

# Metodologías, Métodos, Herramientas, Problemas de la Especificación de Requisitos: Revisión Sistemática de Literatura

## Methodologies, Methods, Tools, Requirement Specification Problems Systematic Literature Review

Yannela Castro, Jhon Henao, Alex Nole, Cesar Santin, Anthonny Torres  
{yannela.castro,jhon.henao, alex.nole, cesar.santin,anthonny.torres}@unl.edu.ec

**Resumen**—Este artículo presenta un estudio sobre el mapeo y revisión sistemática de literatura con el fin de identificar, analizar y clasificar los artículos publicados sobre el uso de metodologías, métodos, herramientas y problemas de las especificaciones de requisitos de software. Para realizar la revisión se utilizó la técnica propuesta por Barbara Kitchenham, la misma que permitió seleccionar 40 artículos que muestran información sobre el presente tema. Así mismo se utilizó la herramienta Parsifal para optimizar el proceso de indagación y exclusión de documentos que no son de ayuda con la finalidad de la temática propuesta. Los artículos se seleccionaron desde el año 2015 en adelante; las principales metodologías, métodos, herramientas y problemas identificadas en la revisión son: Metodologías Arsenal, análisis semánticos, ISO IEEE 830, DUSM. Los métodos analizados, el uso de métodos ágiles, análisis semántico, método de inspección gramatical. Las herramientas más utilizadas TEMOS, BOILERPLATE, PROMINAR. El problema más común descrito es la ambigüedad en la especificación de requisitos de software entre otras.

**Palabras Claves**—Método, Metodología, Requerimientos, Ingeniería, Especificación de Requisitos, Software, Estándar

**Abstract**—This article presents a study on the mapping and systematic review of literature in order to identify, analyze and classify published articles on the use of methodologies, methods, tools and problems of software requirements specifications. The technique proposed by Barbara Kitchenham was used to carry out the review, which allowed selecting 40 articles that show information about the present topic. Likewise, the Parsifal tool was used to optimize the process of inquiry and exclusion of documents that are not helpful for the purpose of the proposed topic. The articles were selected from 2015 onwards; the main methodologies, methods, tools and problems identified in the review are Arsenal Methodologies, semantic analysis, ISO IEEE 830, DUSM. The methods analyzed, the use of agile methods, semantic analysis, grammatical inspection method. The most used tools are TEMOS, BOILERPLATE, PROMINAR. The most common problem described is the ambiguity in the specification of software requirements among others. **Keywords:** Method, Methodology, Requirements Engineering, Requirements Specification, Software, Standard

**Keywords**—Method, Methodology, Requirements Engineering,

Requirements Specification, Software, Standard

### I. INTRODUCCIÓN

La condición previa esencial para la implementación exitosa de los productos de software es el desarrollo preliminar de la especificación de requisitos de software (SRS) de alta calidad. La última tarea se resuelve durante el proceso de ingeniería de requisitos (RE), en el que las ideas inexactas e incompletas sobre qué servicios debe proporcionar el producto de software para resolver los problemas de los usuarios potenciales se convierten en un SRS formal [1].

El fracaso de un proyecto de software (SW), se da por que las empresas le dan solo un mínimo esfuerzo a los requisitos y se enfocan en otras etapas como lo son el desarrollo y mantenimiento neto del SW, si bien es cierto es importante una buena especificación de requisitos porque nos otorga una descripción completa sobre los requisitos funcionales y no funcionales de un sistema de SW a desarrollar. SRS ayuda al analista a comprender las necesidades del cliente y es el documento base para varias actividades del desarrollo de SW. La calidad de SRS es el factor que más contribuye al éxito o fracaso del proyecto a desarrollar, una medida de calidad es que SRS debe ser inequívoco [2]. El documento SRS no es ambiguo si y sólo si los requisitos contenidos en este documento tienen una sola interpretación, sin embargo, la mayoría de los SRS de Lenguaje Natural (NL) contienen requisitos ambiguos que pueden afectar negativamente en todo el proceso de desarrollo de software. En el presente documento se dará a conocer el proceso de la revisión sistemática de literatura (RSL), para la obtención de información relacionada con la especificación de requisitos de SW, presenta la siguiente estructura II. Metodología de investigación, III Resultados, IV. Discusión, V. Conclusiones.

