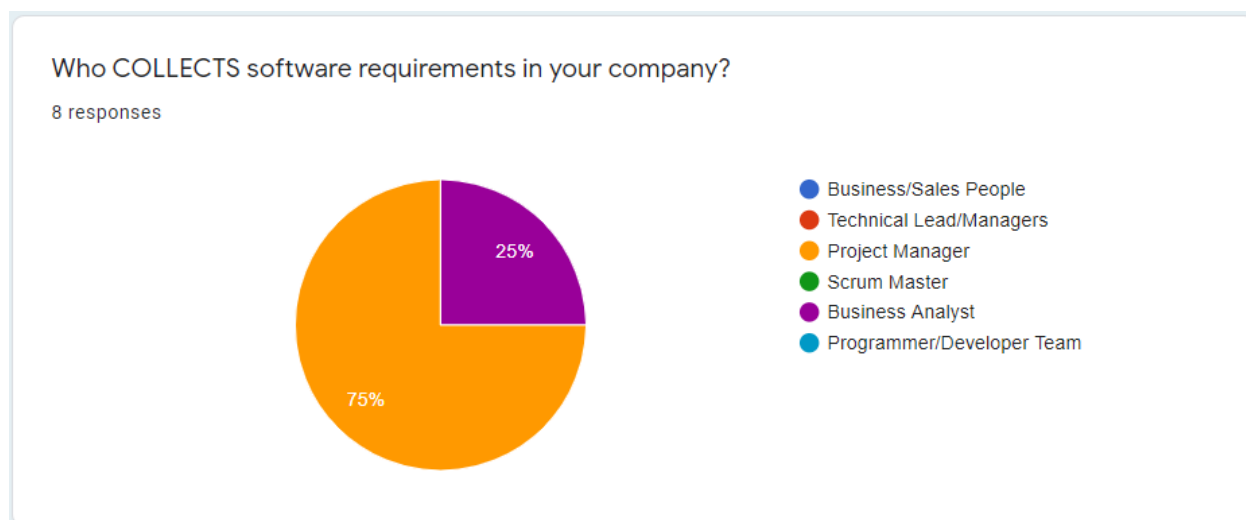
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

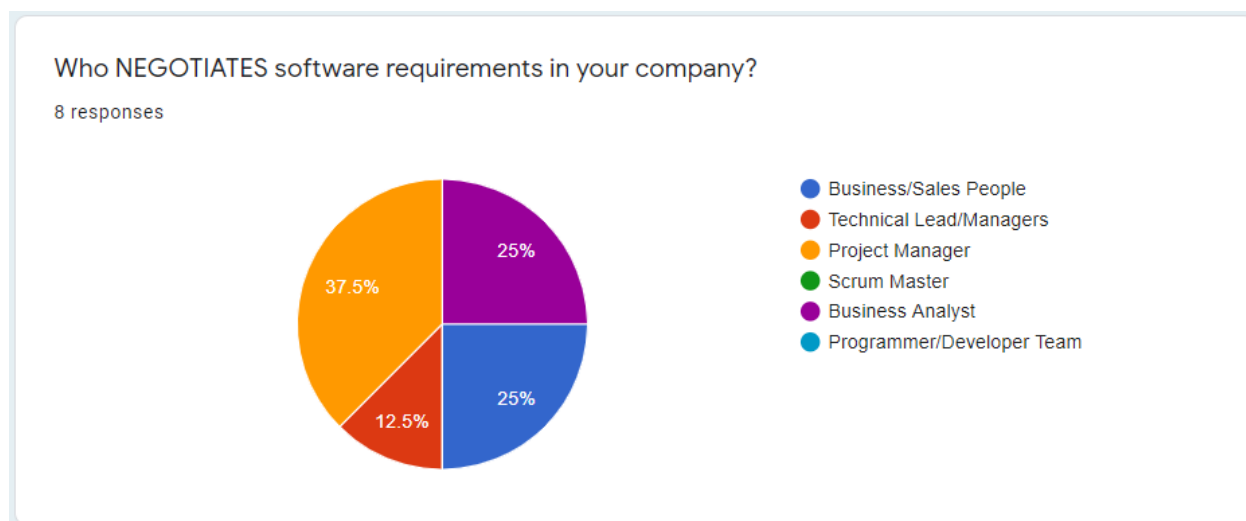
Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Finding 4 (a): It is also found that, it is the Project Manager who collect the requirements in maximum software companies.



Finding 4 (b): It is also found that, it is the Project Manager who negotiates the requirements in maximum software companies.



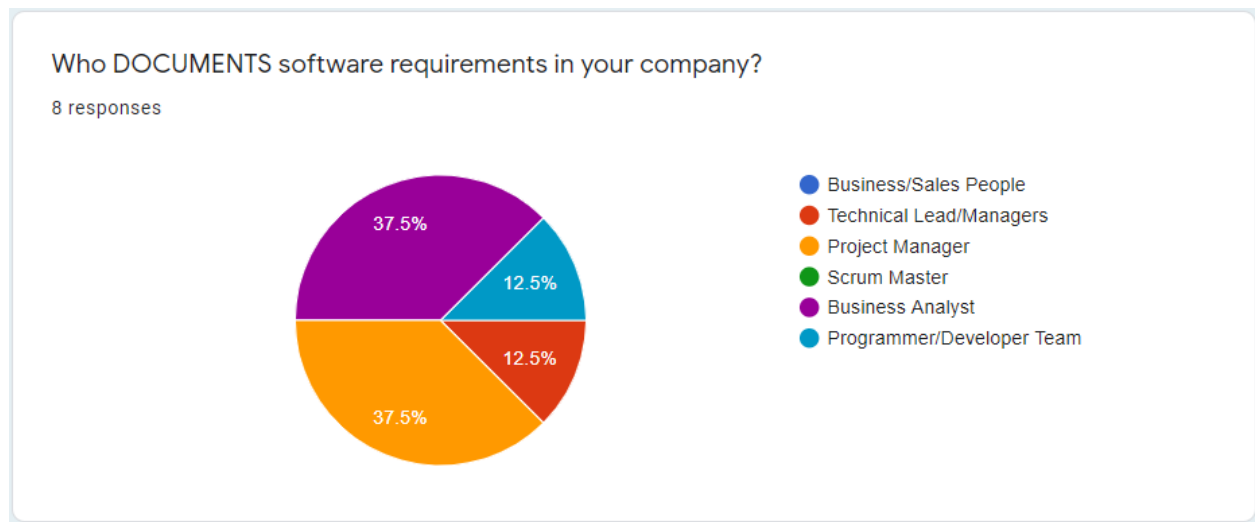
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

