JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 06 Issue 02
2024

Review Article
https://doi.org/10.56979/602/2024

Challenges in Requirements Engineering for IoT Solutions

Muhammad Asgher Nadeem[1], Muhammad Hasnain[2*], M. Mohsin Saleemi[3], Muhammad Awais Mohsin[4], Mohammad Adeel Ansari[5], and Wissal Essalah[6]

[1]Department of Computer Science, University of Sargodha, 40100, Pakistan.
[2]Department of Computer Science & Information Technology, Ghazi University Dera Ghazi Khan, 32200, Pakistan.
[3]Faculty of Computer Information Science (CIS), Higher Colleges of Technology, UAE.
[4]Head of Products, AHAD Softech LLP, Islamabad, 44230, Pakistan.
[5]Dubai Aviation Engineering Projects, 25314, Dubai.
[6]Departement of Computer Science , National School of Electronics and Telecommunications, University of Sfax, Tunisia.
*Corresponding Author: Muhammad Hasnain. Email: hasnainmalik2020@gmail.com

Received: November 18, 2023 **Accepted:** January 26, 2024 **Published:** March 01, 2024

Abstract: The Internet of Things (IoT) defines a network of devices connected to the internet and sharing a massive amount of data between each other and a central location. Requirement Engineering is a widespread field of knowledge that is influenced by a large number of factors, from client's participation to analyst's expertise. In this paper, all the major problems in the field of requirement engineering are categorized on the basis of their area of origination. This paper categorizes these problems in three parts: the analyst's, the client's, and the generic problems. All the major problems of requirement engineering are separated in the above-mentioned categories. We are presenting the possible solutions for IOT issues and challenges.

Keywords: Requirement engineering; Requirement analysis; Challenges; Expertise; IOT.

1. Introduction

Nowadays, Internet of things (IoT) devices are connected, embedded, and producing a substantial amount of data. Hence, causing various network management issues for both academia and industry and opens new research areas and directions. Requirement engineering is influence by a large number of factors from unclear objectives to technology limitations, environmental effects to legislative restrictions, and many more [1]. This paper aims to identify all the major factors that cause hindrance in the successful completion of the process of requirement engineering. In addition, providing appropriate solutions for the problems raised by these factors is also one of the key objectives of this paper.

The purpose of the paper is to review requirement engineering problems with sufficient details of their causes, effects, and appropriate solutions. The paper identifies the origins of requirement engineering problems so that a reader can better understand the roots of these problems and the areas of struggle to eliminate them. In addition, some unavoidable problems in the way of the requirement engineering process are identified, and the techniques to adopt these problems without risking the success of the project are also well described.

The aim is to understand each problem with its source and then provide its best possible solution. That is the reason for which all the major requirement engineering problems are firstly categorized in subsections on the basis of their origin, and then the appropriate solutions are linked with each of the problems [2]. Also, this method helps to separately identify the responsibilities of both the major parties involved in the process: the analyst and the client/clients. With the help of categorization, it becomes easy for a reader to understand and evaluate each problem separately.

2. Requirement Engineering Issues

Requirements engineering is recognized as a critical task for the success of any project. Most project

failures originate from inconsistent, incomplete, or simply incorrect requirements specifications. This paper analyzes the root problems of requirements engineering from several viewpoints.

2.1 Problems at Analyst's Part

It requires a lot of knowledge of the domain and experience for anyone to become a professional requirement analyst. The analyst is the person who is responsible for the timely delivery of the Requirement Specification Document, which is the result of the whole requirement engineering process.

A significant part of requirement engineering problems lies at the analyst's part [3]. Following is the detailed description of the problems that are occurred by the analyst either accidentally or due to the lack of knowledge and experience.

2.1.1 Poor Understanding of Requirements

The job of the requirement analyst is to collect requirements from the stakeholders and describe them in an easy and applicable way. The problem arises when the analyst collects the requirements from the stakeholder, and he/she does not clearly understand the domain of the problem. The Requirement Specification Document will be as ambiguous and fuzzy as the unsure client will be about the requirement domain and objectives [4].