### II. METODOLOGÍA DE INVESTIGACIÓN

En [3] menciona que toda investigación debe estar basada en evidencias que permitan identificar cómo ha sido abordado

el objeto de investigación por otros autores. Sin embargo, es necesario buscar y agregar evidencias usando estudios secundarios como son las RSL y estudios de mapeo sistemático [4]. Para esta investigación se utilizó el protocolo definido por [5] y se estructuró el documento en referencia a [6] [7].

#### A. Preguntas de investigación

A partir de la temática central denominada “Especificación de Requisitos de Software” se planteó ocho preguntas de investigación, clasificadas en preguntas para la revisión sistemática (RQ) y de mapeo sistemático (MQ):

- RQ1: ¿Qué metodología se utiliza para la especificación de requisitos?
- RQ2: ¿Qué estándares existen para la especificación de requisitos?
- RQ3: ¿Qué software utilizan los investigadores para la especificación de requisitos?
- RQ4: ¿Qué problemas se presentan en el proceso de la especificación de requisitos?
- MQ1: ¿Cuántos estudios se publicaron a lo largo de los años en el área de la especificación de requisitos?
- MQ2: ¿Quiénes son los actores más relevantes y activos en esta área de la especificación de requisitos?
- MQ3: ¿Cuáles son las revistas y congresos que se han publicado en el área de la especificación de requisitos?
- MQ4: ¿Cuántos documentos han sido seleccionados y cuantos se han aceptado para la revisión?

#### B. Proceso de Búsqueda.

Se determinaron un conjunto de términos basados en las preguntas de investigación para construir la cadena de búsqueda. Para esto se utilizó la metodología PICOC proouesto por [6] para definir el ámbito del RSL:

- **Population(P):** Requirements Engineering
- **Intervention(I):** Requirements Specification
- **Comparison(C):** No Key word
- **Outcome(O):** Methodology, Standard, Method, Software
- **Context(C):** Requirements Specification

#### C. Definición de los criterios de inclusión (IC) y exclusión (EC)

Se definieron 6 Criterios de Inclusión (IC):

- IC1: Los documentos deben contener información en relación a estándares de SRS. AND
- IC2: Los documentos deben contener información en relación a metodologías de especificación de requisitos. AND
- IC3: Los documentos deben estar escritos en inglés
- IC4: Los documentos deben ser publicados desde el 2015. AND
- IC5: Los documentos detallan el uso de software para la SRS. AND
- IC6: Los documentos detallan problemas sobre la SRS. OR

A demás, se definieron 6 criterios de exclusión (EC):

- EC1: Los documentos que no tengan información en relación a metodologías de especificación de requisitos. OR
- EC2: Los documentos que no contengan información en relación de estándares en la SRS. OR
- EC3: Los documentos que estén escritos en español. OR
- EC4: Los documentos que sean publicados antes del 2015. OR
- EC5: Los documentos que no estén relacionados en el área de la especificación de requisitos dentro de la ingeniería de software. OROR

#### D. Cadenas de Búsqueda

Se definieron palabras claves a través del método PICOC [8] que permitieron construir la cadena de búsqueda con una sinonimia de cada palabra en conjunto con operadores lógicos, “AND” para los conceptos complementarios y “OR” para los documentos similares, para lo cual se utilizó la misma cadena de búsqueda en las bases de dato escogidas pero con diferente estructura de la misma, dependiendo de la base de datos.

##### IEEE Library:

- (“Requirements Specification”) AND (“Method” OR “Methodology” OR “Software” OR “Standard”)

##### Scopus:

- TITLE-ABS-KEY ( ( “Requirements Specification” ) AND ( “Method” OR “Methodology” OR “Software” OR “Standard” ) ) AND ( LIMIT-TO ( PUBYEAR , 2020 ) OR LIMIT-TO ( PUBYEAR , 2019 ) OR LIMIT-TO ( PUBYEAR , 2018 ) OR LIMIT-TO ( PUBYEAR , 2017 ) OR LIMIT-TO ( PUBYEAR , 2016 ) OR LIMIT-TO ( PUBYEAR , 2015 ) ) AND ( LIMIT-TO ( DOCTYPE , “cp” ) OR LIMIT-TO ( DOCTYPE , “ar” ) ) AND ( LIMIT-TO ( SUBJAREA , “COMP” ) OR LIMIT-TO ( SUBJAREA , “ENGI” ) ) AND ( LIMIT-TO ( LANGUAGE , “English” ) )