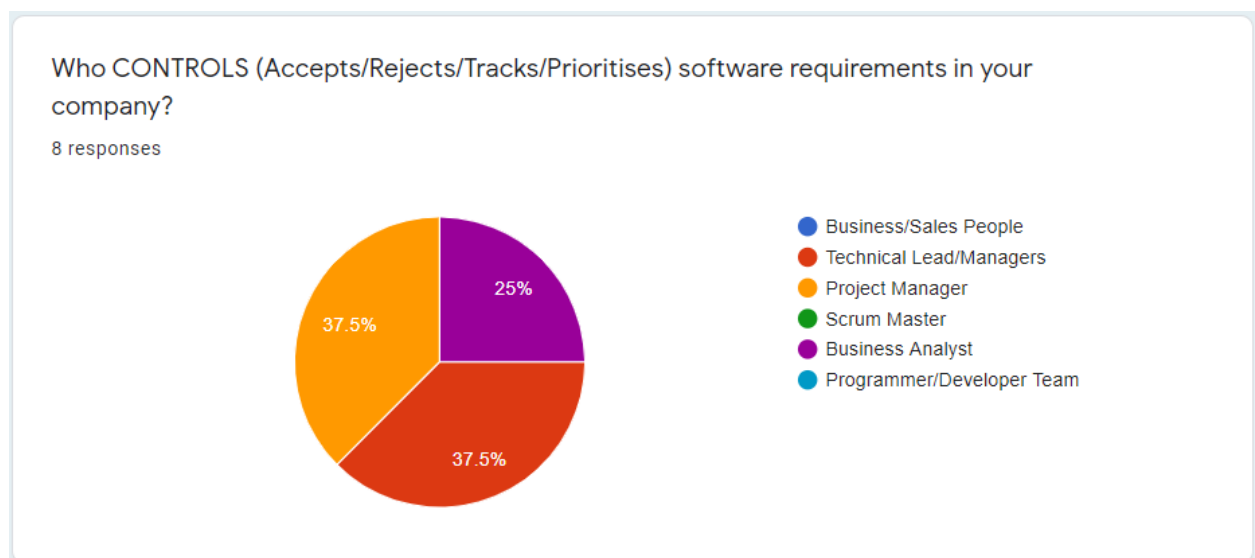
Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Finding 5: It is also found that, the Project Managers in maximum cases document the requirements from clients:



Finding 6: It is found that either Project Managers or Technical Managers control the requirements.



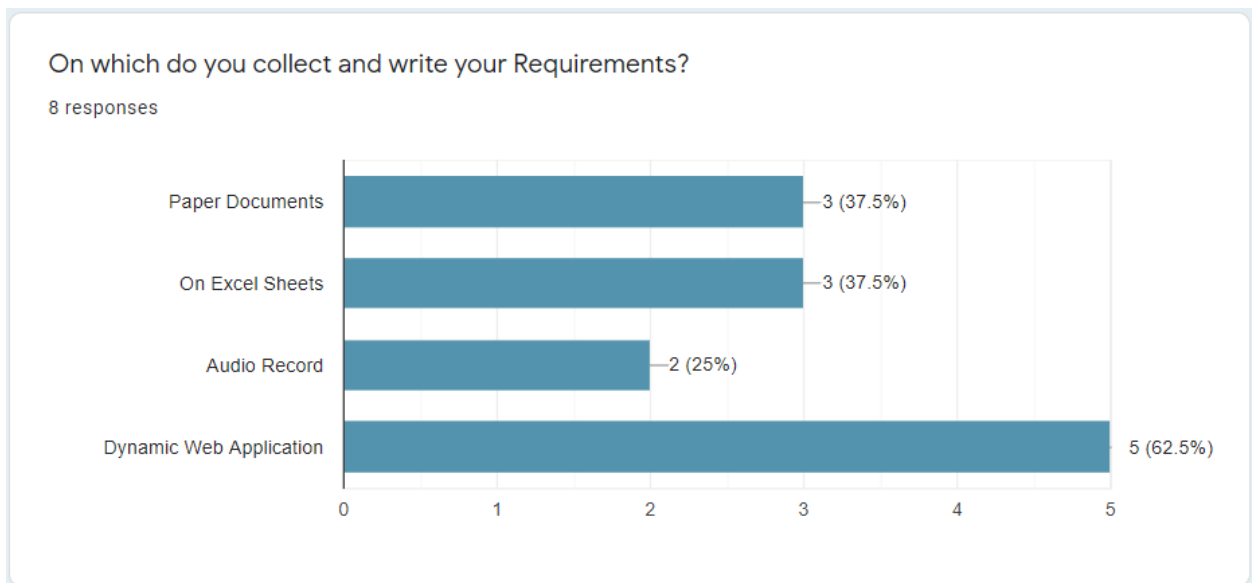
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

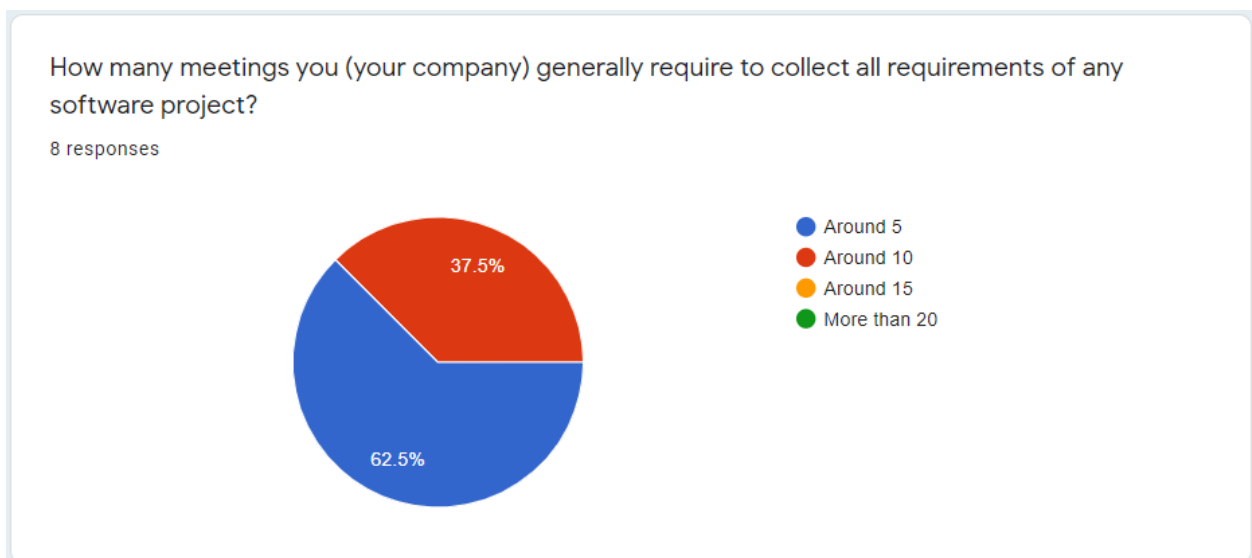
Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Finding 7: Software requirements are collected and recorded over Digital Tools.



Finding 8: It 5-10 meetings require to collect all requirements of a software project from client.