2.1.2 Poor Stakeholder's Identification and Their Functions

There can be more than one stakeholder of a project. The analyst has to distinguish among the stakeholders and their roles. Satisfying stakeholders is crucial for the success of any project. The analyst often fails to correctly identify the actual stakeholders from whom he has to collect the requirements for the system meant to be developed instead, he either uses his own suppositions or collects requirements from invalid sources. As a result, the system developed is either challenged or entirely rejected by the stakeholders [5].

2.1.3 During Process of Elicitation Poor Communication

Requirement elicitation is the first requirement engineering phase, where the clients collect requirements directly. The success of this phase is mainly dependent on the communication level between the analyst and the client [6]. This phase requires certain expertise as well as field experience of the analyst as different clients have different nature and interests. If the analyst fails to develop a consistent and timely mechanism for the meetings and other communication channels between all the clients and himself at the time of requirement elicitation, the requirement specification document will surely be unreliable as well as incomplete.

2.1.4 Poor Requirements Management

Requirement management is the process of documenting, analyzing, tracing, prioritizing, and agreeing on requirements and then controlling change and communicating with the relevant stakeholders. It's a never-ending method throughout a project. The analyst has to maintain and describe requirements in a well-organized manner. If requirements are not appropriately managed, the requirement specification document will either provide less or too much details of the system to be constructed. Poor requirement management can also cause difficulties in describing the objectives of the system [4].

Following are the requirement management problems that must be resolved to ensure the project's successful completion.

2.1.4.1. Poor Requirement Traceability

Requirements traceability is defined as "the ability to describe and follow the life of a requirement in both a forwards and backward direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases). The job of a requirement analyst is far more technical than just collecting and delivering requirements. Instead, he has to understand and describe the life a requirement and its evolution with the passage of time. It is one of the most difficult tasks of requirement analyst to describe the evolving nature of requirements with respect to the time and on-going refinement and iterations [7].

2.1.4.2. Undefined Functional Requirements

The official definition of a functional requirement is that 'it essentially specifies something the system should do. The client hardly tells you what he wants. Now being a requirement analyst, it is your job to distinguish between the functional and nonfunctional requirements. Also, you have to take care that either there is something missing regarding the functionality of the system meant to be developed [8]. Another important point is the purpose of the requirement engineering process is to provide enough knowledge

that can initiate the next phase, which is modeling. The designer acquires you to provide requirements as the functionalities of the system rather than a mixture of ambiguous desires. Hence, the job of the analyst is to collect and describe the complete functional requirements.

2.1.4.3. Poor Requirement Dependency Handling

Requirements dependency is another problem of requirement engineering. Some requirements can't be defined at first; instead, these are derived during the next iterations from the resultants of the previous requirements. It makes it complex to interlink between the requirement during different iterations of the project; also, it is difficult to manage these requirements in different phases [9].

2.1.5 During Proce Analyst Having Poor Knowledge of Domain

An analyst is an individual who performs analysis of a topic. A requirement analyst is person who performs the process of requirement engineering from requirement gathering to the requirement specification to the requirements validation and verifications. Requirement engineering is a broad field that encompasses many sub-fields and models [10]. It takes years of studies and field experience for one to be a professional requirement analyst. An analyst having poor knowledge of the domain is a real threat for the system meant to be constructed as figure 1 shows the methodology for adressing reqirement engineering problems.

