doi.org/10.54473/IJTRET.2024.8408

International Journal of Trendy Research in Engineering and Technology
Volume 8 Issue 4 August 2024
ISSN No. 2582-0958

A NEW STRUCTURE FOR ENHANCED REQUIREMENT ELICITATION OF SOFTWARE PROJECTS

Syed Osama Ali Shah
Bahria University Karachi
s.osamaali72@gmail.com

Received 16 June 2024 Received in Revised form 19 June 2024 Accepted 24 June 2024

ABSTRACT

In order to develop desirable software, the software product development process goes through many phases, each of which must be carried out in a specific and appropriate manner. The phase of software development known as requirement elicitation is one of the most important phases, and production of the software product that the client desires is impossible if requirements are not properly elicited. There are many different techniques for elicitation, and I have proposed a new and advanced requirement elicitation technique in which we divide user requirements into three categories: domain, standard, and bespoke. There are many different techniques for elicitation, and there are also many different techniques available. Each of these categories of requirements is further broken down into their respective subcategories. When we talk about standard requirements, we talk about all of the different standards that are associated with different entities and that need to be followed and implemented in order to produce an application that works properly. We discuss the industrial and environmental requirements that need to be met in the domain requirements section. These requirements need to be met in order to produce the software that is desired for that particular industry, and all software that is related to that industry must fulfil all of the industrial requirements. The operational environment of the application can have a bearing on the environmental requirements that must be met. Our customers always come up with some one-of-a-kind requirements for their product and for the tools that must be used in order to produce the product they want, and we will cover all of these requirements under the category of bespoke requirements.

Keywords: Requirements Elicitation, Project Management, Software Product, Bespoke Requirements, Domain Requirements, Standard Requirements.

I.INTRODUCTION

The phase of the software development process known as requirement engineering is considered to be the most productive. Its goal is to elicit useful requirements from many stakeholders in an appropriate manner. The objective of requirements engineering, often known as RE, is to guarantee that the needs of a system's many stakeholders, such as its users, sponsors, and consumers, are met. It is frequently considered to be one of the oldest activities in the field of software engineering; nonetheless, it has since expanded into a set of activities that now touches practically every phase in the process of software development [1]. Challenges in requirements engineering are a natural consequence of large-scale agile system engineering. Because of the magnitude of these systems, it is difficult for consumers and end users to relate to or provide input on the things that engineers are working on [2]. 2It is essential for every company to produce high-quality software solutions that can cater to the specific requirements of its end users. In the software development process, requirements engineering is a complex activity that evaluates product demands from a large variety of perspectives, roles, responsibilities, and goals. These factors are all taken into consideration. Because of this, it is essential to incorporate requirement engineering principles into every stage of the software development process [3]. The gathering of user requirements through the process of requirement elicitation is an essential step in the development of high-quality software. It is an extremely important phase because, if we are unable to gather and comprehend the requirements in their entirety, not only will we be unable to develop software that will fulfill the requirements of our clients, but it will also hurt the reputation of the company. For example, A software problem was discovered in an F-35 Joint Strike Fighter jet a couple of years ago. The flaw led to planes flying in formation to detect and lock on to the erroneous targets. The business stated that every aircraft in a formation must identify a target from several perspectives. The software, however, was unable to distinguish between a single target and several targets. It's one of the many examples where