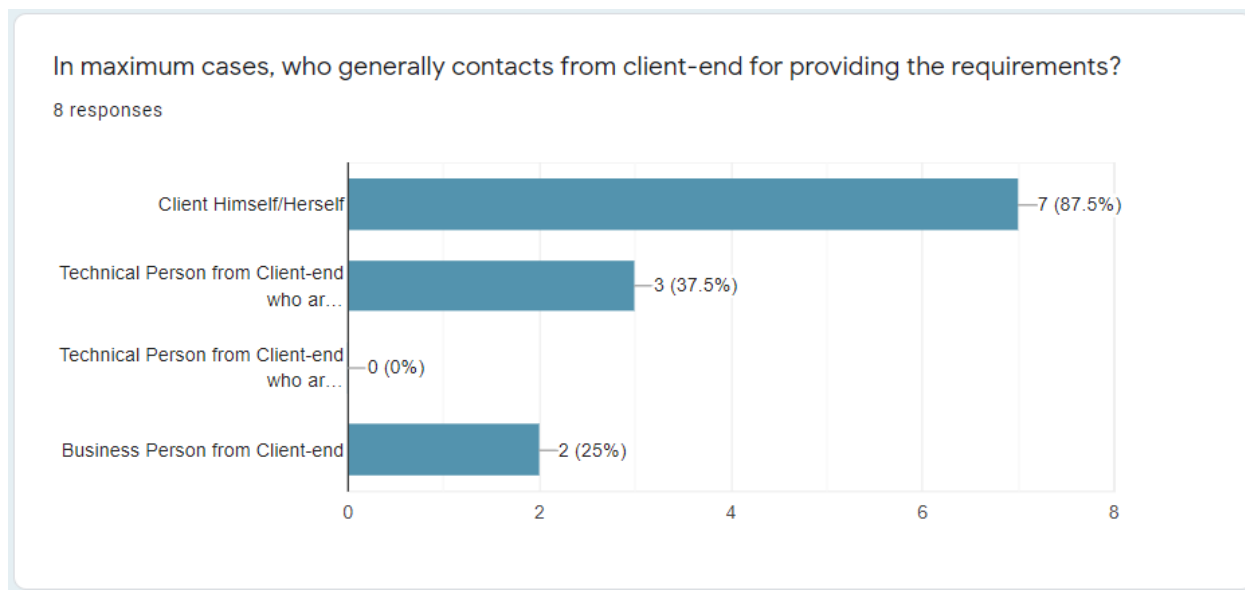
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

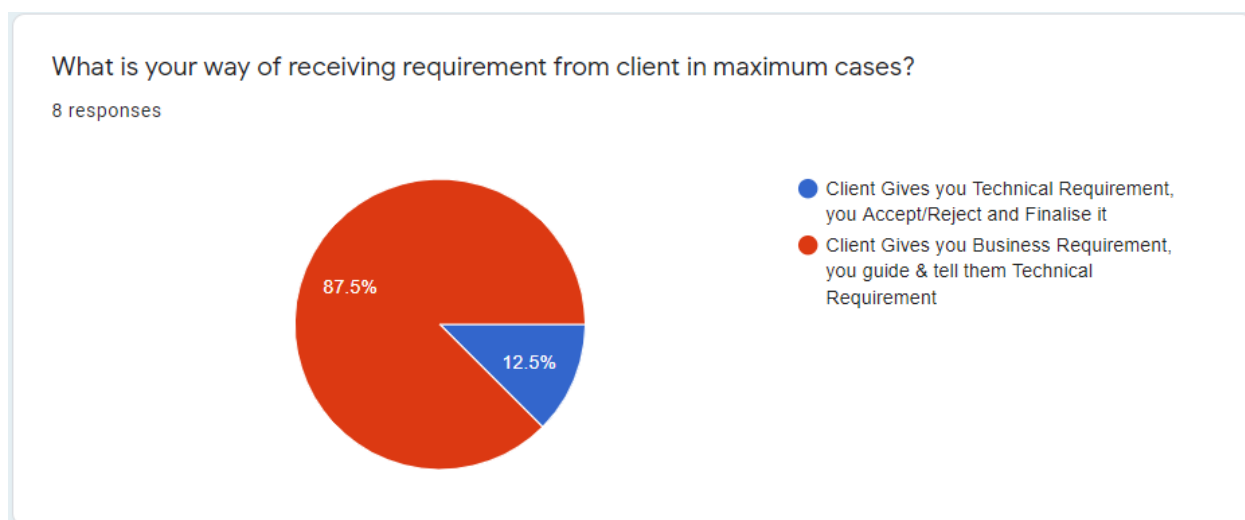
Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Finding 9: Generally, client provides the requirements of his software. Therefore, non-technical people provide software requirements.



Finding 10: It is also found that, client provides business (non-technical) requirement and software companies provide technical requirements based on the business need of client.



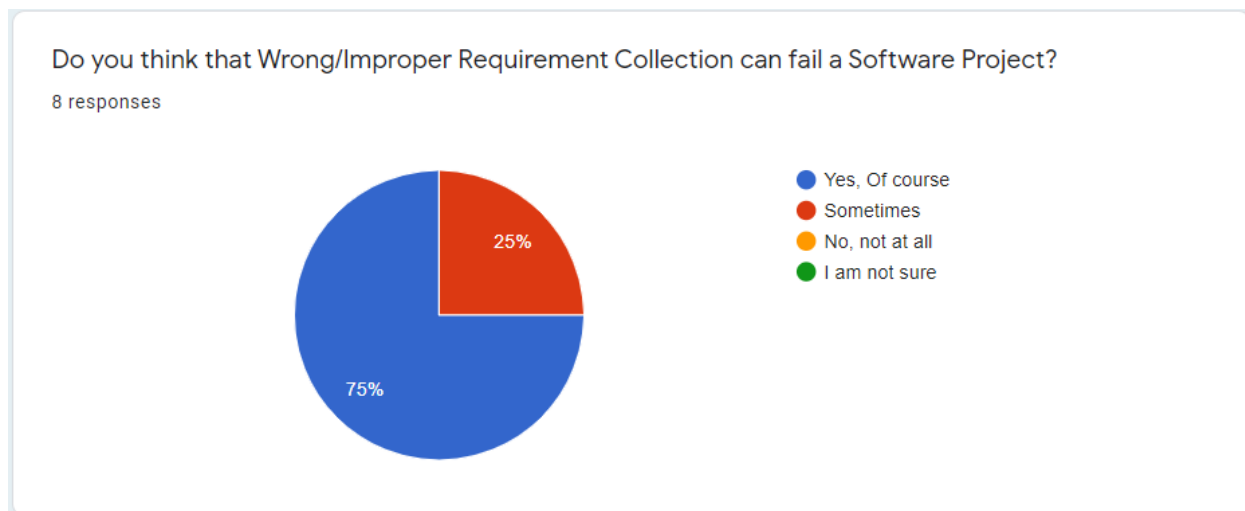
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