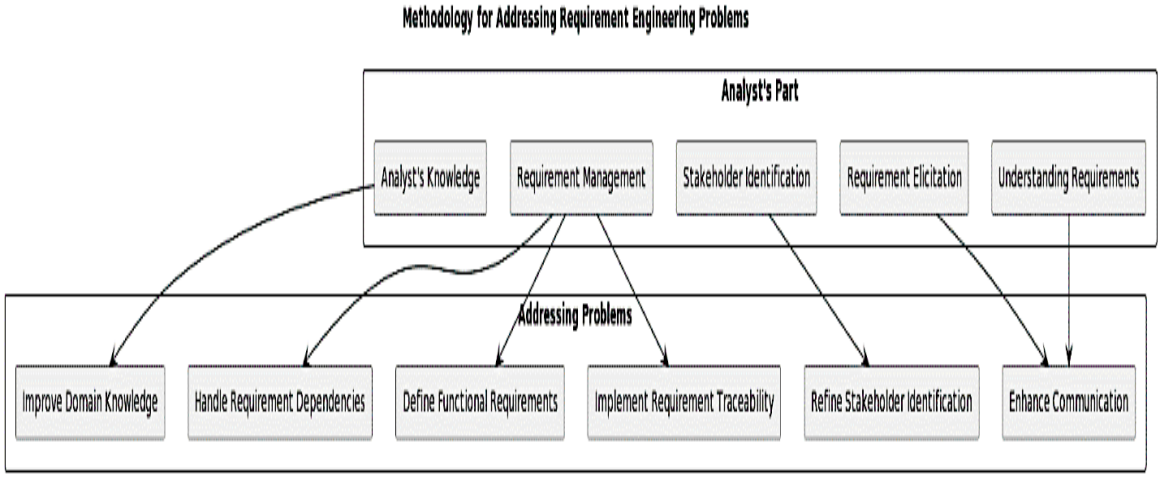

Figure 1. Methodology for addressing Requirement Engineering Problems

2.2 Problems at Clients/Stakeholders

Stakeholders play a vital role in the preparation of accurate and reliable Requirement Specification Document. In requirement engineering, the stakeholders have a continuous role from the beginning, i-e, requirements elicitation to the end of the practice i-e validation and verification of the requirement. In the section below, we will have a brief look on the problems caused by the stakeholders [11].

2.2.1 Collection of unclear requirements

One of the significant problems of requirements engineering is client's ambiguity. The customer does not know what he wants. Also, the client is unaware of the computer capabilities and limitations. As a result, he has a set of unordered desires to present in front of an analyst. Hence, it is a tricky business to derive the actual project requirements from the client who provides only a slight overview of the system and even he is not sure about that [12].

2.2.2 Lack of Client's Interest and Participation in the Process

Sometimes the analyst fails to deliver a reliable Requirement Specification Document because the clients do not pay interest in the requirement engineering process. They are too busy in their daily lives that they can't manage to provide appropriate and enough durations for the meetings with the analysts and the managers [13]. As a result, the schedules and milestones for the project have to be altered which can cause the project outcome to become less reliable and, in some cases, a complete failure.

2.2.3 Poor Understanding of System and Technology Limitations

The client is hardly aware about what he wants as an outcome. He does not know about the technical difficulties and limitations of the platform. He will give you a slight and still unsure overview of the system from which the analyst has to derive the supposed system's functionalities. He does not know that even it is possible to accomplish these requirements with the available system and technology [14]. Customer who is unaware of the system and technology limitations can provide a mistaken overview of the targeted system, which is a real problem.

2.2.4 Conflicting viewpoints among the team members.

When there is more than one stakeholder, the problem arises when different stakeholders provide different requirements, and it is obvious that these requirements start to conflict [15]. Hence, it is quite difficult to satisfy all the stakeholders at the same time or to convince one of them to compromise on his requirements to progress further in the process.

2.2.5 Prioritization of Requirements

Another client's responsibility is to emphasize the priority order of the requirements clearly. The client must evaluate the essential requirements among the whole bunch of requirements and clearly describe it to the requirement analyst that these problems are critical for the system to be developed [16]. The client is often either unaware of the critical requirements of the project or he could not emphasize these requirements to the requirement analyst; as a result, the system could not meet its primary objectives as figure 2 illustrates problems of client and stakeholders