the product was unable to meet the requirements of the business [4]. It is important that, software systems be created in such a way that they preserve or increase the sustainability of the socio-technical systems for which they are intended. On the other hand, a paradigm change is necessary to increase the knowledge of software professionals regarding the possible implications of software systems on sustainability. Although Requirements Engineering is widely regarded as the essential component for bringing about this change, it is not possible for requirements engineers to act as facilitators for wider discussions on the effects of sustainability due to their lack of knowledge, experience, and methodological backing [5]. Building an application can be a difficult effort in any industry, whether because there is a lack of requirements collecting or because the developer does not grasp the actual demands of the customer. There is more than one instrument available to choose from when trying to collect the needs of the customers. The requirement analysis step is where the majority of the errors are occurring. It is regarded as a delicate undertaking, as the slightest error or misunderstanding could have severe repercussions for the finished product if not handled properly [6]. There is a vast variety of approaches to need elicitation that may be found and utilized all around the world. A new model for more advanced requirement elicitation has also been developed by me. This model will collect all of the client's requirements in the form of three requirement categories, and the business analyst will be able to grasp the requirements in a more specific manner. Our model is composed of three basic elements, each of which covers a different category of criteria, ranging from functional to nonfunctional. There is the potential for further negotiations despite the time and financial constraints. Domain requirements, standard requirements, and bespoke requirements are the three primary components that make up the entirety of our requirement elicitation paradigm. In this paper, we shall discuss various elements of each of the aforementioned needs. In this lesson, you will learn how the requirements should be collected and analyzed to have the best possible effect on a software product, as well as how the needs should be organized into categories. We will also talk about the enormous impact that each of these needs has on our product, as well as how a project can finally turn into a catastrophe if the elicitation activity is not carried out appropriately.

II.MODEL STRUCTURE

This model breaks the activity of eliciting requirements down into three distinct parts. We can elicit the requirements from the user and then place them into one of the three categories that have been discussed in this paper. For us to develop software that does not violate any community, organizational, industry, or government standards, the user will need to provide us with the standard requirements that must be met. An analyst needs to take all of these considerations into account when working on an application or website. The application or website in question must be connected to a specific industry, and it must also be designed to function in a given environment. Alongside this, the user has some specific tool requirements, such as stack or hardware tools, which need to be fulfilled to build a successful product for the customer that is by his requirements. Both of these sets of requirements need to be satisfied to build a successful product for the customer. To acquire a more in-depth understanding of the model that is the subject of this paper, we are going to conduct in-depth research into all three categories as well as their subcategories. To acquire a more in-depth understanding of the model that is the subject of this paper, we are going to conduct in-depth research into all three categories as well as their subcategories.

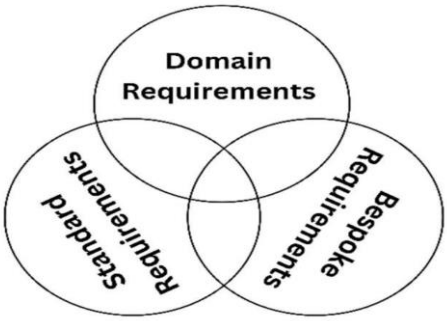

Fig 1.0 Model Structure

(Fig 1.0) explains the main structure of the new model we are introducing through this paper, and it shows us the inter connectivity between the three main elements of our model.

II.1 Standard Requirements

Any piece of software that is developed in any part of the world is required to adhere to certain standards, which may be governed by the government, industry, or the organizations themselves. The business analyst needs to make sure that all of the standard requirements, which will play a significant part in determining whether or not this product is successful, are carefully taken into consideration. In addition to this, the analyst is responsible for considering fundamental standards or ethical requirements that must be met, such as confidentiality, safety, and so on. Without standards, the software will not be able to maintain a long-term presence in the market since there is a possibility that it will encounter resistance from regulatory bodies or end customers who are already making use of the product. Numerous programs are unable to function properly because they do not comply with the requirements. Therefore, in the process of requirement elicitation, one of the most significant activities is collecting all of the necessary standard requirements from the consumers to create a product that won't result in problems in the future. The mandatory implementation of standard requirements is required since doing so prevents fraudulent, unlawful, and unethical behaviors. There will be instances in which the analyst will be required to comprehend the standard requirements on their own because the customer will not make them in open words.

As a result, the analyst needs to maintain a sharp concentration on the words to recognize the keywords that direct attention to the principal standardized requirements for the product that needs to be constructed.

