ISSN 2303-4521
Periodicals of Engineering and Natural Sciences *Original Research*
Vol. 12, No. 2, July 2024, pp.429-443

Requirements management in software development companies in Colombia

Lengua-Cantero Claudia[1*], Lambraño Pérez Laudyt[2], García Medina María[3], Solorzano Peralta Namuel[4], Acosta Meza David[5]
[1,4] Faculty of Basic Sciences Engineering and Architecture, Corporación Universitaria del Caribe, Colombia
[5] Faculty of Humanities and Education, Corporación Universitaria del Caribe, Colombia

ABSTRACT

Software requirements management is vital in the system analysis stage, so the methods used in software development companies in Colombia were analyzed. The study aimed to explore the techniques used in the software development industry in Colombia for requirements management. It was carried out through a mixed approach of descriptive depth, interviewing five Chief Information Officers (CIOs) in Bogota, Medellin, and Sincelejo. Natural language processing techniques were used for the analysis, finding similarities among the interviewees in terms of software requirements and a positive perception towards requirements analysis; however, differences in requirements management were identified. Companies use hybrid methodologies to develop software projects, integrating principles of agile methodologies and methods such as Scrum and Kanban, while keeping traditional approaches as a base. This trend has an impact on requirements management since it allows greater flexibility and adaptability in the project development process.

Keywords: Software requirements management, software requirements, software development methodology, software engineering

Corresponding Author:

Lengua-Cantero Claudia
Faculty of Basic Sciences Engineering and Architecture,
Corporación Universitaria del Caribe, Colombia
E-mail: claudia.lengua@cecar.edu.co

1. Introduction

Requirements analysis is considered one of the most important phases in the systems life cycle or software life cycle since the effective response of analysts and developers to the challenges posed in organizations depends on it. This involves obtaining and analyzing data and information related to the system under study, to flow to a specification of functionalities to be implemented in the software. In parallel it is necessary to perform the activities required to ensure the success of a good requirements analysis, at this point appears the requirements management, as a necessary activity for planning, execution, and monitoring. This can lead to obtaining a correct, clear, unambiguous, and complete specification of software requirements, contributing significantly to customer satisfaction and the overall success of the software project.

1.1 Software requirement

The requirements of a system are the essential functions that the system must fulfill and the constraints on its operation, reflecting the user needs [1]. The process of identifying and confirming these services and constraints is called requirements engineering (RE). In addition, RE facilitates understanding customer needs, assessing

feasibility, negotiating a clear solution, specifying, validating, and managing it. It involves seven tasks: conception, inquiry, elaboration, negotiation, specification, validation and management [2].

In this regard, requirements are the specifications of what the software must do, and the descriptions of the behavior, properties, and restrictions that must be developed. They can be defined as the objectives to be satisfied in the software development stage; they also obey to satisfy the modeling of the real world. In essence, the number one activity, even before prioritizing the software development methodology to be used, is creating a list of requirements containing what the system being developed should do [3].

1.2 Requirements classification

Traditionally, requirements are classified into functional and non-functional categories; however, in practice, the distinction between them is not always clearly defined. Functional requirements are the specifications that the user expects the system to perform [1], Therefore, they should take into account the inputs and behavior of such a system. They are defined as both the user's requirements and the activities to be performed by the system. [3].

On the other hand, non-functional requirements, as stated in its name, are not related to the services provided by the system [1], such as the properties and specifications inherent to it, may also include the restrictions that are implemented for its proper functioning, such as safety, reliability, speed, robustness, and size, among others, as well as how the system should function concerning its operational environment [3].

1.3 Requirements engineering: Fundamentals and processes

Software development requires a systematic and disciplined approach, under software engineering principles. Adopting a systematic approach involves following a sequence of well-structured and organized phases, the application of which in various projects ensures consistent positive results. Conventional phases of software development include requirements analysis, software design, coding, testing, and deployment. Requirements analysis, in particular, stands out as a crucial and unavoidable phase, although all of them are important. Lack of user and customer involvement, as well as poor quality in requirements specification, are significant causes of failure in software development projects [1]. This situation has prompted a re-evaluation of requirements analysis, leading to the emergence of Requirements Engineering as a systematic and disciplined approach to address this fundamental aspect of software development.

1.4 Roles and responsibilities in requirements engineering

The activities or phases that comprise an RE process require the participation of people with clearly defined roles and responsibilities. Each role carries a specific set of responsibilities, behaviors, commitments, attitudes, and skills. For example, the requirements engineer is assigned to individuals with training and experience in information technology, software development, and related disciplines. This professional actively participates in all stages of the RE process and must possess skills such as empathy and active listening. On the other hand, customers, who have a high level of authority within organizations, make important decisions related to the financing of software development projects. However, users directly experience the operations of the organization and processes, enabling them to identify and express their tangible needs. Finally, the project manager assumes the responsibility of leading the project, plans, guides, coordinates, and facilitates to ensure the effective implementation of the RE process in all its facets.

1.5 Process Main activities of requirements engineering

The RE is characterized as a systematic and disciplined process, in its systematic nature, it establishes a flow of activities that includes elicitation, analysis, specification, validation, and verification, complemented by planning, monitoring, and control activities. As a disciplined process, the RE is open to adopting good practices and standards at each stage, to ensure that high-quality software requirements specifications are obtained.

1.5.1 Capture and elicitation of requirements

Elicitation is the first phase of the RE process, in practice requirements engineers must obtain information from the problem domain and learn firsthand the needs of customers and users.

1.5.2 Capture and elicitation techniques

Techniques are considered documented steps with a sequence, to ensure compliance with rules that enable a quality product, recommended for the process modeling phase, and include associated tools and notations [2]. Haz clic aquí para escribir texto. The following is a brief description of some of the techniques most commonly used in the requirements-gathering phase:

Observation: It is the monitoring and note-taking with the project stakeholders and customers, it can be passive or active, trying not to interfere in the performance of the processes.

Interviews: a technique based on open-ended questions that are useful for gathering and validating information.

Surveys or questionnaires: technique based on closed-ended questions useful for effectively gathering customer needs and expectations.

Workshops: This is a method aimed at gathering information to visually represent the process or processes through a scheme or flowchart.

Brainstorming: Allows the collection of ideas in a limited time, visual aids are usually used, and the process is supervised by an actor who establishes this role. It is essential to establish rules that allow to be punctual in the ideas based on established criteria, the intervention with criticism is not recommended since it would lead to a possible discussion.

User story: It is a technique used to gather software requirements formally, it uses the description of stories to represent the functionality and the expected result of a process or expected action [3].