##### ScienceDirect:

- (“Requirements Specification”) AND (“Method” OR “Methodology” OR “Software” OR “Standard”)

#### E. Evaluación de Calidad

Definida la cadena de búsqueda, es necesario la evaluación de calidad de los documentos seleccionados en cada base de datos. Para cada artículo, son evaluadas con los siguientes criterios:

- QA1: ¿El autor detalla o al menos utiliza un software para la especificación de requisitos?
- QA2: ¿El autor identifica o detalla algún problema dentro de la especificación de requisitos?
- QA3: ¿Existe en el artículo citas sobre estándares para la especificación de requisitos?
- QA4: ¿El autor describe o detalla al menos una metodología para la especificación de requisitos?

### III. RESULTADOS

Definido el protocolo en la fase anterior, los resultados obtenidos de la RSL se describen en los siguientes pasos:

- 1) Se ejecutó las cadenas de búsqueda en cada base de datos y se obtuvieron 1008 artículos que representan la muestra del total.
- 2) Al subir los 1008 documentos a la herramienta parsifal, esta detectó que existen 175 duplicados que representan el 17,36% del total, los mismos que fueron descartados, quedando 833 artículos por revisar.
- 3) Los 833 artículos fueron revisados y analizados en su título y resumen en relación con la temática central, tomando en consideración los criterios de inclusión y exclusión. Del total se rechazaron 752 artículos (74,60%), que son irrelevantes para el objetivo de estudio. Además, hacen referencia a ejemplos de SRS basadas en aplicaciones con inteligencia artificial, así como en proyectos industriales, por lo tanto, su argumentación referente a la SRS es débil y no dan contestación a las preguntas de investigación planteadas. En la tabla I se detalla el proceso de selección de los estudios, de los cuales quedaron 81 artículos para la evaluación por los criterios de calidad.

Base de Datos	Artículos				
	Encontrados	Duplicados	Revisados	Eliminados	Selección
IEEE	198	80	118	101	17
Science Direct	68	23	45	39	6
Scopus	742	72	670	612	58
Total	1008	175	833	752	81

Table I: ARTÍCULOS REVISADOS.

- 4) Por último, cada una de las preguntas planteadas para el desarrollo del control de calidad tiene un puntaje de 1 si esta es calificada con "Sí", 0,5 si la respuesta es "Parcial" y 0 si la respuesta fue evaluada con "No". Cada artículo tuvo una calificación de 0 a 4 puntos. Si tiene un puntaje de igual o superior a 2, será seleccionado para extraer su información. Los que cumplieron con los criterios de calidad son 40 que representan (3,96%).

#### A. Informe del mapeo sistemático.

**MQ1: "¿Cuántos estudios se publicaron a lo largo de los años en el área de la especificación de requisitos?"** En el lapso de tiempo entre 2015-2020 se han publicado 40 artículos. En la Fig.1 el número de artículos por año, en el año 2018 tiene un porcentaje del 32,5% que corresponde a la mayoría de documentos extraídos.

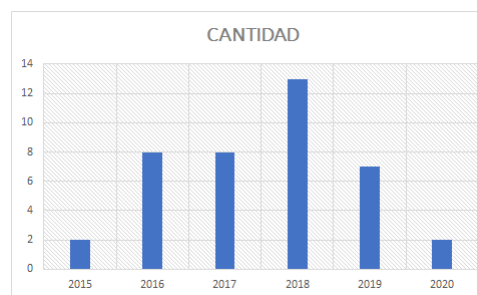


Figure 1: Publicaciones por año.

**MQ2: "¿Cuántos documentos han sido seleccionados y cuantos se han aceptado para la revisión?"** En la Fig.2 se muestra la cantidad de documentos seleccionados en cada base de datos, marcada de color azul en donde para la base de datos Scopus se seleccionaron 748 artículos, en IEEE 198 artículos y en ScienceDirect 68 artículos, lo cual da un total de 1008 documentos. De los documentos seleccionados se aceptaron 58 artículos en Scopus, 17 en IEEE y 6 en ScienceDirect, lo cual da un total de 81 documentos.