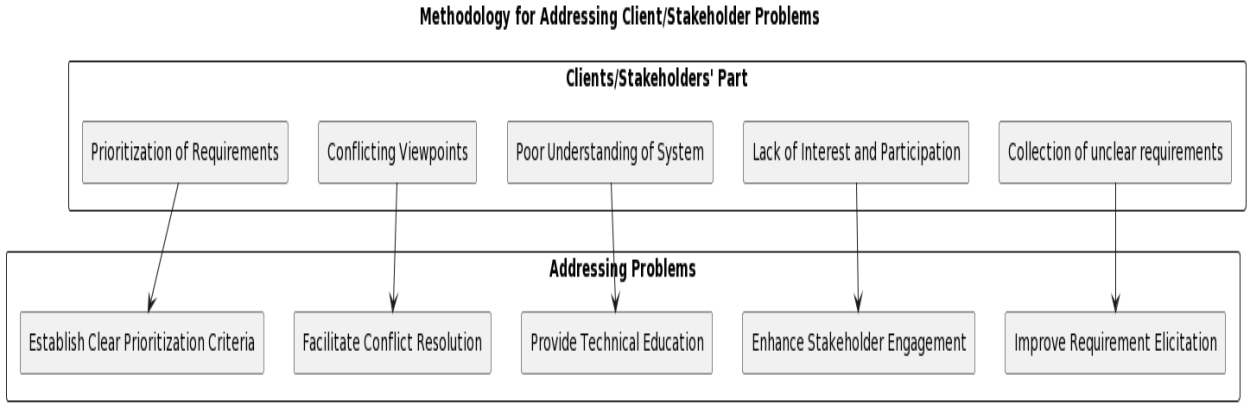

Figure 2. Methodology for addressing client/stakeholder Problems

2.3 Generic Problems

Requirement engineering also has it generic issues whose responsible is neither the analyst nor the stakeholder/stakeholders. Although the analyst has to track and manage these issues, we still can't blame the analyst or challenge his/her expertise as the cause of these problems. Some problems are environmental, which will more or less affect the project outcome either negatively or, in rare cases, positively [17].

2.3.1 Unmanaged Scope Creep and Feature Creep

Scope creep (also called requirement creep or kitchen sink syndrome) refers to changes, continuous or uncontrolled growth in a project's scope, at any point after the project begins. Scope creep do have causes like poor change control, lack of initial identification of project objectives, and poor communication between parties, but even still, if all these causes are managed to be carried out, scope creep will occur due to the change in requirements as the project starts [18].

Feature creep is the addition of excessive features to a product or service such that it becomes less valuable to customers. These excessive features can cause the product to become either too complex or more resource-consuming than originally foreseen [19]. The difference between scope creep and feature creep is that scope creep applies to the projects while feature creeps apply to the product. The point to ponder is that neither scope creep nor feature creep are bad. These are managed properly, but unmanaged scope creep and feature creep can cause problems like cost overrun time troubles and project failures.

2.3.2 Requirements Volatility

In most cases, the requirements begin to change during the project. This is known as requirements

volatility. These changes can occur for many reasons, such as technology, organizational, market, economics, etc [20]. The Standish Group study noted the three most commonly cited factors that caused projects to be challenged:

- Lack of user input: 13 percent of all projects
- Incomplete requirements and specifications: 12 percent of all projects
- Changing requirements and specifications: 12 percent of all projects

2.3.3 Linguistic and Terminological Barriers

Requirement engineering involves continuous communication between different parties. Here the linguistic and terminological barriers start to cause hindrance. The different languages of the analyst and the clients are a barrier in the affective communication of both parties. Moreover, different terms have different meanings in different fields. Hence a term given in the requirements by the client may have different sense according to the analyst according to his field [21]. Figure 3 illustares methodology for adressing generic problems