1.6 Software requirements

Software requirements management involves the process of gathering, compiling, and analyzing the needs and requirements of a system or product under development. For a requirements management process to be successful, it must ensure that all deliverables meet the customer's expectations [4]. Principle of the form

The requirements of software allow for determining what is expected in a system, detailing the functions, and establishing the properties around the customer's expectations and the perspective of the development team. The purpose of requirements gathering is to capture the information needed to develop a statement that enables the understanding of what the software is expected to do, from the point of view of users in different roles, such as administrators or customers.

Requirements gathering and analysis is one of the most important processes in the success of a software development project. In this sense, the proper preparation of requirements specifications helps to reduce risks and costs associated with the development process [5].

1.7 Software Requirements Analysis

The software requirements analysis is considered as the set of procedures and techniques known as good practices, which allow to determine the necessary aspects to develop and build a software project, being an activity that is part of software engineering, aimed at identifying the operational characteristics, interfaces, restrictions and specifications that the software must perform to meet customer expectations[6]. The requirements specification provides the software developer and the customer with the necessary elements to assess compliance with processes, functions, results, and information management.

However, both the customer and the developers can be involved in carrying out a requirements analysis. This process includes clarifying ideas and managing data and software functions. Developers, acting as consultants,

must understand the customer's needs. Through this process, agreement can be reached on the system's functionalities, providing necessary guidelines to software designers.

While requirements analysis is beneficial for any project, it is most common in software engineering. Requirements analysis defines the expectations for the new software being built or modified. Requirements analysis in software engineering allows managers and project leaders to maintain clear direction, keep user needs at the forefront, and perform thorough documentation of the development process. Requirements analysis in software engineering is usually an iterative and continuous process throughout a project [7].

1.8 Validation of requirements

The task of validation of software requirements is the one that contributes to managing and generating a quality product, it consists of verifying that all the requirements have been written without ambiguity, omissions, or inconsistencies that may generate future errors, for this purpose it is worked under the perspectives of the standards defined in the project [8]. The software validation process addresses the stages of the software life cycle, in the sense that this process is used to determine whether a given product meets the needs and requirements established by the user for its development process. Software validation contributes to objective verification of the processes and products that are part of the software life cycle. These processes, based on standards such as ISO/IEC 15288:2008 and ISO/IEC 12207:2008, allow for conceiving the concept of quality and establishing parameters to measure whether the requirements are accurate, consistent, correct, and verifiable. Within the software validation process, there is a sub-process called software testing validating requirements, which are supported in the IEEE 1008 and ISO/IEC 29119 standards, arising from a need to ensure the quality of the product generated in the software life cycle [9].

Also, although there are several techniques for the validation of software requirements, such as requirements reviews, prototyping, test case generation, or automatic consistency analysis, the construction and use of conceptual models based on requirements are relevant for the validation [10].

Another model is a software verification and validation methodology based on the CMMI standard (Capability Maturity Model Integration), which allows to verify and validate software by describing the planning of strategies and activities to be applied in the software life cycle, and its purpose. The validation process tries to demonstrate that a product component or a completed product fits an intended use when placed in a production context [11]. The above models are based on standards and models that contribute to the management of business processes. Most of them focus on a specific part of the business and not on the whole, which is crucial considering the importance of applying a systemic approach to solve the problems of most industries. The following is a description of the methodology and results of this study, whose *objective* was to analyze the techniques used in the software development industry in Colombia in requirements management.

2. Research method

The study was framed in basic research with a mixed approach of descriptive depth, with a cross-sectional design. A semi-structured interview with 20 questions, including both open and closed-ended questions, was designed and validated. Five CIOs from software development companies located in Bogota, Medellin, and Sincelejo, Colombia, were selected intentionally to be interviewed.

3. Results and discussion

Natural language processing techniques were used to analyze the interviews:

3.1 Extraction of sentiment from interviews

In terms of emotions, the results of the interviews identify a higher level of confidence and anticipation around requirements management for software development, as shown in Figure 1 and Figure 2.

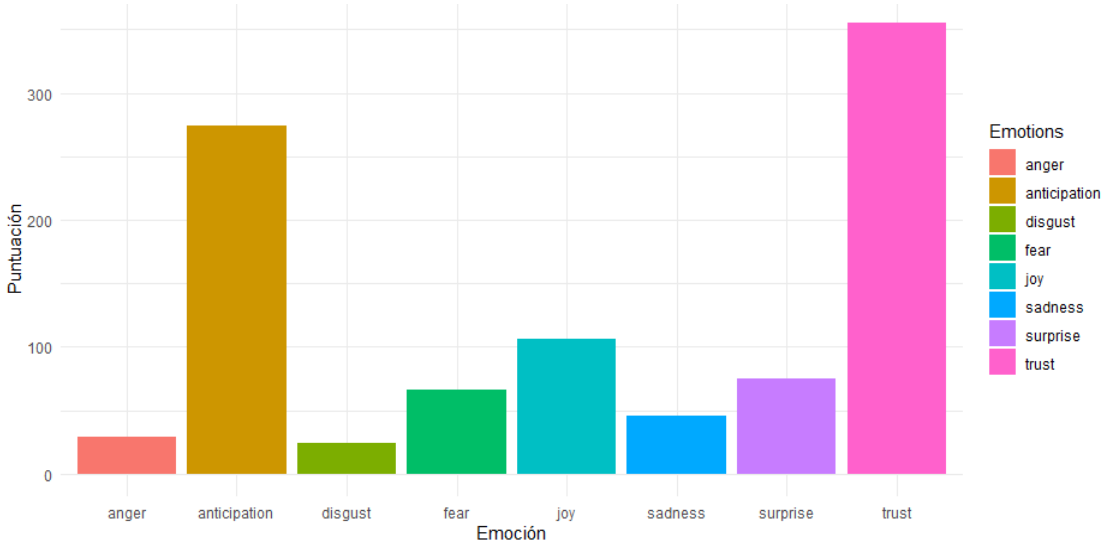

Figure 1. Emotion scoring

Taking it in a general way only negative and positive concerning the interviews, a greater number of positive feelings are identified around everything related to requirements management for software development.