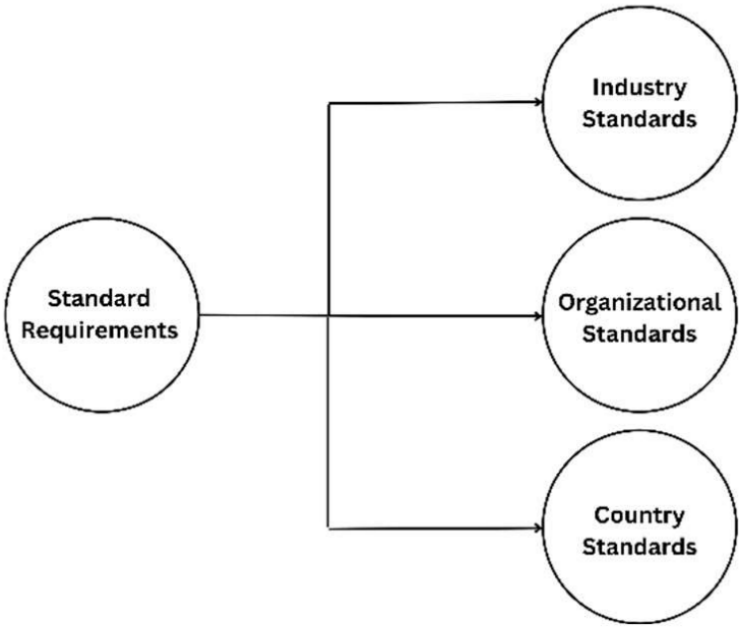

Fig 2.0 Standard Requirement Categories

(Fig 2.0) explains to us the main components of standard requirements. It shows us the 3 main components of the standard requirements.

Industry Standards:

Products based on software are increasingly being developed for virtually every sector. You may find thousands of smartphone apps and websites designed specifically for specific businesses, such as the ones that deal with manufacturing, medicine, architecture, electrical work, and many more. Every sector of the economy has its own set of specific standards that must be met, and one of these is a standard. The standards for one industry can be very different from those of another, and there is room for a lot of variety. When developing a medical application, you will be required to adhere to a predetermined set of criteria and to construct your software under the regulations that have been established by a medical regulatory authority. The same is true for companies

that are associated with any other industries, as well, given that to create an application for their company, they are required to adhere to particular rules and regulations.
All of the necessary industry standard requirements must be satisfied; if a product does not satisfy the necessary standard requirements of a regulatory body, it runs the risk of being inoperable for an extremely extended time until it satisfies all of the standard requirements. Privacy and safety are two important concerns for websites and mobile applications used in the medical business; both of these concerns need to be addressed for your product to operate as intended.

Organization Standards:

Every company and organization operate according to a predetermined set of guidelines, and those standards must be adhered to by any product that is developed for them. Sometimes the customer will explain the organizational standards to the analyst, but other times these criteria are elaborated with the help of nonverbal communication, and the analyst needs to pay sufficient attention to fetch them. Standards within an organization are notoriously difficult to pin down, and because of this, they are amenable to modifications. Moreover, when there is a shift in the leadership of an organization, the new personnel may have significant alterations they wish to see made to the product that was in the process of being developed. Because of this, the software needs to be developed flexibly for it to be able to accommodate alterations that may take place as a result of alterations to the organizational standards or rules. For the analyst to have a better understanding of both the requirements and the organization that has requested the product, they must pay adequate attention to the policies and rules that are in place within the organization. When organizational needs and standards are understood, product development can proceed in a manner that is both more effective and more efficient.

Country Standards:

There are a great number of countries on this planet, and each one of them possesses its own unique set of laws, regulations, and guidelines. If you are manufacturing any product either within the country or for the country, then you are required to follow some rules and laws to be permitted to either produce software or have it operational in that particular country. If you are manufacturing any product either within the country or for the country, then you are required to follow these rules and laws. If a product is to be manufactured in a country or if a software product is to be made operational in a particular nation, both of those countries' individual production and operation requirements for software products need to be adhered to. To construct a product that does not potentially cause issues after postproduction, the analyst needs to collect information from the customer regarding the country (or countries) in which the product will be operational, and all of these particulars need to be meticulously noted down. If the software is being developed in a certain country, then the software firm will need to ensure that the production process of the software adheres to all of the standards for software creation in that country before it can release the software.