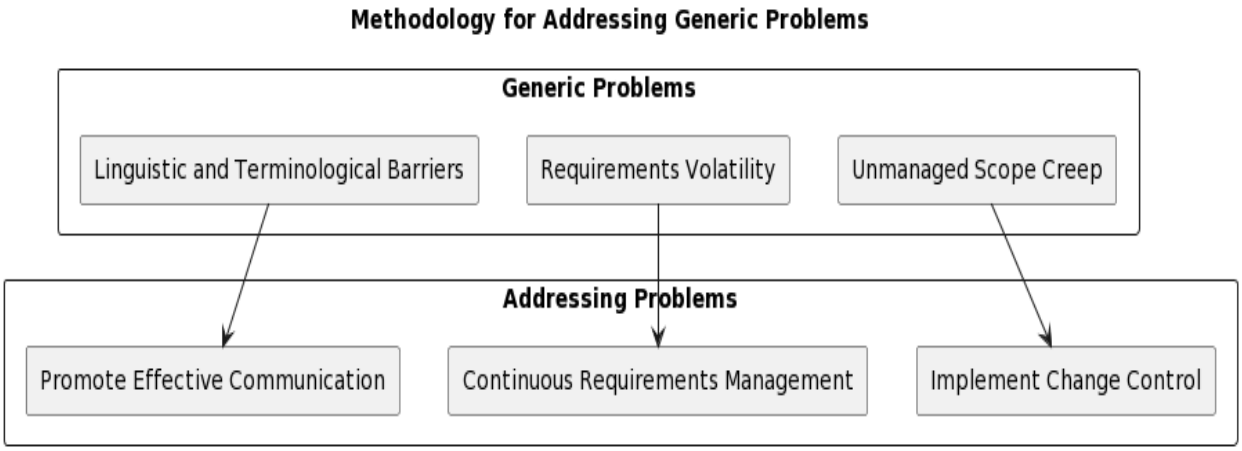

Figure 3. Methodology for Addressing Generic problems.

3. Proposed Solutions

This section will recommend the best methodologies and practices that can be carried out to either prevent or reduce the requirement engineering problems. This section is linked with the previous section, and here we will provide the best possible solutions for each requirement engineering problem described before.

3.1 Solutions to the problems at Analyst's Part

This section aims to provide best practices to tackle the requirement engineering problems at the analyst's end. Following is the detailed list of solutions that can either reduce the requirement engineering problems at the analyst's end or prevent them entirely.

3.1.1 Poor Understanding of Requirements

The way a requirement is worded and understood has a significant impact on the number, quality, and type of proposed ways to achieve the objective of the problem. To completely understand a requirement, the analyst has to clarify the requirement completely. This process can be completed in the following steps.

- **Start with what you know**: The analyst has to clear what information he/she already possesses about the domain of the requirement.
- **Decide what information is missing**: From the information, the analyst knows that he possesses, he has to estimate that what information he lacks regarding the requirement he has to document.
- **Gather the missing information:** Now, the analyst is aware of the information he does not possess. So, he will start gathering the missing information about the requirement.

By passing through this process, it is quite sure that the analyst has spent a fair amount of time on the requirement and has detected and gathered the missing information as well. Hence, we can assure that he will be clear about the requirement domain and objectives now.

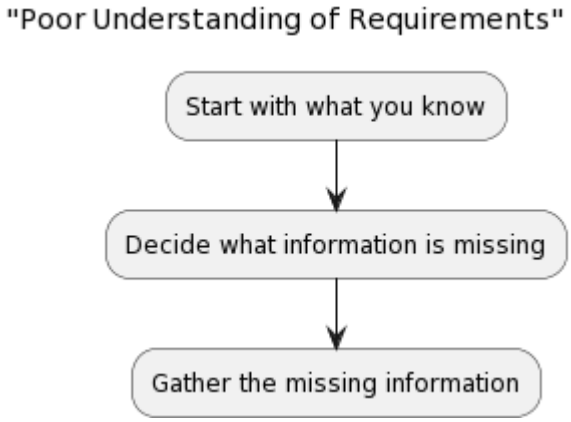

Figure 4. Proposed Solution for Analyst

3.1.2 Poor Stakeholder's Identification and Their Function

It is one of the major objectives of RE practice to satisfy the stakeholders. We can define stakeholder as "A stakeholder is an individual, group or organization who is impacted by the outcome of a project". So, the question arises: "how to correctly identify the true stakeholders and their functionalities?" When the number of stakeholders is greater than one, it is also important to estimate the significance of each stakeholder and his ambitions.

The best thing an analyst can do is to acquire all the stakeholders to create a Stakeholder Representative Group. The Stakeholder Representative Group or Stakeholder Representatives is the group of people chosen to speak on behalf of a large group of stakeholders in a negotiation process. After this, the analyst will be clear from whom he has to hear and to whom he has to inform. This stakeholder representative group works as a bridge between the analyst and many stakeholders [22].

3.1.3 Poor Communication during Process of Elicitation

There can be one or more than one causes at the same time for the poor communication at the time of requirement elicitation. These causes can be the poor communication channel between the parties, the analyst with poor communication skills, the client/clients with poor communication skills, linguistics problem, and terminological barriers. Following are the steps that an analyst can take to eliminate or reduce the problems at the time of requirement elicitation.