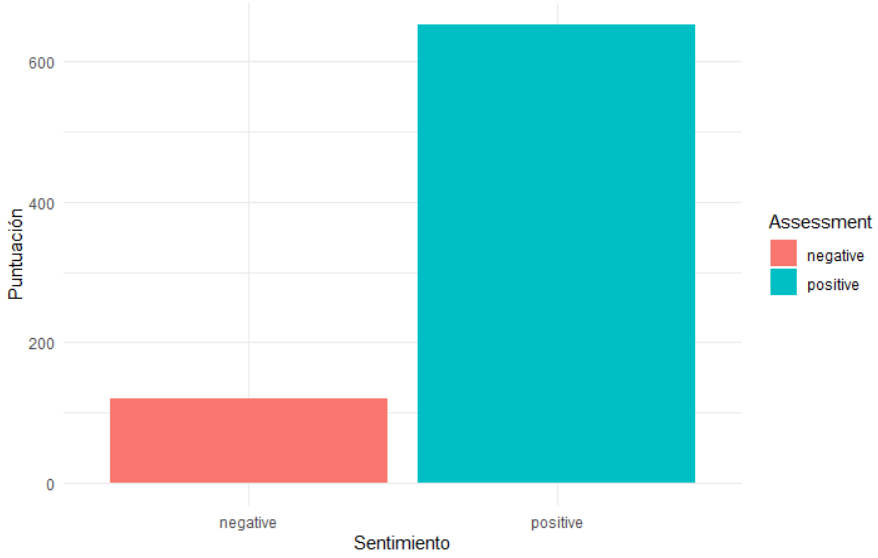

Figure 2. Positive and negative feelings

3.2 Analysis of similarities and correlations among interviewees

In this case, the cosine similarity analysis was used [12], and the results show moderately high similarity between them, interviewees 1 and 3 present a higher relationship in terms of software requirements, in terms related to data, base, management, and business, see figure 3.

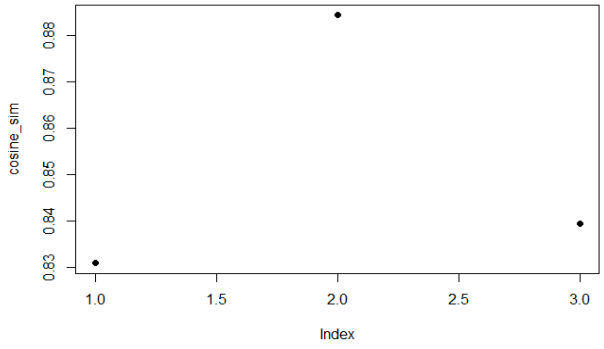

Figure 3. Similarities in interview results

3.3 Lexical dispersion and controlled vocabulary

Using Relative Tokens Index (RTI), the RTI of a term was calculated is as follows:

$$RTI = \frac{f_t / \sum_i f_i}{F_t / \sum_i F_i}$$

Where:

f_t: it is the frequency of the term in the corpus of interest.

$\sum_i f_i$: it is the sum of the frequencies of all terms in the corpus of interest.

F_t: it is the frequency of the term in the reference corpus (generally a larger corpus or a broad collection of texts).

$\sum_i F_i$: is the sum of the frequencies of all terms in the reference corpus.

3.4 Innovation and experience

In the case of the word 'innovation' represented by the color red, it can be observed that it was only used by interviewee 2 for almost the entire interview, with the other interviewees not using it. As for the word 'experience' in blue, it was used by all three interviewees both at the beginning and the end of the interview, as depicted in Figure 4.

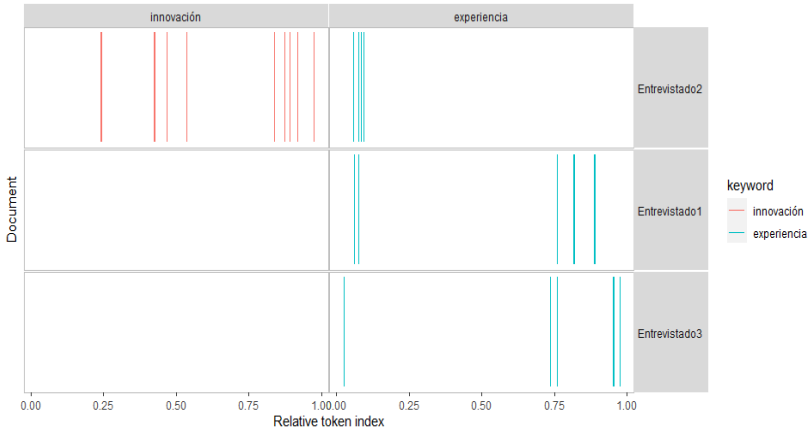

Figure 4. Lexical dispersion: innovation and experience

3.5 System and follow-up

In the case of the word follow-up represented by the color red, it can be observed that it was used by all three interviewees, where it is highlighted that interviewee 2 used it very little. As for the word blue follow-up, it was used by all three interviewees at the beginning and end of the interview, as shown in Figure 5.

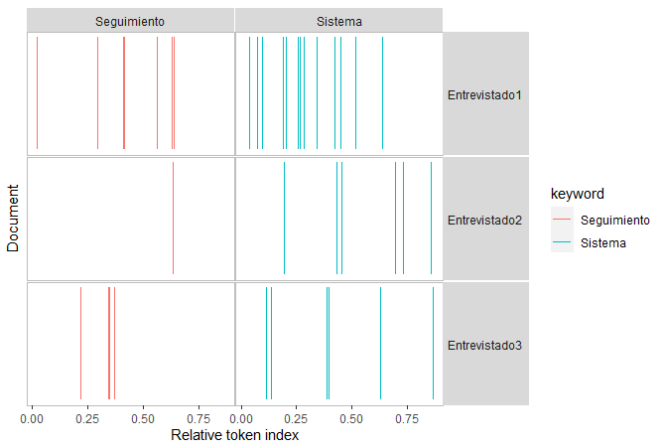

Figure 5. Lexical dispersion: system and follow-up

3.6 Cascade and agile

In the case of the word Cascade represented by the color red, it can be observed that it was used by interviewees 1 and 3, where it is highlighted that interviewee 3 used it very little. On the other hand, the word Agile, represented by blue, was used by interviewees 1 and 3 at the beginning and the end of the interview, see Figure 6.

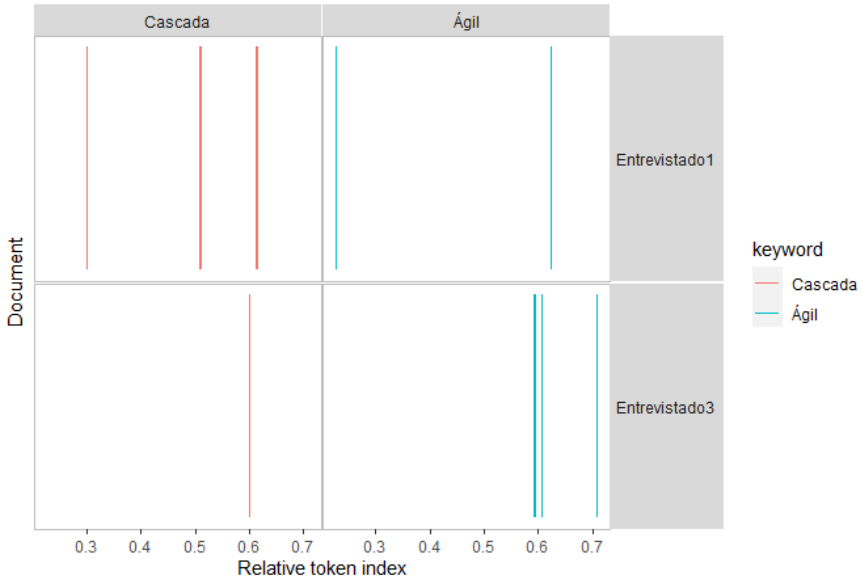

Figure 6. Lexical dispersion: cascading and agile

3.7 Security and management

In the case of the word security, represented by the color red, it is observed that it was used by interviewees 1 and 3, where it is highlighted that interviewee 3 used it very little. As for the word management represented by blue, it was used by interviewees 1 and 3, where interviewee 3 used it at the beginning and end of the interview and interviewee 1 only minimally at the end, as shown in Figure 7.