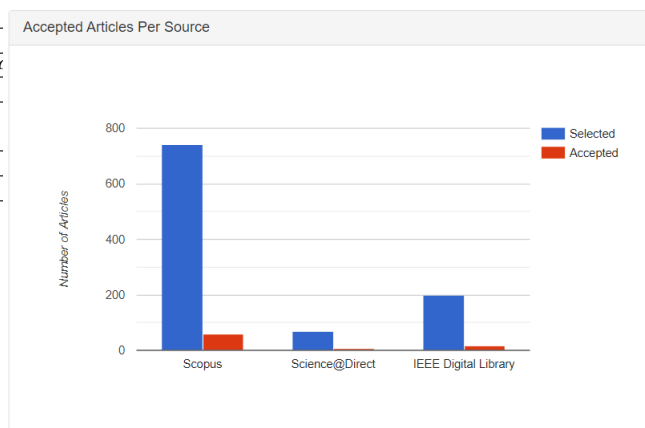


Figure 2: Artículos aceptados por criterio

**MQ3: "¿Quiénes son los actores más relevantes y activos en esta área de la especificación de requisitos de SW?"** En el área de la especificación de requisitos los actores más relevantes que se encontró en el RSL y que han incursionado esta área son: Šenkýř, D. and Kroha, P. los cuales han aportado con 3 documentos y Hovorushchenko, T. and Pomorova, O con la publicación de 2 documentos.

**MQ4: "¿Cuáles son las artículos y conferencias que se han publicado en el área de la especificación de requisitos?"** En la Fig.3 se detalla que se han publicado 26 conferencias y 14 artículos acerca de la SRS.

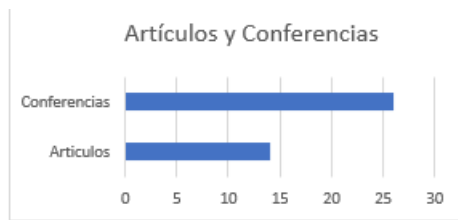


Figure 3: Lugar de publicación de artículos y conferencias

En la tabla II se presentan los 40 artículos seleccionados donde se detalla la información, el título, el año de publicación y sus fuentes de publicación para identificar posibles escenarios de publicación.

Table II: ARTÍCULOS EVALUADOS

Título	Año	Artículos/ Conferencias
Requirements Dependency Graph Modeling on Software Requirements Specification Using Text Analysis [9]	2019	1st International Conference on Cybernetics and Intelligent System, ICORIS 2019
Methodology of evaluating the sufficiency of information on quality in the software requirements specifications [10]	2018	IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT 2018
Adding quality in the user requirements specification: A first approach [11]	2016	1th International Conference on Digital Information Management, ICDIM 2016
A simplified systematic literature review: Improving Software Requirements Specification quality with boilerplates [3]	2015	
An automated approach to validate requirements specification [12]	2020	Compusoft
Requirements Specification-by-Example Using a Multi-objective Evolutionary Algorithm [13]	2016	
A Method for Verifying Non-Functional Requirements [14]	2017	Procedia Computer Science
The prevalence and severity of persistent ambiguity in software requirements specifications: Is a special effort needed to find them? [1]	2020	Science of Computer Programming

Table III: Continuación de la Tabla II

A framework to succeed the requirement specification document of IT projects in Morocco [15]	2019	ACM International Conference Proceeding Series
Problem of incompleteness in textual requirements specification [16]	2019	ICSOF 2019 - Proceedings of the 14th International Conference on Software Technologies
The detection of conflicts in the requirements specification based on an ontological model and a production rule system [17]	2019	CEUR Workshop Proceedings
Bridging the gap between natural language requirements and formal specifications [2]	2016	CEUR Workshop Proceedings
A framework for detecting ambiguity in software requirement specification [18]	2017	
ARSENAL: Automatic requirements specification extraction from natural language [19]	2016	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
Software requirement elicitation using natural language processing [20]	2016	Advances in Intelligent Systems and Computing
Requirements specification in the Prometheus methodology via activity diagrams [21]	2016	Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS
Requirements Evolution in Software Product Lines: An Empirical Study [22]	2015	
Descartes-Agent: Verifying Formal Specifications Using the Model Checking Technique [23]	2018	
Robust design requirements specification: a quantitative method for requirements development using quality loss functions [24]	2016	Journal of Engineering Design
A method of software requirements specification and validation for global software development [25]	2017	Requirements Engineering
Requirements elicitation and specification using the S-BPM paradigm [26]	2017	CEUR Workshop Proceedings
XML-DocTracker: Generating Software Requirements Specification (SRS) from XML Schema [27]	2017	ICISS 2016 - 2016 International Conference on Information Science and Security
Measuring software requirements specification quality [28]	2017	Journal of Telecommunication, Electronic and Computer Engineering
Empirical evaluation of formal method for requirements specification in agile approaches [29]	2018	ACM International Conference Proceeding Series
Use of formal languages to represent the ERTMS/ETCS system requirements specifications [30]	2015	