- Arrange face-to-face meetings as this is the most effective way of communication.
- Hire a translator to eliminate the linguistic and terminological barriers at the elicitation time.
- Work with requirement engineering experts to observe and adopt their communication standards.
- Value and encourage the client's debate to create a friendly relationship with the client and to avoid hesitation.

3.1.4 Poor Requirements Management

As described before, requirement management is the process of documenting, analyzing, tracing, prioritizing, and agreeing on requirements and then controlling change and communicating the relevant stakeholders.

One of the most efficient and effective way of requirement management is to use the Requirement Traceability Matrix. Requirement Traceability Matrix or RTM captures all requirements proposed by the client or software development team and their traceability in a single document delivered at the conclusion of the life cycle. In other words, it is a document that maps and traces user requirements with test cases. The main purpose of the Requirement Traceability Matrix is to see that all test cases are covered so that no functionality should miss while doing project testing. In this way, RTM ensures that there is no undefined or untested functional requirement, helps to handle the requirements that evolve over time, enables effective handling of requirement dependency.

3.1.5 Analyst Having Poor Knowledge of Domain

Certainly, the RE practice made by a person with poor knowledge of the domain will be questioned.The solution can be none other than the requirement analyst should achieve the knowledge first and then work as a requirement analyst. Following are the steps that an analyst should perform before individually handling the requirement engineering projects.

- The analyst should accomplish some field experience by working with professional RE analysts.

- The analyst should do a deep study of the entire field of Requirement Engineering.
- The analyst should also gather the associated knowledge to the RE, such as business knowledge and environmental knowledge etc.

3.2 Solutions to Problems at Clients/Stakeholders' Part

Providing an effective solution to the problems caused by clients in the process of requirement engineering is mandatory to guarantee the successful completion of any project. This section describes techniques and practices to solve the above-mentioned problems caused by clients/stakeholders.

3.2.1 Collection of Unclear Requirements

The best mechanism to deal with this problem is Prototyping. The prototype is defined as: "a partial implementation of a product expressed either logically or physically with all external interfaces presented". Prototyping is the process of making a prototype. It is not essential for a prototype to have all the functionalities of the targeted system, yet it must enable the client or user to test all the important system functionalities before the actual implementation.

Prototype helps the client to understand the approximate structure and functionality of the targeted system. With this approximate view, the client becomes able to clarify his mind about what he actually wants and what not. As a result, the client approaches the analyst with more precise and accurate project requirements and objectives. In the field of application development, there is also a new generation of tools called Application Simulation Software which can promptly simulate applications before their actual development.

3.2.2 Lack of Client's Interest and Participation in the Process

Clients/Stakeholders participation in the requirement elicitation process is manifest. The solution to this problem is none other than the creation of an effective communication mechanism between the analyst and the client. The analyst should express the significance of the client's timely and adequate participation, and the client has to ensure his appropriate participation and cooperation during the whole process.

3.2.3 Poor Understanding of System and Technology Limitations

Analyst can solve this problem as analyst is or ought to be the person with sufficient and accurate knowledge of his field. He should be aware of available systems and technology and their margins. It is his responsibility to make the client aware of the available system and technology limitations. This problem can only be solved during requirement elicitation. Every requirement given by the client must be confirmed achievable with the help of available systems and technologies before passing to further refinement process.

3.2.4 Conflicting View Points among the Team Members

What if numerous clients give numerous but also conflicting requirements for the same project? These contradictory requirements will make the analyst puzzled and stressed. As a result, the whole later process of RE will produce a disputed paper instead of any useful information.

One thing that analyst can do is to use his convincing communication and ask one of the clients with conflicting requirements to do a compromise for the sake of the project. But the appropriate solution for this problem is also the creation of Customer Representative Group. In this way, all the customers will choose a group of people to talk on their behalf. These customer representatives will first settle the conflicts between the customers and then convey these settled requirements to the analyst. These customer representatives will work as a bridge between the parties, ensuring a more fluent and reliable requirement engineering process progression.