II.2. Domain Requirements

Domain requirements are another crucial aspect of the requirement elicitation process. Without prior knowledge of the application's domain, the company won't be able to customize the product to work in the environment or domain that it was designed for in the first place. The domain knowledge includes knowledge of the boundaries of the domain, as well as its vocabulary and alternative structures. With this information, an analyst may more accurately define their expectations for the program's content [7]. The analyst has a responsibility to consider the requirements of the domain because they have the potential to determine the success or failure of the applications. A successful product can only be an application that is suitable for the environment in which it will be used and the domain in which it will be operational. Failure to understand the targeted industry and environment in which the product will be functional is a common cause of product failure. There are many instances in which the analyst was unable to comprehend the targeted industry and environment. If the product is not designed correctly, making improvements will not only take a significant amount of time, but they may also result in a rise in cost in certain circumstances.

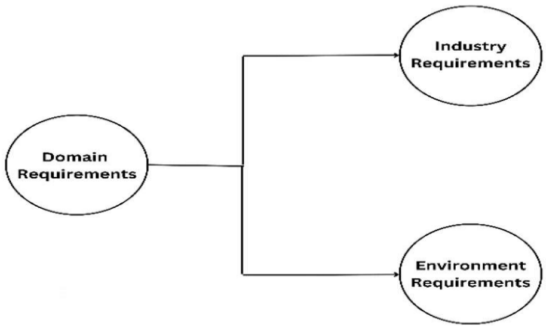

Fig 3.0 Domain Requirement Categories

(Fig 3.0) explains to us the 2 main components that will help us to understand the domain requirements in a precise manner.

Industry Requirements:

Every sector of the economy in the globe has its own distinct set of operating methods, organizational structures, and requirements. As a result, each sector has its own particular set of needs to be fulfilled by a software solution. The commercial and industrial needs must be met by the software products under all conditions, and they should include all of the capabilities necessary to manage all of the operations that are required by the firms operating in that particular industry. These kinds of criteria are always the same for all businesses operating in a given sector, such as if we have to develop an app for an online food delivery service, then it needs to include a product page, a cart feature, a checkout mechanism, a payment gateway, and many other features. All applications for food delivery will include the characteristics that have been described because these qualities are necessary for a food delivery business to thrive in the field of online meal delivery. There may be some particular business needs that need to be met to turn a firm into a distinct and valued offering on the market that can draw customers' attention to itself.

Environment Requirements:

Every company that operates in an industry that requires software will have that software running in a specific setting. If it is developed for employees of the firm, the environment in which the business operates will have a significant bearing on the software that is produced. Every company that intends to release a software product such as an application or website will be required to attach a great deal of significance to the environment in which the product will be functional. Examples of such products are websites and applications. If the product is released into a setting that is not optimal, it will be an unmitigated disaster for the company and will result in a substantial loss for the company. During the requirement elicitation process, the analyst needs to dig out all of the data connected to the environment in which the software will be operational once it has been deployed. This is because the environment is quite critical for any software product. The software product needs to have the appropriate adjustments made to it for it to be desirable for the environment in which it will be used.

II.3 Bespoke Requirements

Every customer is different, and every customer has the same goal, which is to break into the market with a brand-new and original product that has the potential to expand and bring in a large number of consumers. In today's world, bespoke needs are very significant, and they should be implemented in addition to standard and domain requirements. This is because consumers always bring a unique idea to the table, and this idea should be incorporated in addition to the standard and domain requirements. These needs are the most crucial, and they should be carefully documented because they are the distinguishing characteristics of the product that give it its identity. Customers will often make demands regarding the particular instruments that should be used in the production of their product; these criteria must also be satisfied.

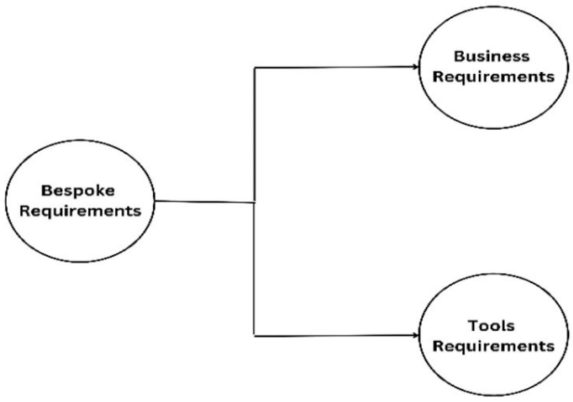

Fig 4.0 Bespoke Requirement Categories

(Fig 4.0) explains to us the two major parts of the bespoke requirements. These two parts will help us in the elicitation of our bespoke requirements from the client.