Table IV: Continuación de la Tabla II

Design semantic framework for software requirement specification [31]	2016	2016 6th International Conference on IT Convergence and Security, ICITCS 2016
An approach for detecting syntax and syntactic ambiguity in software requirement specification [32]	2018	Journal of Theoretical and Applied Information Technology
A tool for analyzing software requirements document quality [33]	2019	International Journal of Recent Technology and Engineering
A framework for software requirement ambiguity avoidance [34]	2019	International Journal of Electrical and Computer Engineering
DUSM: A Method for Requirements Specification and Refinement Based on Disciplined Use Cases and Screen Mockups [35]	2018	Journal of Computer Science and Technology
Back to basics: Extracting software requirements with a syntactic approach [36]	2018	CEUR Workshop Proceedings
Information technology of evaluating the sufficiency of information on quality in the software requirements specifications [37]	2018	CEUR Workshop Proceedings
Software visual specification for requirement specification validation [38]	2018	ACM International Proceeding Series
An empirical investigation on a tool-based boilerplate technique to improve software requirement specification quality [39]	2018	International Journal of Advanced Computer Science and Applications
Promirar: Tool for identifying and managing implicit Requirements in SRS documents [40]	2018	Lecture Notes in Engineering and Computer Science
A tool-based boilerplate technique to improve SRS quality: An evaluation [41]	2018	Journal of Telecommunication, Electronic and Computer Engineering
Patterns of Ambiguity in Textual Requirements Specification [42]	2019	Advances in Intelligent Systems and Computing
Patterns in textual requirements specification [43]	2018	ICSOFTE 2018 - Proceedings of the 13th International Conference on Software Technologies
Towards a model about quality of software requirements specification in agile projects [6]	2017	Proceedings - 2016 10th International Conference on the Quality of Information and Communications Technology, QUATIC 2016
Process to enhance the quality of software requirement specification document [44]	2018	International Conference on Engineering and Emerging Technologies, ICEET 2018

## IV. DISCUSIÓN

### A. Metodologías analizadas/ utilizadas en la especificación de requisitos de software

Los [20] [2] [41] [19] presentan información sobre las metodologías ARSENAL y SBVR, se divide en tres fases: i) Obtención de vocabulario, ii) las reglas semánticas de negocios, iii) Partiendo de un lenguaje semi formal o también conocido como lenguaje natural (NL), esto permite obtener un documento de SRS entendible para las partes interesadas. Los [13] [31] presentan información sobre la metodología del análisis semántico, que realiza el estudio de frases en función de su significado así como la interpretación de los sistemas informáticos del lenguaje natural. En [6] especifica la norma ISO-IEEE 830 como metodología de SRS, define ocho características de calidad para un buen SRS: i) corrección, ii) inequívoco, iii) completo, iv) consistente, v) prioritario, vi) verificable, vii) modificable, y viii) trazable. El [35] presenta información sobre la metodología DUSM, para describir y perfeccionar la SRS, define tres características: i) la definición de la forma de las especificaciones de requisitos mediante un meta-modelo, ii) la lista de limitaciones de buena forma y malos olores para esas especificaciones, y iii) la validación de DUSM mediante un estudio de caso. En [44] [29] proponen la metodología Moller, compuesto de cuatro procesos efectivos; cada proceso tiene su propia funcionalidad mejorando diferentes factores de atributos de calidad. En nuestro primer paso los requisitos de entrada se agregan en la fase de análisis que asegurarla integridad de los requisitos, especialmente el dominio requisitos del producto, luego los requisitos de salida son de entrada a la fase de mapeo que actúa como validación y verificación paso de los requisitos desde diferentes perspectivas de las partes interesadas. Después de eliminar la incorrección utilizando el proceso de mapeo, los requisitos se agregan en SRS y luego se proporcionan a externos para una mayor inspección.