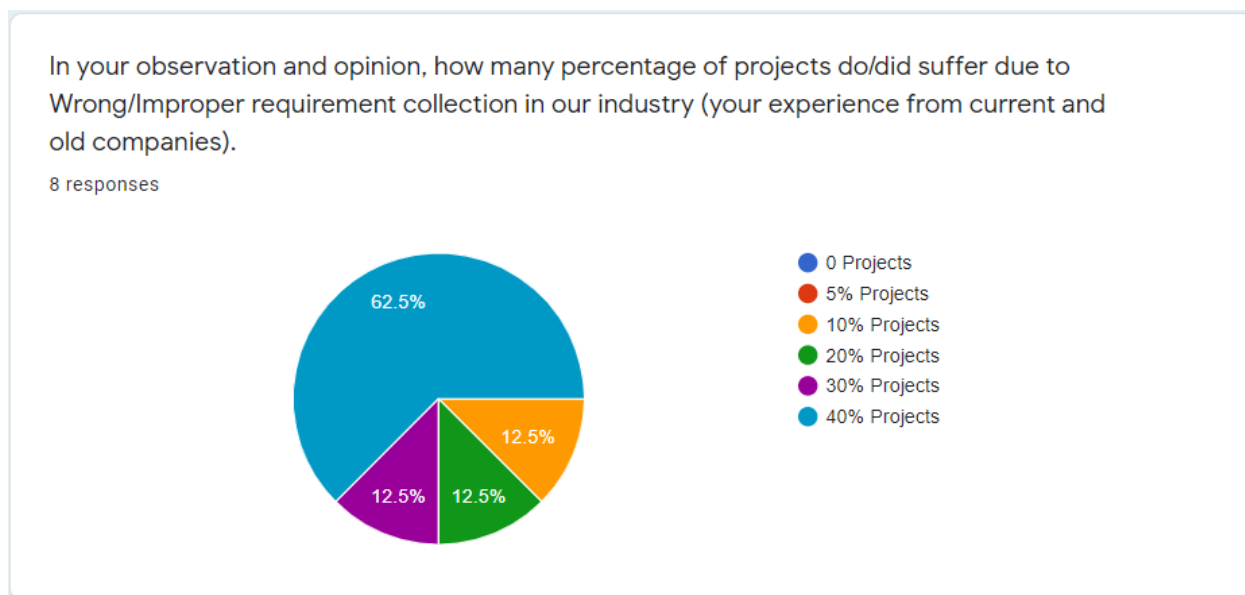
Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Finding 11: Software company experts agree that, wrong or improper collection of requirements can cause failure in software projects.



Finding 12: Experts say that, 10-40% software projects fail due to collection of wrong or improper requirement from customer.



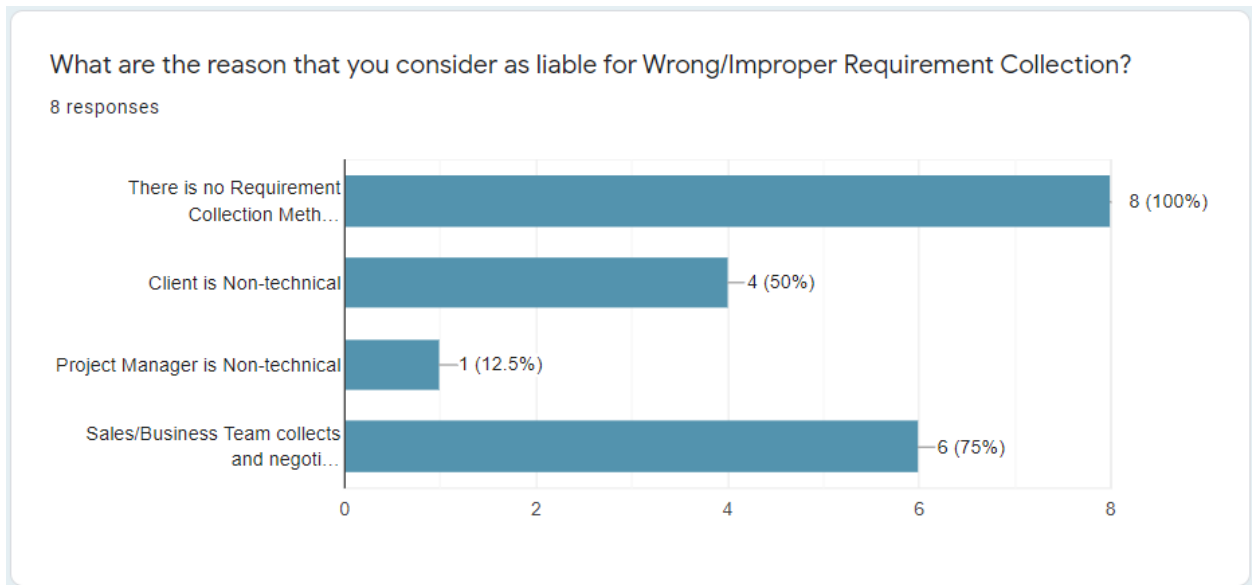
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

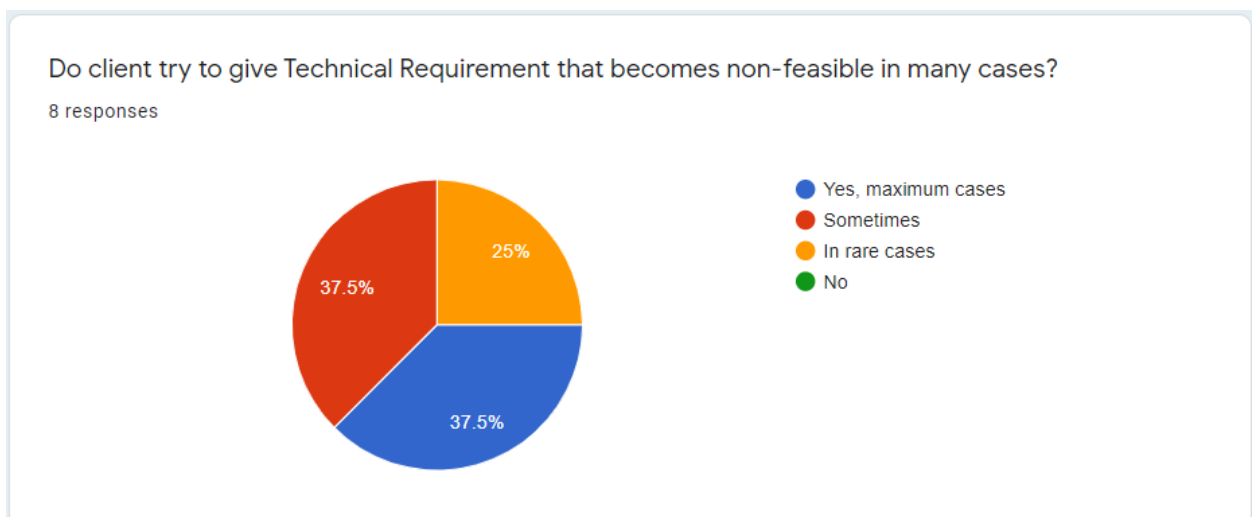
Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Finding 13: Industry experts say that, there is no reputed and worldwide established and used Requirement Collection Framework or Methodology available in the industry.



Finding 14: It is also found from majority expert opinions that, client tries to provide Technical Requirement while sharing their Business Requirements in their software, which is a wrong approach from client, and causes threat for the software project.