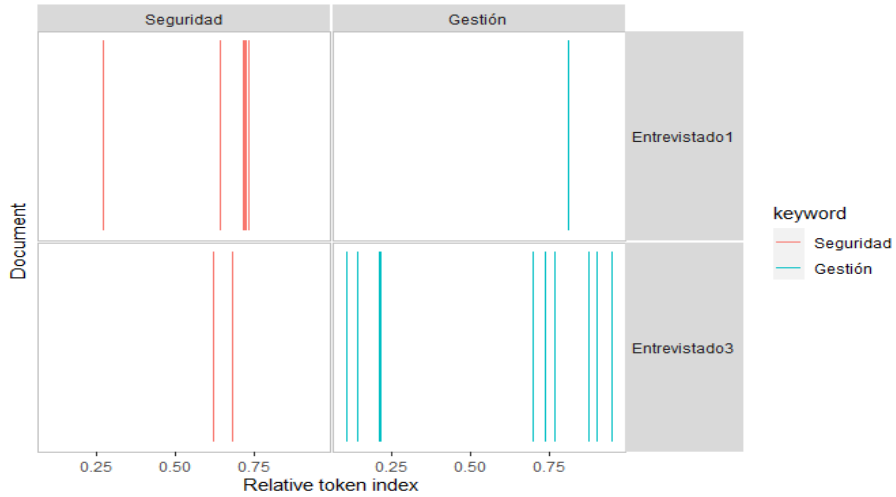

Figure 7. Lexical dispersion: advanced security

3.8 Software methodology and artificial intelligence

In the case of the word security, represented by the color red, it is observed that it was used by interviewees 1 and 3, where it is highlighted that interviewee 3 used it very little. As for the word management represented by blue, it was used by interviewees 1 and 3, where interviewee 3 used it at the beginning and at the end of the interview and interviewee 1 only minimally at the end, as shown in Figure 8.

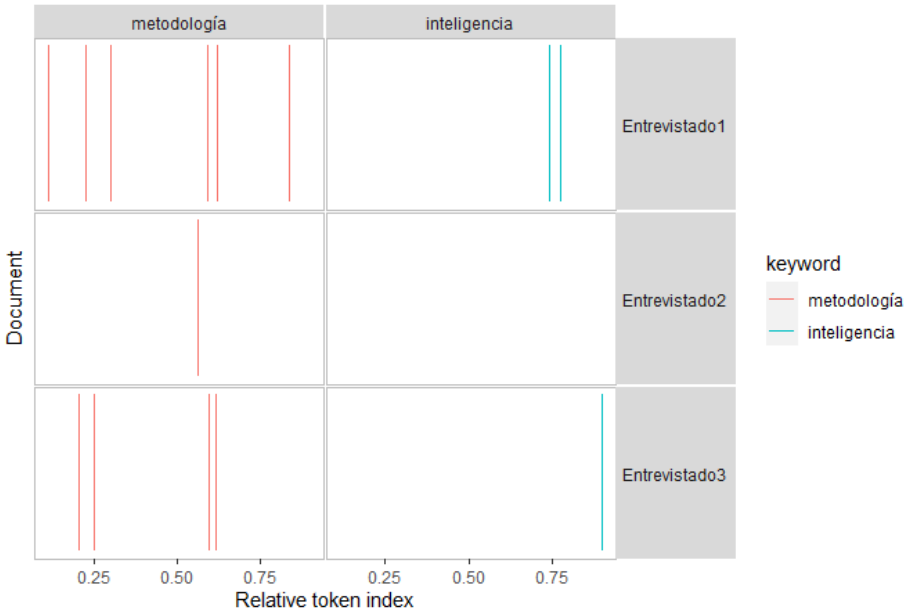

Figure 8. Lexical dispersion: software methodology and artificial intelligence

3.9 Software and Requirements

These two words are used by the two interviewees heavily throughout the interviews, almost that, jointly, see Figure 9.