3.2.5 Prioritization of Requirements

Correct identification and understanding of a system's critical requirements or objectives is of vital importance. This problem can be solved by goal-oriented approach. The client has to identify his actual goals and objectives. The client should clearly emphasize on the critical requirements while consulting them with the analyst. The analyst should also adequately compel the client to provide the criticality of the requirements. The use of the Requirement Traceability Index is also helpful in prioritizing the requirements at the analyst's end.

3.3 Solutions to the Generic Problems

This section explains the best methods and practices to resolve the generic problems in the process of

RE described before. Here the point to ponder is that some generic problems cannot be entirely prevented to occur. Instead, we have to find a way that ensures the successful completion of a project even in the presence of these problems. Following is the detailed list of solutions to generic requirement engineering problems.

3.3.1 Unmanaged Scope and Feature Creep

The major causes of scope creep and feature creep are poor scope identification at the beginning and the client's continuously growing wish list. So firstly, the client and the analyst both must be very accurate about their objectives [23]. Also, they have to create a proper mechanism to handle and integrate new requirements into the system. Following is the process that will surely be helpful in reducing the possibilities of scope creep as well as handling scope creep in appropriate manners if it exists.

- Clearidentification of the goals of the client as well as the limitations of the system.
- Collaborative work between the analyst the client to generate a settled solution.
- Definition of the scope of the work that will provide the solution.
- Definition of the mechanism for handling and integrating later requirements.
- Documentation of details of the service delivery in the written agreement.

3.3.2 Requirements Volatility

Adapting to changing requirements is recognized as one of the key project management challenges project managers commonly face. Someone said that prevention is the best medicine. The easiest method to prevent requirement volatility is to make the project scope statement as specific as possible. This works only when the project requirements are considerably consistent and linear.

What to do when requirements are not consistent? It has been observed that requirements volatility becomes an unavoidable problem in many cases. So, we have to follow a way that could enable us to adopt the requirements volatility in a way such that these requirements changes would not risk the successful completion of the project. Adaptation of Agile development method is the best practice in this situation as one of the key principles of Agile Manifesto is: "Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage." Agile development methods allow customer input throughout the process [24]. Requirement Traceability Index, as described before, is also an effective way to handle requirement volatility as it enables us to track the complete life-cycle of a requirement with its evolution phases through which it changed from the starting to the end of the project.

3.3.3 Linguistic and Terminological Barriers

Language and terminology differences are a real threat in the effective and reliable completion of the process of requirement engineering [25]. To tackle this problem, the analyst and the client have to be patient as well as cautious. Both parties have to keep their communication simple and specific. One can ensure to encounter linguistic and terminological barriers by using the strategies below.

- Speak slowly and clearly.
- Ask for clarification.
- Frequently check for understanding.
- Avoid idioms and metaphors.
- Be Specific.
- Hire a translator in case of uncommon understandable language.
- Choose effective communication medium.
- Be patient.

4. Conclusions

We believe that the process of Requirement Engineering is error-prone, and these errors can originate from different sources. These sources, as described before, are: The analyst, the client/clients, and the other generic causes. We can conclude by saying that for successful completion of Requirement Engineering Process and for useful but also reliable Requirement Specification Document creation, all the problems raised from all these sources must be timely identified and tackled. Otherwise, the project can face dreadful results; even these problems can cause the project to become a complete failure.