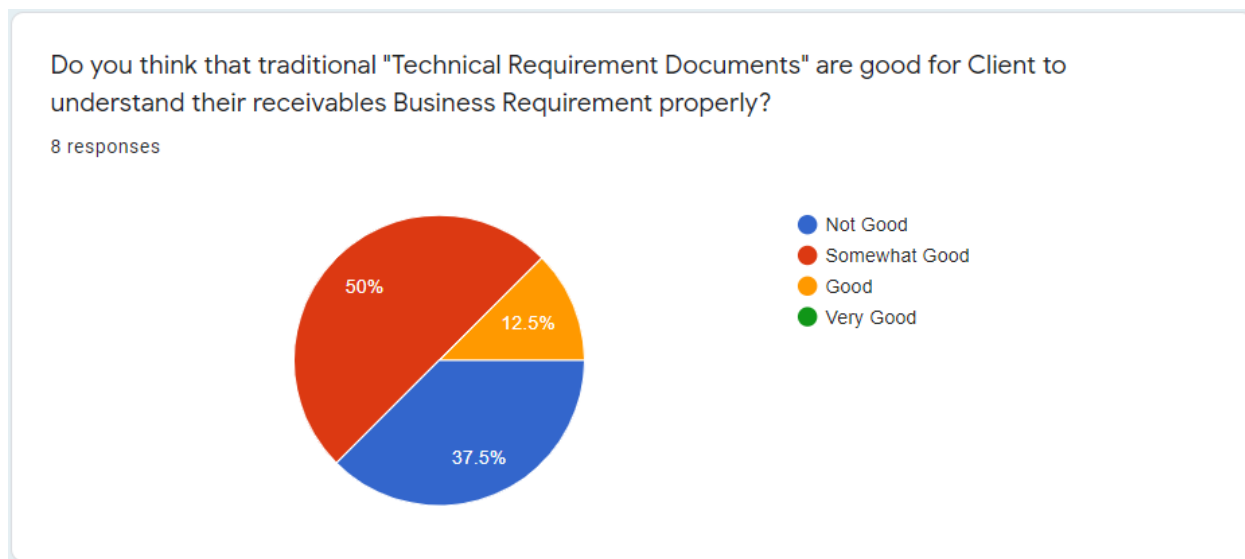
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

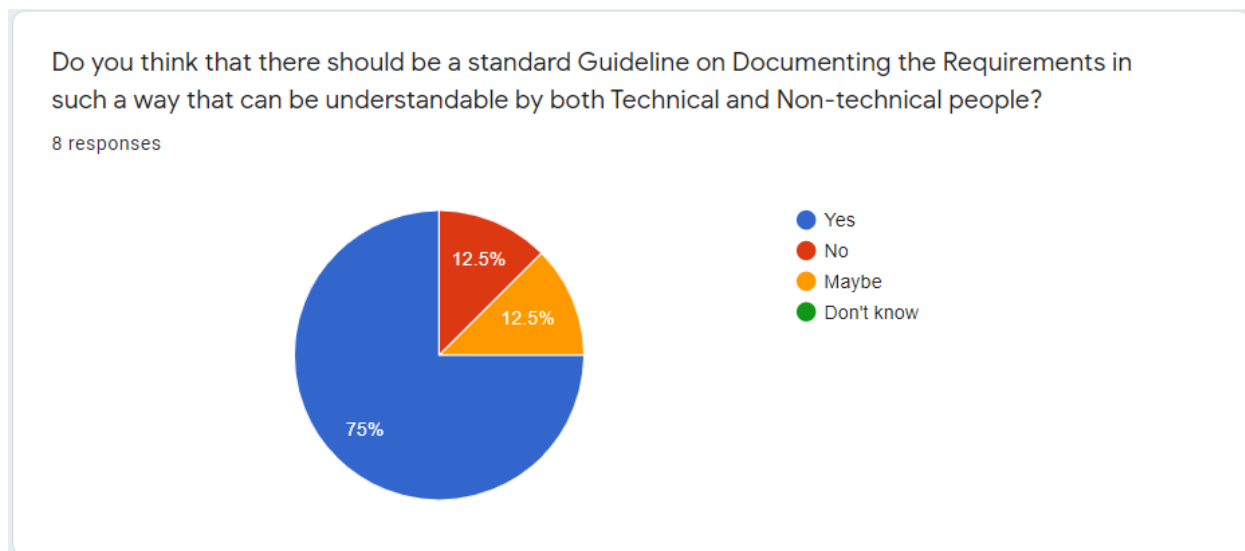
Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Finding 15: Maximum experts say that, traditional way of requirement documentation is not being a much fruitful way of requirement collection.



Finding 16: Software industry experts expect that, there should be a standard guideline on how to document the requirements properly for software.



International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

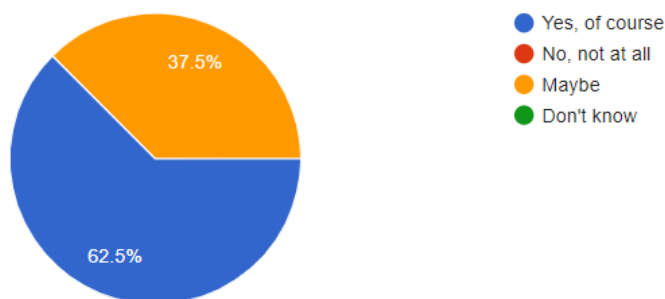
Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Finding 17: Maximum software companies suggest that, there should be a person in every software company who would understand Business requirements and Technical requirements both.

Do you think that, the Software Projects should have a person as Requirement Engineer/Manager who can collect, assess, negotiate, finalise and control requirements throughout the project who knows Technical and Business skill both?

8 responses



Finding 18: All industry experts say that, whenever any requirements raises, those should be documented properly.

Do you think that in each Part of a Software Development Life Cycle (SDLC) whenever any requirement comes, each of them should be documented?

8 responses



International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

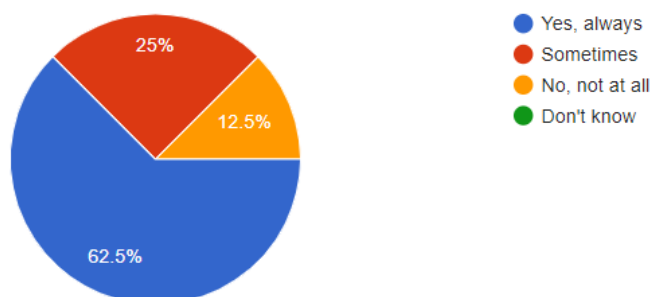
Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Finding 19: Maximum experts from software industry say that, making a design prototype of what they are going to deliver to the client as final product is always good to visualize customer about what they are going to receive finally once the project development is done.

Do you think that, showing Design Prototype of what Engineers are going to build, is a good way of making client understand about the requirements while collecting requirement?

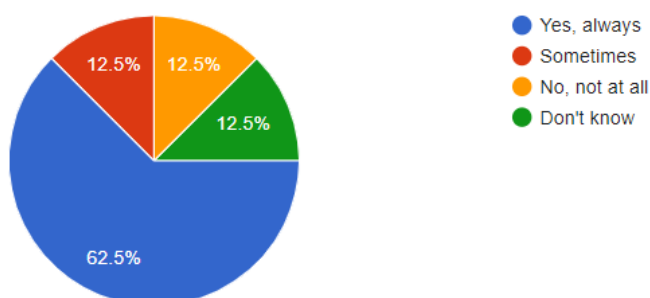
8 responses



Finding 20: Some experts say that, showing similar software or demo also helps client to understand about what they are going to receive once the project development is finished.