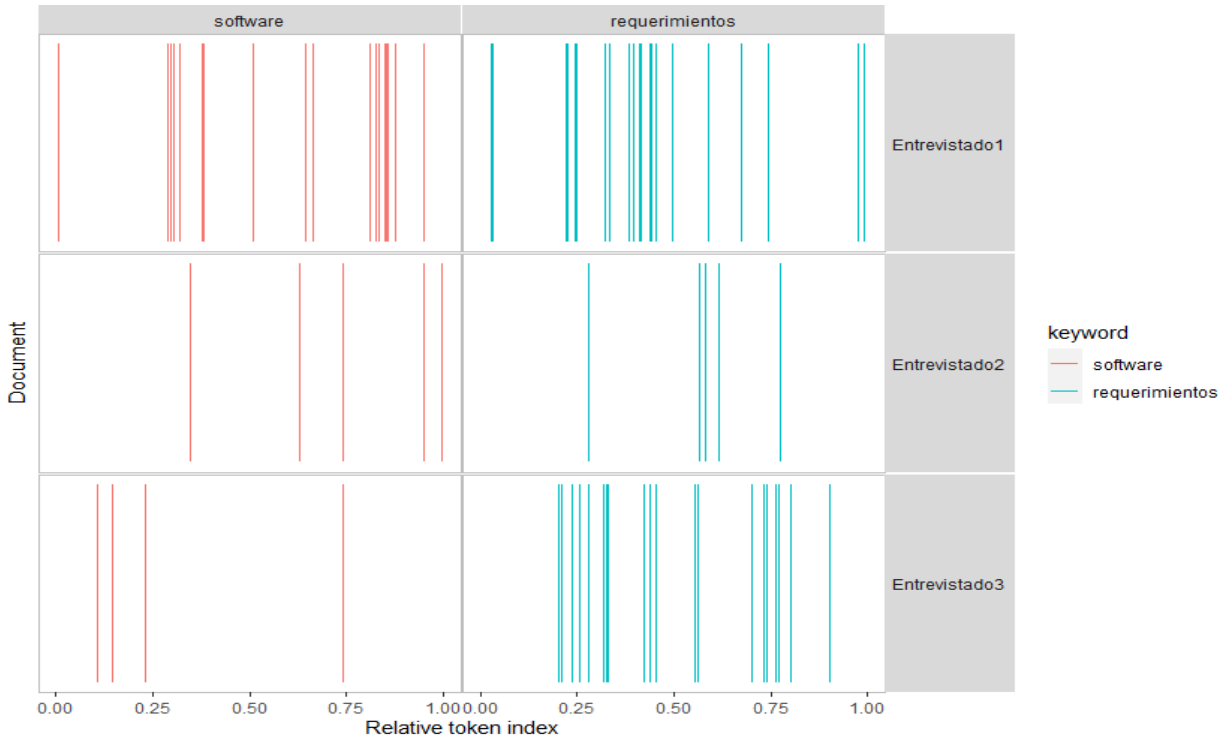

Figure 9. Lexical dispersion: software and requirements

3.10 Use of frequent words

In the word cloud, it can be seen that the most frequent words used by interviewee 1 regarding software requirements are security, need, data, and customers, among others. While for interviewee 2 they are development, innovation, mayor's office, secretary, and health, among others. Finally, for interviewee 3, processes, management, and business, among others, are shown in Figures 10, 11, and 12.

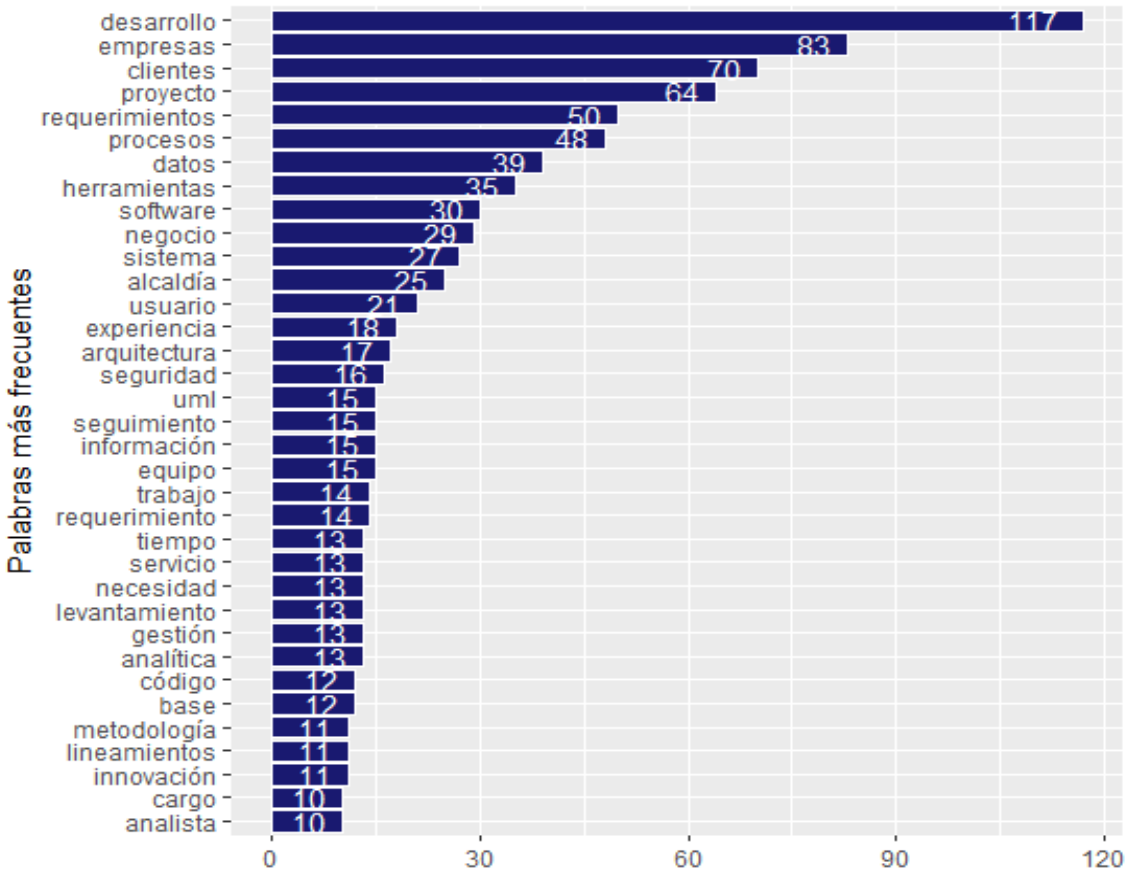

Figure 10. Frequent words used by interviewees

Figure 11. General word cloud, with words used

Figure 12. Word cloud separated by interview, with the words used.

3.11 Comparison of keywords identified by interviews

3.11.1 Interviewee 1

The most important keywords in the corpus of Interviewee 1 in terms of their relative importance, not including reference information. Distinctive words in the context of software requirements for Interviewee 1 are those with higher chi-square values compared to other sections of the text. These aspects include data, database, customers, security, UML, and user stories, as depicted in Figure 11.

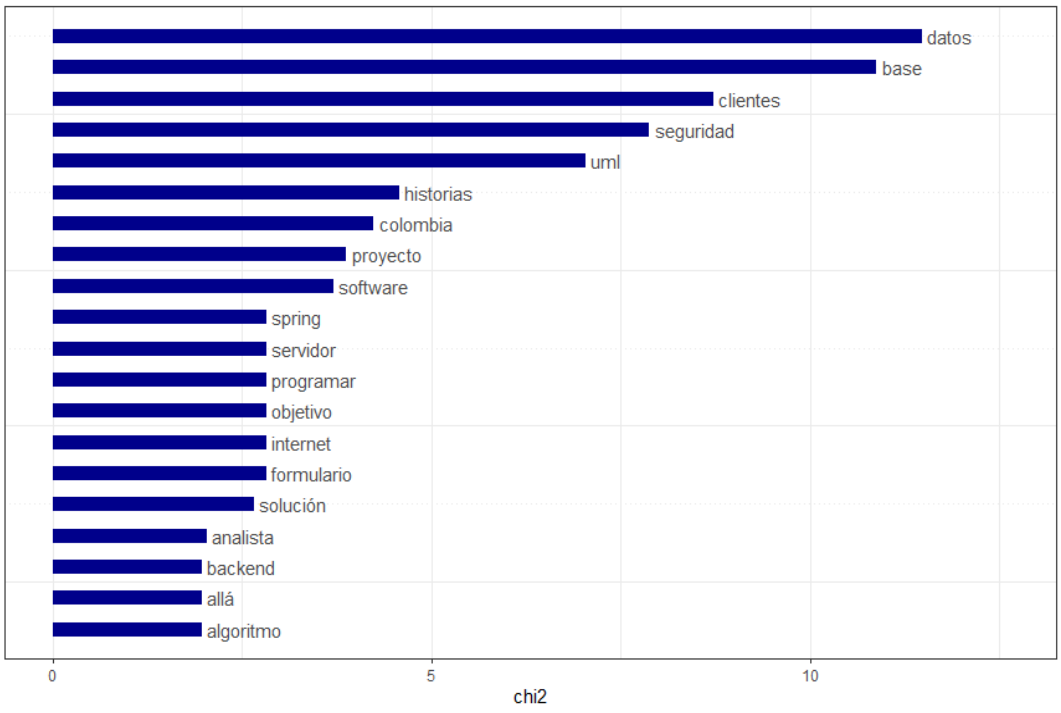

Figure 13. Comparison of most used words, according to their importance of the interviewee