Business Specifications:

Every company is one of a kind and operates with its specific objectives. When a software product is built for a company or business, they need that product to be a bit unique and better as compared to other similar products that are already available and operational. Companies want their software solutions to meet special, one-of-a-kind requirements so that they can remain competitive in the market and expand their operations. Because they are the primary focus of attention in the software product that needs to be produced, these specifications and domain requirements need to be meticulously documented. The implementation of all standards must be done correctly from the beginning, as doing so afterward may require additional effort, time, and money, and may also be detrimental to the reputation of the software firm that created the software.

Tools Requirements:

Tools are very significant and play a vital part in the creation of software products, yet the tools used for the development of a software product might vary greatly from product to product. At this point, the customer is not technologically illiterate. When a customer contacts a software company, they have already conducted extensive research and know the tools that will be used to build the product that the customer requires. As a result, many clients approach software development businesses in the modern day with a specific list of tools that must be utilized to build their products. These lists can be seen on clients' websites. The customer will provide direction regarding the programming language, hosting platform, and database that should be used when developing their project. Therefore, the analyst is obligated to make a list of all the tools that are required by the client and that will be used to construct the software. If particular tools are not available, the analyst is obligated to negotiate with the client and should also request that the client make modifications to the tool list. if the required tools cannot be used for the production of the software product that is sought, or if the required tools cannot be implemented due to financial or other restraints.

III CONCLUSION

In this article, we explore a requirement elicitation model that can be used by software development businesses to gather needs from customers in a more efficient and forward-thinking manner. This model may be used by software development companies to collect requirements from clients. The three requirements structures that are being addressed in this model completely cover all of the needs that are necessary for the development of a software product, and these requirements may be improved much more by the limits imposed by budget and time. These two limitations will surely have an effect on the product in the end, as well as on our needs, which will be significantly impacted by them. The model that was presented in the study has the potential to be improved further, and in the future, many new points might be included in this work to make it a better and more preferable model for the elicitation of requirements.

REFERENCES

[1]. Bennaceur, A., Tun, T.T., Yu, Y., Nuseibeh, B. (2019). Requirements Engineering. In: Cha, S., Taylor, R., Kang, K. (eds) Handbook of Software Engineering. Springer, Cham. https://doi.org/10.1007/978-3-030-00262- 6_2

[2]. Rashidah Kasauli, Eric Knauss, Jennifer Horkoff, Grischa Liebel, Francisco Gomes de Oliveira Neto, 'Requirements engineering challenges and practices in large-scale agile system development', Journal of Systems and Software, 172(2021)110851

[3]. D. Pandey, U. Suman and A. K. Ramani, "An Effective Requirement Engineering Process Model for Software Development and Requirements Management," 2010 International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, India, 2010, pp. 287- 291, doi: 10.1109/ARTCom.2010.24. Available: https://ieeexplore.ieee.org/abstract/document /5656776.

[4] **Nate Nead** , "10 Examples of Software Development Failure," ReadWrite, Aug. 2023 [Online]. Available: https://readwrite.com/software-development-failure/

[5]. Leticia Duboc, Birgit Penzenstadler, Jari Porras, Sedef Akinli Kocak, Stefanie Betz, Ruzanna Chitchyan, Ola Leifler, Norbert Seyff, Colin C. Venters, "Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow" Requirements Engineering," 25(2020) 469-492. https://doi.org/10.1007/s00766-020-00336-y

[6]. Mrs. M. Geogy and Dr. A. Dharani, "A Scrutiny of the Software Requirement Engineering Process," Procedia Technology 25(2016)405-410

[7].S. Rugaber, "The use of domain knowledge in program understanding - Annals of Software Engineering," 9(2000)143-192.