Do you think that, showing Similar Solution is a good way of making client understand about the requirements while collecting requirement?

8 responses



International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

5.2. INTERVIEW METHOD FINDINGS:

After finishing the fixed questionnaire based Survey Method a detailed Interview Method was taken to know more about the industry practice requirement collection methods, and finding the problems of existing practices.

Following are the outputs that has come through detailed Interview which was conducted on the experts who participated the survey method as well.

5.2.1. REQUIREMENT DOCUMENTATION PROBLEM:

Maximum software companies do text based requirement documentation, and sign it off by their client. As text has no visualization, client make different meaning of the written text and conflict raises between Software Company (vendor) and Client (customer).

PROBLEM IDENTIFICATION AND SOLUTION:

Textual requirement documentation is not sufficient, there should be design based visualization of requirement that client expects as receivable after project completion.

5.2.2. DESIGN PROTOTYPE PROBLEM:

Some software companies make design prototype of only important parts of the software they develop to visualize their client the ultimate receivable product. But even after showing such design prototypes to client, the software companies cannot control requirement change from their client.

PROBLEM IDENTIFICATION AND SOLUTION:

Only few part's design prototyping is not sufficient to make client visualize about the receivable product. Moreover, without having client's formal signature on the design prototypes, it will never be possible to control change requirements from client.

5.2.3. CHANGE REQUIREMENT MANAGEMENT PROBLEM:

Maximum software companies do not follow any proper documentation on change requirements that they receive from their client. To complete the project and make client happy they receive an unfixed amount of requirements from client. This not only make the project time longer, but also increases expense of the project.

PROBLEM IDENTIFICATION AND SOLUTION:

If we keep receiving requirement from client, the project time and cost and will be increasing, but client will not increase budget accordingly in maximum cases! So, to make the project successful, we need to record all change requirements and need to inform client about the additional cost of these changes.

5.2.4. ADDITIONAL REQUIREMENT MANAGEMENT PROBLEM:

Maximum software companies in many cases accept additional requirements while doing development of a project. Sometimes the software companies do it for convincing client, sometimes they do it to make the project successful and keep client happy. Maximum software companies do not keep track of additional requirements officially.

PROBLEM IDENTIFICATION AND SOLUTION:

If additional requirements are added to the project, the cost of developing the project will be increased. Whether the software company charges for this additional requirement or not, they should record each additional requirement with proper documentation so that they can visualize client that how much additional effort they have made for the project. Sometimes client give additional requirement within the development time-frame and when the project delays they cannot understand why the delay occurred. Software companies should record additional requirements properly so that they can explain how additional requirement has impacted on the delay of delivering project on timeline.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

5.2.5. PROBLEM WITH REQUIREMENT COLLECTOR:

In maximum cases software companies send Project Managers to collect requirement who are not directly involved with project development or coding therefore technology. Again in some other cases few software companies send their business people to record the requirements, which is even a worst case. At the other side, some software companies send their Technical Leads or Development Team people for collecting software requirement. Now the fact is, pure technical people will also not be able to understand a software's requirement from client, because a software's requirement combines both technological and business domain knowledge.

PROBLEM IDENTIFICATION AND SOLUTION:

Software requirement collection and management will not be successful if it is collected by a technology person or a business person. If the requirements are collected and managed by such a person who is knowledgeable in both technology and business. Therefore, the Requirement Management Expert should both technology and business expert. In such case the project will be more successful, because client's requirement will be perfectly recorded and managed then.

5.2.6. REQUIREMENT DOCUMENTATION TEMPLATE:

There is no proper format of collecting requirement. Each software company and each individual related to requirement collection maintain their own way of writing requirements. There is even no industry standard template for collecting requirement.

PROBLEM IDENTIFICATION AND SOLUTION:

Without any proper requirement collection form the requirement collection will become messy. A single format of requirement collection will also be able to manage multiple project's requirement in a well-organized way.

A standard Requirement Collection Form (RCF) can be developed that can be a benchmark to collect requirement from client. Then whether the requirement collector is technical or business person he/she would miss less things as the template/form would guide him/her to collect necessary requirements.

5.3. CASE STUDY FINDINGS:

After finishing the fixed interview method, a case study was collected from industry experts to know more about industry practices in requirement collection methods and to find the problems with existing practices.

Following are the outputs that have come through the case study collection conducted with the industry experts who also participated in the interview method.

5.3.1. CASE OF A FOREIGN WEB PROJECT (India)

One of the industry experts from India shared their story about a foreign web project which was a 10-month-long big project. The budget of the project was around 1 million USD. The client party was non-technical, therefore to avoid communication gaps, many meetings were arranged. Besides, sufficient wireframes, prototypes, and documentation were made to make the client understand what the features were going to be. Still, the project became complicated, the project was delayed heavily, and many confusions were raised.

PROBLEM IDENTIFICATION AND SOLUTION:

As the wireframes, prototypes, and documents were static, the dynamic features did not get properly visualized to the client. Seeing the static images, the client assumed the features one way, while the engineers understood it differently. The industry expert suggested that if interface design is made at the beginning and shown to the client interactively, it helps the client to understand how the dynamic features will behave. This approach would resolve the misunderstanding created by just presenting static screens.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

5.3.2. CASE OF A GOVERNMENT SOFTWARE PROJECT (Bangladesh)

One of the industry experts shared their story about a Bangladeshi government software project which was a 6-month-long large project. The budget of the project was also significant. In this project, the software company's business team communicated with the client initially, and later the software development team sat for requirement collection. The development team collected the requirements carefully. However, later it was found that the client was mentioning missing requirements.

In an audit by the software company, it was discovered that before the work order was signed, the client shared some additional requirements with the business team over email and verbal communication. Those were considered as draft discussions but were not included during the later requirement collection sessions. When the client recalled those missed points during the project, a conflict arose, causing a delay of two additional months.

PROBLEM IDENTIFICATION AND SOLUTION:

Even though the company took sign-off on the detailed requirement specification, the issue occurred because of earlier informal communications which became critical later. Generally, any formal communication can legally become part of the project. This problem could have been solved if the final requirement specification document mentioned explicitly that "there is no other requirement besides what is written here." Alternatively, this issue would not have occurred if the business team had not discussed technical requirements informally before proper requirement collection sessions.

5.3.3. CASE OF A FINANCIAL SOFTWARE PLATFORM (USA)

An industry expert from the United States shared their experience about a financial technology project for a regional bank. The project lasted about 8 months with a budget of 1.8 million USD. The client representatives were senior business executives with limited technical background. Initially, the project team collected detailed written requirements. However, during the development phase, it was found that the client misunderstood key workflow processes.

As a result, the project scope had to be changed multiple times, leading to major delays, additional costs, and dissatisfaction from both sides.