3.11.2 Interviewee 2

The distinctive words in the context of software requirements for the case of Interviewee 2 compared to segments of the same text are those with a higher chi-square value and were aspects related to innovation, development, health, digital, service, and public, see Figure 14.

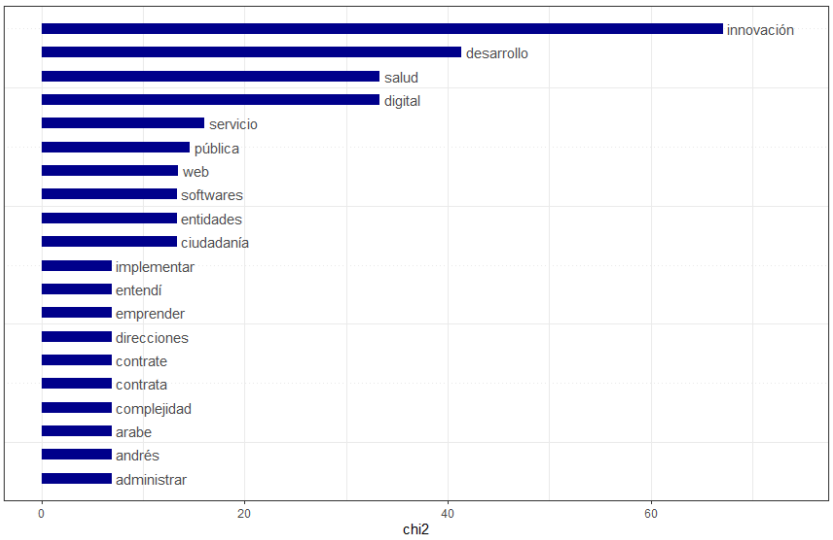

Figure 14. Comparison of most used words, according to their importance to the interviewee 2.

3.11.3 Interviewee 3

The distinctive words in the context of software requirements for the case of Interviewee 3 compared to segments of the same text are those with a higher chi-square value and were aspects related to management, business, sales, areas, internal, and time, see Figure 15.

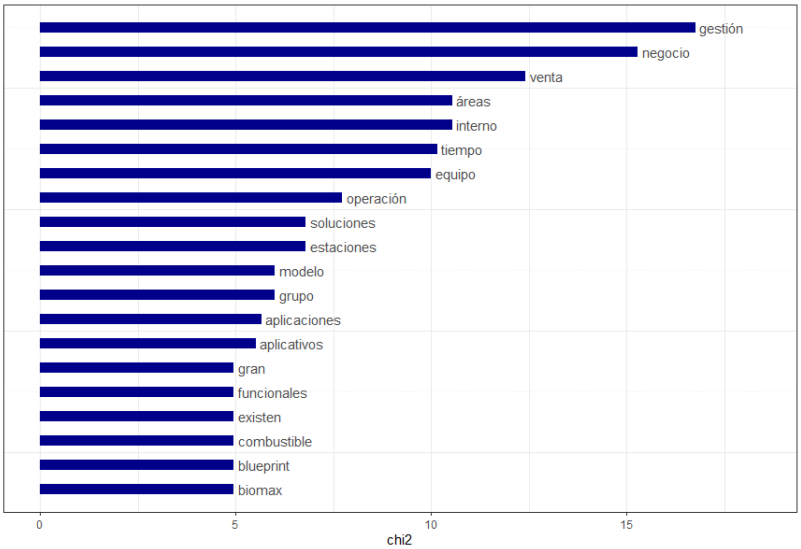

Figure 15. Comparison of the most used words, according to their importance for the interviewee 3.

3.12 Differences between interviews and grouping among respondents

3.12.1 Hierarchical Grouping

The distance between "Interviewee 1" and "Interviewee 2" is quite high, suggesting that these two interviews are different from each other in terms of characteristics, responses, or content. The distance between "Interviewee2" and "Interviewee3" is lower than the above distance, suggesting moderate

similarity. This indicates that these two interviews share some similarities in their characteristics or content. The distance between "Interviewee1" and "Interviewee 3" is intermediate compared to the other two distances. This suggests that these two interviews are moderately different from each other, but they also share some similarities, see Figure 16.

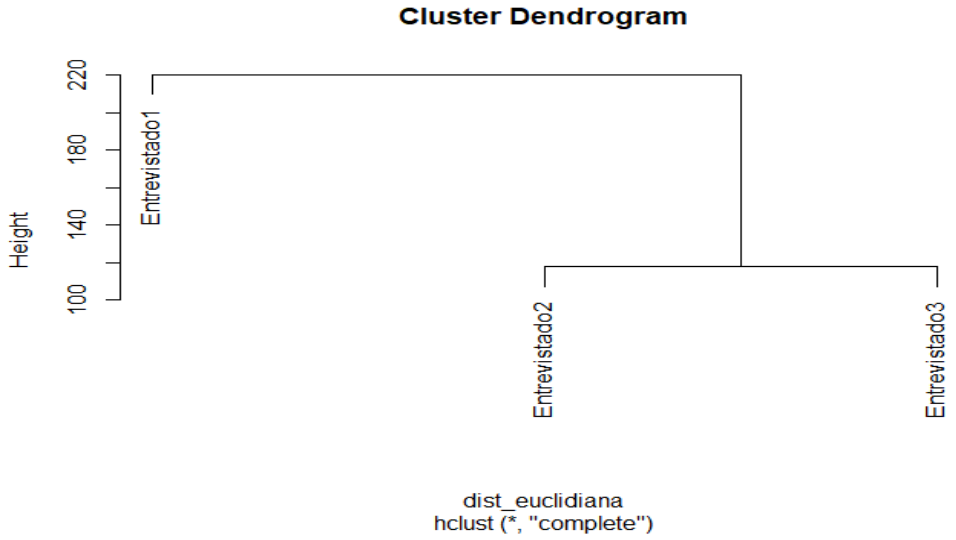

Figure 16. Euclidean distance between respondents.