References

1. P. Achimugu, A. Selamat, R. Ibrahim, and M. N. r. Mahrin, "A systematic literature review of software requirements prioritization research," Information and software technology, vol. 56, no. 6, pp. 568-585, 2014.
2. S. Sharma and S. K. Pandey, "Requirements elicitation: Issues and challenges," in 2014 International Conference on Computing for Sustainable Global Development (INDIACom), 2014, pp. 151-155: IEEE.
3. H. Bani-Salameh, "Towards a comprehensive survey of the requirements elicitation process improvements," in Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication, 2015, p. 60: ACM..
4. J. Dick, E. Hull, and K. Jackson, Requirements engineering. Springer, 2017.
5. R. Snijders, F. Dalpiaz, M. Hosseini, A. Shahri, and R. Ali, "Crowd-centric requirements engineering," in Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on, 2014, pp. 614-615: IEEE.
6. Y. I. Alzoubi and A. Q. Gill, "Agile global software development communication challenges: A systematic review," in Pacific Asia Conference on Information Systems, 2014: AIS.
7. J. Cleland-Huang, O. C. Gotel, J. Huffman Hayes, P. Mäder, and A. Zisman, "Software traceability: trends and future directions," in Proceedings of the on Future of Software Engineering, 2014, pp. 55-69: ACM.
8. S. N. Bhatti, M. Usman, and A. A. Jadi, "Validation to the requirement elicitation framework via metrics," ACM SIGSOFT Software Engineering Notes, vol. 40, no. 5, pp. 1-7, 2015.
9. H. Zhang et al., "Investigating dependencies in software requirements for change propagation analysis," Information and Software Technology, vol. 56, no. 1, pp. 40-53, 2014.
10. A. M. Aranda, O. Dieste, and N. Juristo, "Effect of domain knowledge on elicitation effectiveness: an internally replicated controlled experiment," IEEE Transactions on Software Engineering, vol. 42, no. 5, pp. 427-451, 2016.
11. D. Damian, "Stakeholders in global requirements engineering: Lessons learned from practice," IEEE software, vol. 24, no. 2, 2007.
12. D. M. Fernández et al., "Naming the pain in requirements engineering: comparing practices in Brazil and Germany," IEEE Software, vol. 32, no. 5, pp. 16-23, 2015.
13. A. Ahimbisibwe, R. Y. Cavana, and U. Daellenbach, "A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies," Journal of Enterprise Information Management, vol. 28, no. 1, pp. 7-33, 2015.
14. P. F. Katina, C. B. Keating, and M. J. Ra'ed, "System requirements engineering in complex situations," Requirements engineering, vol. 19, no. 1, pp. 45-62, 2014.
15. G. Gousios, M. Pinzger, and A. v. Deursen, "An exploratory study of the pull-based software development model," in Proceedings of the 36th International Conference on Software Engineering, 2014, pp. 345-355: ACM.
16. A. M. Pitangueira, R. S. P. Maciel, and M. Barros, "Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature," Journal of Systems and Software, vol. 103, pp. 267-280, 2015.
17. Y. Sedelmaier and D. Landes, "Practicing soft skills in software engineering: A project-based didactical approach," in Overcoming Challenges in Software Engineering Education: Delivering Non-Technical Knowledge and Skills: IGI Global, 2014, pp. 161-179.
18. R. Al-Azawi, A. Ayesh, and M. A. Obaidy, "Towards agent-based agile approach for game development methodology," in Computer Applications and Information Systems (WCCAIS), 2014 World Congress on, 2014, pp. 1-6: IEEE.
19. D. Lübke, "Using metric time-lines for identifying architecture shortcomings in process execution architectures," in Proceedings of the Second International Workshop on Software Architecture and Metrics, 2015, pp. 55-58: IEEE Press.
20. T. Kamina, T. Aotani, H. Masuhara, and T. Tamai, "Context-oriented software engineering: a modularity vision," in Proceedings of the 13th international conference on Modularity, 2014, pp. 85-98: ACM.
21. P. Clarke et al., "An investigation of software development process terminology," in International Conference on Software Process Improvement and Capability Determination, 2016, pp. 351-361: Springer.
22. K. Davis, "Different stakeholder groups and their perceptions of project success," International journal of project management, vol. 32, no. 2, pp. 189-201, 2014.
23. F. Cervone, "Effective Project Management," Mistakes in Academic Library Management: Grievous Errors and How to Avoid Them, p. 19, 2009.
24. S. Wagner, D. M. Fernández, M. Felderer, and M. Kalinowski, "Requirements engineering practice and problems in agile projects: results from an international survey," arXiv preprint arXiv:1703.08360, 2017.
25. V. R. Santos, A. L. Soares, and J. Á. Carvalho, "Knowledge sharing barriers in complex research and development projects: an exploratory study on the perceptions of project managers," Knowledge and Process Management, vol. 19, no. 1, pp. 27-38, 2012.