PROBLEM IDENTIFICATION AND SOLUTION:

The problem occurred because the requirements were collected only in text form without showing real-world usage examples. Business processes were documented, but the lack of live mockups and simulations made it difficult for the client to visualize the system's operations. The expert recommended that early-stage interactive prototypes, flow simulations, and step-by-step demonstrations should be included in requirement sessions to avoid business workflow misunderstandings in financial projects.

5.3.4. CASE OF A HEALTHCARE SOFTWARE SOLUTION (United Kingdom)

An expert from the United Kingdom shared the case of a healthcare management software developed for a group of private clinics. The project took almost 12 months and involved a budget of around 2.5 million GBP. The client side was composed mostly of doctors and administrative staff with no technical knowledge.

During requirement collection, many assumptions were made based on verbal communications and informal meetings. However, when the system was finally delivered, there were many mismatches between what the client expected and what the software delivered, especially in patient workflow and billing modules.

PROBLEM IDENTIFICATION AND SOLUTION:

The misunderstanding arose because initial assumptions were not validated properly with the client. Verbal agreements were not documented formally. To prevent such issues, the expert suggested using detailed visual workflows, formal acceptance of each module design, and clear documentation of even verbal discussions through meeting minutes or formal emails. Ensuring continuous client validation at every major milestone would have prevented the confusion.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

5.3.5. CASE OF AN E-COMMERCE PLATFORM (Sweden)

An expert from Sweden shared the story of an e-commerce platform project targeted at local fashion brands. The project duration was about 7 months with a budget of approximately 750,000 EUR. Although the clients were experienced in the fashion business, they were not familiar with technical aspects like database structures, customer journey design, or backend functionalities.

While the front-end designs were well received initially, major conflicts arose regarding the back-end features like inventory updates, customer returns, and payment gateway handling. These features were not understood properly by the client during requirement collection.

PROBLEM IDENTIFICATION AND SOLUTION:

The gap occurred because back-end processes were not demonstrated in a client-friendly manner during requirement collection. The expert suggested that in future projects, it is important to use flow diagrams, simple system maps, and process walkthroughs for hidden system operations like inventory management and transaction flows. This would ensure that non-technical clients have a full understanding, not just of what they will see on the screen, but how the system will behave in the background.

VI. PROPOSED FRAMEWORK

After conducting interviews, surveys, and case studies with industry experts, a model has been prepared to collect requirements in the best possible way from non-technical clients. This model presents a structured and clear approach that aims to minimize misunderstanding and improve the accuracy of requirement collection. It is designed specifically to help project managers and requirement engineers gather technical requirements effectively from customers who may not have a strong technical background.

The model begins by arranging a meeting with a Requirement Engineer who has knowledge of both business processes and technology. Requirements should be collected not only in a documented format but also in an audio or visual format to ensure clarity. A detailed requirement document must then be created, explaining every section or page view feature clearly and systematically. Instead of relying only on wireframes, it is recommended to prepare full design prototypes that better illustrate the final product to the client.

Before developing individual project modules, the complete user interface should be created and shown to the client for review and formal approval. This step ensures that the client fully understands what they will receive at the end of the project. During the requirement collection phase, no other teams should communicate requirements-related information to the client to avoid confusion. Finally, once all requirements have been gathered, the client must formally sign off the final requirement document, with a clear statement that no additional requirements exist outside the signed document.

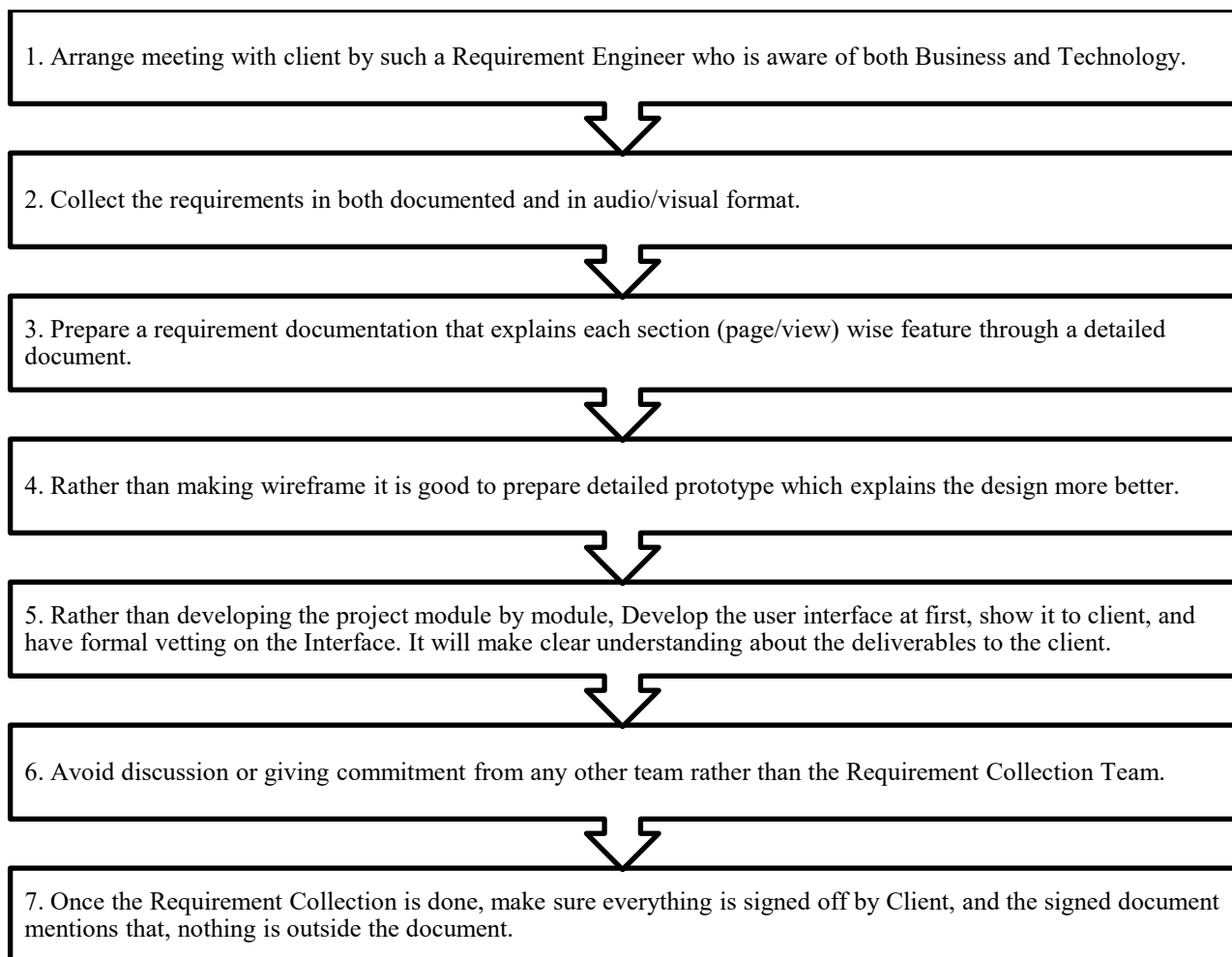
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

The following flow diagram illustrates the detailed steps of this proposed requirement collection model:



VII. LIMITATIONS AND FUTURE WORK

Although this research proposes a structured framework for collecting software project requirements from non-technical customers, there are some limitations that should be noted. Understanding these limitations is important because it can help guide future improvements and further research in this field.