3.13 Word clustering network to assess the interaction and co-occurrence between different words in interviews

3.13.1 Interviewee 1

For the case of interviewee 1, the aspects related to software are data, development, companies, customers, waterfall, security, standards, monitoring, and needs, see Figure 17.

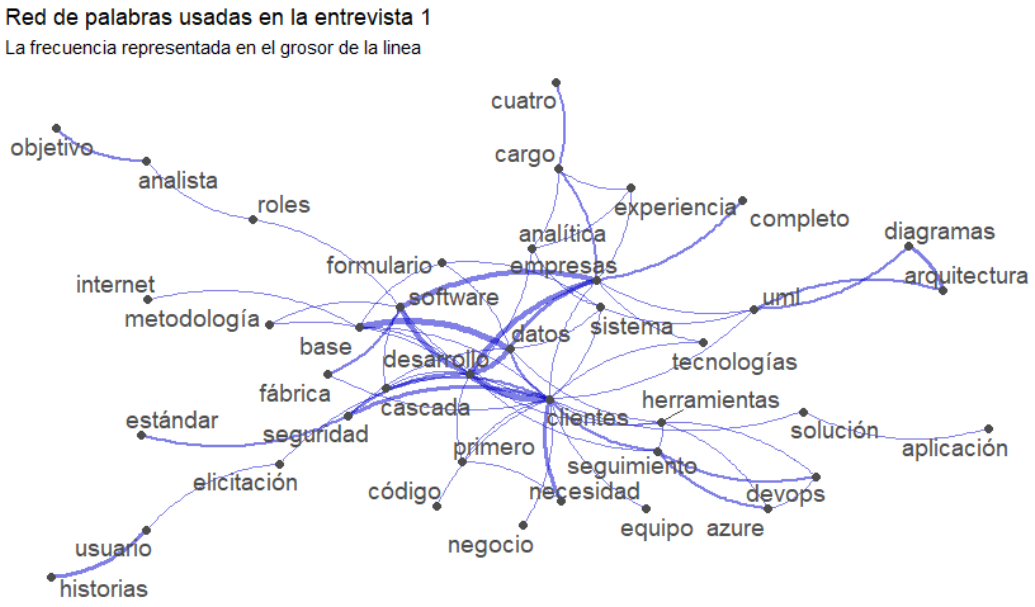

Figure 17. Network of words used by respondent 1

3.13.2 Interviewee 2

In the case of interviewee 2, there is evidence of a pattern of words related to the term development, which is related to innovation, politics, health, applications, software, companies, new services, and experiences (see Figure 18).

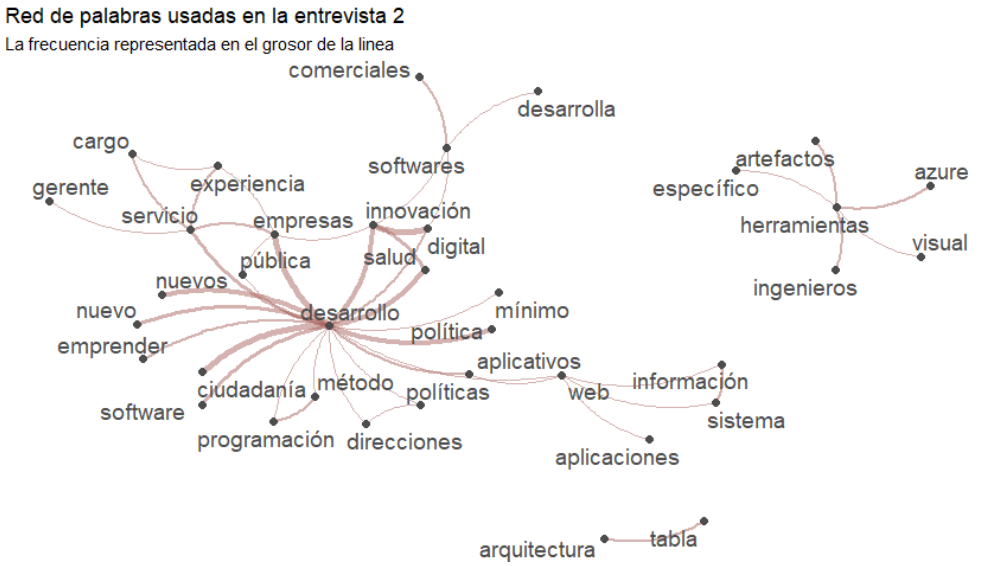

Figure18. Network of words used by the interviewee 2.

3.13.3 Interviewee 3

In the case of interviewee 3, it is evident that in terms of software development, he relates to words such as solutions and improvements, traditional methodologies, management of tools and experiences in the work and time, about customer areas, information, and information gathering (see Figure 19).

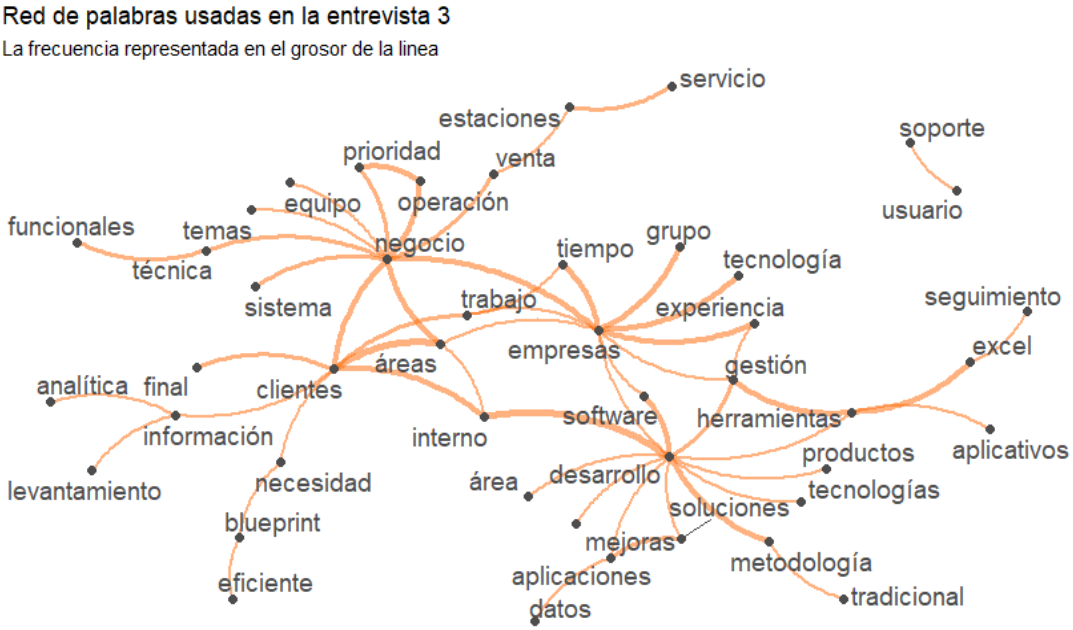

Figure 19. Network of words used by the interviewee 3.

Currently, one of the most popular agile methodologies for software development is Scrum. It is an interactive and incremental framework developed in work cycles called Sprints. Parallel to the use of Scrum, Kanban provides a way to organize and manage the work through visual boards that show the tasks in different stages of the process, likewise, Design Thinking is a working method used to generate disruptive ideas, based on customer needs in the construction of the final prototype [2].

The use of a software development methodology such as Scrum, the use of these working methods (Kanban or Design Thinking), a good conceptual and practical mastery that guarantees a correct implementation of the

methodology in the specific context of the project, and a good disposition of the development team influences the way requirements are managed [13].

In addition, several studies describe the design of automated tools and the use of Artificial Intelligence as a mechanism for classifying software requirements [14], [15], [16] [17]. This suggests a trend towards the implementation and use of information systems that contribute to improving the software requirements management process, marking an important advance in the industry, especially in terms of improving the traceability of software requirements established by systems analysts at the analysis stage.

4. Conclusions

According to the results of the study, for the case of software development companies in Colombia, traditional software development methodologies are the basis of current trends, hence most of the companies surveyed use standardized hybrid methodologies according to their needs. This is how they take the agile methodologies and methods, starting from the premises proposed in Scrum, Kanban, and the traditional ones: Waterfall, Incremental, and Prototypes. On the other hand, the UML standard is taken as a reference in the analysis stage, using use case diagrams to understand the requirements expressed by customers. User stories are used to specify and track project requirements.

Declaration of competing interest

The authors declare that they have no known financial or non-financial competing interests in any material discussed in this paper.

Funding information

This study was funded by Corporación Universitaria del Caribe–CECAR (grant number: ACTA No 31 30/12/2022).

Author contribution

The contribution to the paper is as follows: Lengua-Cantero Claudia: conception, design, analysis and interpretation of results. Lambraño Pérez Laudiyt: data collection and interpretation of results. Garcia Medina Maria: instrument design and data collection. Solorzano Peralta Namuel: interpretation of information. Acosta Meza David: instrument design.

References

[1] I. Sommerville, Software Engineering, 10th ed. New York, 2016.

[2] R. Pressman and B. Maxim, Ingeniería del software un enfoque páctico, 9th ed., vol. 1. México, D. F, 2021.

[3] L. Lis Rinaudo and G. Pantaleo, Ingeniería De Software, vol. 1. Paraguay, 2016.

[4] G. Pantaleo and Rinaudo Ludmila, Ingeniería de Sosfware. Buenos Aires, 2015. Accessed: Apr. 25, 2024. [Online]. Available: https://books.google.com.co/books?id=a8j2DQAAQBAJ&lpg=PP1&pg=PT3#v=onepage&q&f=false

[5] C. López Valerio and J. C. Mejías Cruz, "Guía para la Ingeniería de Requerimientos bajo un enfoque ágil integrando técnicas de usabilidad," Iberian Journal of Information Systems and Technologies, vol. 32, pp. 546–558, 2020, [Online]. Available: http://www.risti.xyz/issues/ristie32.pdf

[6] L. Rondon, "Quality in gathering requirements in software projects," Universidad Militar Nueva Granada, Bogotá, 2019. Accessed: Jun. 06, 2023. [Online]. Available: https://repository.unimilitar.edu.co/handle/10654/31989

[7] M. del C. Gómez Fuentes, Análisis de requerimientos, vol. 1. Ciudad de México, 2022. Accessed: May 07, 2023. [Online]. Available: https://www.cua.uam.mx/pdfs/conoce/libroselec/Notas_Analisis_Requerimiento.pdf
[8] M. Norris and P. Rigby, Ingenieria de software explicada, 3rd ed. 1994.
[9] A. P. Gómez Ramírez, "Validación y verificación de requisitos de software en proyectos basados en desarrollo continúo usando técnicas de PLN," Sep. 2022, pp. 1–7. doi: 10.26507/paper.2657.
[10] U. Kelechi, "How to Carry Out a Requirements Analysis." Accessed: Jun. 06, 2023. [Online]. Available: https://www.wrike.com/blog/how-carry-out-requirements-analysis/#What-is-a-requirements-analysis
[11] L. Pauta Ayabaca and S. Moscoso Bernal, "Verificación y validación de software," Killkana Técnica, vol. 1, no. 3, p. 25, Feb. 2018, doi: 10.26871/killkana_tecnica. v1i3.112.
[12] M. Marciszack, R. Pérez, and C. Castro, "Validación de Requerimientos a través de Modelos Conceptuales-Modelos y Transformaciones," 2013. Accessed: Jul. 28, 2022. [Online]. Available: https://core.ac.uk/download/pdf/296353176.pdf
[13] O. Puello, "Modelo de Verificación y Validación Basado en CMMI," Investigacion e Innovación en Ingenierias, vol. 1, no. 1, Jan. 2013, doi: 10.17081/invinno.1.1.2068.
[14] L. A. Calvo-Valverde and J. A. Mena-Arias, "Evaluación de distintas técnicas de representación de texto y medidas de distancia de texto usando KNN para clasificación de documentos," Revista Tecnología en Marcha, Feb. 2020, doi: 10.18845/tm.v33i1.5022.
[15] K. E. Wiegers and J. Beatty, Software Requirements, 3rd ed., vol. 3. 2013.
[16] B. Li and X. Nong, "Automatically classifying non-functional requirements using deep neural network," Pattern Recognit, vol. 132, p. 108948, Dec. 2022, doi: 10.1016/j.patcog.2022.108948.
[17] M. H. Marghny, E. A. Zanaty, W. H. Dukhan, and O. Reyad, "A hybrid multi-objective optimization algorithm for software requirement problem," Alexandria Engineering Journal, vol. 61, no. 9, pp. 6991–7005, Sep. 2022, doi: 10.1016/J.AEJ.2021.12.043.
[18] R. Sonbol, G. Rebdawi, and N. Ghneim, "Learning software requirements syntax: An unsupervised approach to recognize templates," Knowl Based Syst, vol. 248, p. 108933, Jul. 2022, doi: 10.1016/J.KNOSYS.2022.108933.
[19] H. Liu, M. Shen, J. Zhu, N. Niu, G. Li, and L. Zhang, "Deep Learning Based Program Generation From Requirements Text: Are We There Yet?," IEEE Transactions on Software Engineering, vol. 48, no. 4, pp. 1268–1289, Apr. 2022, doi: 10.1109/TSE.2020.3018481.