One major limitation of this study is the sample size of the primary data. Surveys, interviews, and case studies were conducted with project managers and experts from five countries, namely the United States, United Kingdom, Sweden, India, and Bangladesh. Although these countries represent a wide range of software development environments, the study included only five companies and five key projects. While the selected companies were well-established and diverse, collecting data from more companies, industries, and client types could give even better and broader insights. Future research could expand the sample size by including additional companies and participants to strengthen the findings.

Another limitation is that the research mainly focused on the early stage of requirement collection, particularly before the software development starts. It did not cover much about the challenges that arise when requirements change

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

during the ongoing phases of project development. In many real-world software projects, customers often introduce new requirements or request modifications after the initial requirements have been collected and approved. Managing these changes is a critical aspect of software engineering, and future work could explore how to extend the proposed framework to better handle dynamic and evolving requirements (Bano & Zowghi, 2015).

Moreover, while the proposed framework emphasizes structured requirement collection through documentation, visual prototypes, and formal sign-offs, it may not be fully suitable for every kind of project. For example, projects that are extremely agile, fast-paced, or experimental may require more flexible approaches where locking requirements early is not always practical. Future research can focus on adapting the framework for projects where rapid change is expected and flexibility is essential (Zowghi & Coulin, 2005).

Another important point is that this research relies mainly on qualitative methods, such as expert interviews, surveys, and case studies. While these methods provide deep and valuable insights, they are subjective and based on personal experiences and opinions. A limitation of this approach is that it lacks strong statistical validation. In future work, researchers could use quantitative methods, experimental studies, or statistical models to test and measure the effectiveness of different requirement collection methods in a more objective way (Bubenko & A. Jr., 1995).

Additionally, the framework assumes that customers will actively participate in the requirement collection process, provide detailed feedback, and carefully review documents. In other different real-life projects, however, clients may be very busy, may not give enough attention to the requirement documents, or may approve documents without properly understanding them. Future studies could investigate how to better engage customers, motivate them to participate more actively, and ensure that requirement collection sessions are more effective.

Finally, even though this research covers companies from five different countries, it should be remembered that culture and economy plays a major role in software development practices. The way people communicate, negotiate, and approve requirements can vary significantly based on cultural norms and organizational styles. Future research could study how the proposed framework needs to be adapted for different cultural and organizational contexts to make it even more effective globally.

While this paper proposes a strong and structured framework for collecting technical requirements from non-technical customers, it also opens many opportunities for future research. Expanding the sample size, handling evolving requirements better, validating the framework through quantitative studies, improving customer engagement, and addressing cultural differences can all contribute to making the framework more robust and widely applicable. Continued research in these areas will help the software development industry achieve even greater success in delivering projects that meet customer expectations.

VIII. CONCLUSION

Requirement collection is one of the most critical activities in the Software Development Life Cycle (SDLC). The success or failure of a software project often depends heavily on how accurately the requirements are collected at the beginning. However, collecting technical requirements from non-technical customers remains a major challenge in the software industry across the United States, United Kingdom, Sweden, India, and Bangladesh. Non-technical customers often find it difficult to express their needs in technical terms, and project managers who are not fully technical sometimes struggle to translate these needs into complete and clear requirements.

This paper has identified the existing problems in requirement collection practices through detailed surveys, interviews, and case studies. It was found that issues such as misunderstanding textual documentation, improper change management, the lack of formal prototyping, and the wrong choice of requirement collectors are common across the industry. These challenges not only cause project delays and cost overruns but also lead to dissatisfaction among customers and software development teams.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 7, Issue 11, November 2018

Based on the findings, this research has proposed a structured framework for collecting requirements more effectively from non-technical customers. The framework emphasizes involving a Requirement Engineer who understands both business and technology, using both written and visual communication methods, developing full user interface prototypes early, maintaining strict communication discipline, and obtaining formal sign-offs from customers. This proposed model aims to reduce miscommunication, improve project clarity, and ensure that customers and development teams have a shared understanding of the final product.

While the framework offers a practical solution for improving requirement collection, the research has also highlighted certain limitations. These include the limited sample size, the focus mainly on the initial stages of requirement gathering, and the need to adapt the framework for highly dynamic project environments. Future work can further refine this model by expanding the research scope, including evolving requirements management, and adapting it to various cultural and organizational settings.

In conclusion, by adopting the proposed framework, software development companies across different countries can significantly improve their requirement collection processes. This will not only enhance communication with non-technical customers but will also contribute to higher project success rates, better customer satisfaction, and more efficient use of resources. This research lays a foundation for further studies aimed at continuously improving requirement engineering practices in the global software industry.

REFERENCES

1. Ågerfalk, P. J., & Fitzgerald, B. (2008). Outsourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy. *MIS Quarterly*, 32(2), 385-409. <https://doi.org/10.2307/25148845>
2. Bano, M., & Zowghi, D. (2015). A systematic review on the relationship between user involvement and system success. *Information and Software Technology*, 58, 148-169. <https://doi.org/10.1016/j.infsof.2014.06.011>
3. Bano, M., & Zowghi, D. (2015). A systematic review on the relationship between user involvement and system success. *Information and Software Technology*, 58, 148-169. <https://doi.org/10.1016/j.infsof.2014.06.011>.
4. Begum, Z., Khan, M. S. A., Hafiz, M., Islam, M. S., & Shoyaib, M. (2009). Software Development Standard and Software Engineering Practice: A Case Study of Bangladesh. *Journal of Bangladesh Academy of Sciences*, 32(2). <https://doi.org/10.3329/jbas.v32i2.2432>
5. Bubenko, J., & A. Jr., J. (1995). Challenges in requirements engineering. In *Proceedings of the Second IEEE International Symposium on Requirements Engineering (RE '95)* (p. 160). IEEE Computer Society.
6. Cusumano, M. A. (2012). *Staying Power: Six Enduring Principles for Managing Strategy and Innovation in an Uncertain World*. Oxford University Press.
7. Demirel, S. T., & Das, R. (2018). Software requirement analysis: Research challenges and technical approaches. 2018 6th International Symposium on Digital Forensic and Security (ISDFS). <https://doi.org/10.1109/isdfs.2018.8355322>
8. Jones, C. (1995). Patterns of Software System Failure and Success. ITP Media.
9. Rajlich, V. 2006. Changing the paradigm of software engineering. *Communications of the ACM*. 49, 8 (Aug 2006), 67-70. DOI=<http://doi.acm.org/10.1145/1145287.1145289>
10. Tiwana, A., & Keil, M. (2004). The one-minute risk assessment tool. *Communications of the ACM*, 47(11), 73-77. <https://doi.org/10.1145/1029496.1029497>
11. Wiegers, K. (1999). First things first: Prioritizing requirements. *Software Development*, 7(9), 48-53.
12. Zowghi, D., Coulin, C. (2005). Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In: Aurum, A., Wohlin, C. (eds) *Engineering and Managing Software Requirements*. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-28244-0_2.