En [30] propone una metodología basada en el lenguaje formal para la SRS, donde se ha referido a los siguientes pasos: I) análisis de todos los documentos (UNISIG, RFI SRS), II) elección del Modelo de Lenguaje Formal (Statecharts), III) definición de las reglas para los Modelos Statechart (condiciones de estado, transiciones de estado), IV) desarrollo del Modelo Statechart para el Subsistema de Trackside del ETCS, V) prueba del Modelo mediante herramientas específicas: i) Statemate Magnum Certifier para pruebas lógicas implementadas por el usuario, ii) evaluación automática del modelo mediante archivos Real Datalog. En [11] hace hincapié en la metodología DAQURES que tiene por objetivo guiar a los desarrolladores de SW durante la especificación y obtención de los requisitos de calidad de los datos, consta de dos fases: 1) Los diferentes roles involucrados en la metodología y actividades en las que hay una comunicación entre la especificación de requisitos. 2) Guiar a los desarrolladores de software durante la especificación y obtención de requisitos de calidad de datos. Los [10] [15] proponen la metodología evaluación de la suficiencia de la información sobre la calidad en las especificaciones de los requisitos de software, incluye:

1) modelos del dominio temático "Ingeniería de Software" (parte "Calidad de Software"): modelos teóricos establecidos de la calidad del software por SQuaRE (ISO 25010:2011) y para el análisis métrico; modelos del dominio temático "Ingeniería de Software" (partes "Calidad de Software", "Calidad de Software". Análisis métrico) basados en ontologías; modelos estructurales y modelos basados en la ontología del SRS (en cuanto a la información sobre la calidad); 2) modelos y métodos de evaluación de la suficiencia de la información sobre la calidad en el SRS: modelos del proceso de evaluación de la suficiencia de la información sobre la calidad para la evaluación de la calidad del software por la ISO 25010: 2011 y por el análisis métrico; métodos de evaluación de la suficiencia de la información de calidad en el SRS sobre la base de la ontología y la ontología ponderada; modos de generación y llenado del modelo de ontología para la calidad del software concreto; métodos de la conclusión lógica sobre la suficiencia o insuficiencia de la información sobre la calidad en el SRS. Por último, en [12] describe un enfoque como metodología, que menciona varios niveles, comenzando con el diagrama caso de uso y terminando con el logro de la integridad, utilizando una herramienta programada para verificar el proceso interno de cada nivel. Luego, debe recibir una retroalimentación informando al desarrollador que verifique el diagrama de caso de uso o el SRS. En síntesis, se determinó que la metodología que más impacto tiene en la SRS es ARSENAL [2] [20] ya que esta, se basa del NL y crear un documento legible en un lengua formal entendible para las partes interesadas(Stakeholder), así también, surgen ambigüedades al momento de construir este documento, para evitar estos problemas esta detalla tres fases de implementación para llevar de una manera más organizada.

### *B. Métodos analizados/ utilizadas en la especificación de requisitos de software*

En [6] [37] detallan el uso de métodos ágiles que son utilizados en la mayoría de los proyectos y proponen la adopción de procesos flexibles y adaptables, para aceptar los cambios como parte inseparable de su proceso de desarrollo. En [43] [42] [39] [38] mencionan el uso del método de inspección gramatical, basada en el reconocimiento de las dependencias y el etiquetado de las partes del discurso, identifica los roles gramaticales de las palabras en los requisitos textuales, es decir, objeto, sujeto, etc. Así como la clasificación para tratar un problema de sobrecarga y de conceptos de sinónimos, además utilizan una técnica repetitiva en investigaciones empíricas que evalúan la usabilidad de la plantilla con la finalidad de reducir la brecha entre los procesos de SRS con la psicología humana haciéndolo partícipe al factor humano. Los [35] [14] [20] [9] [17] [30] [25] describen la utilización de un método basado en prototipado para integrar de forma completa las maquetas de pantalla con los requisitos y hace hincapié a tres características: 1) Método de detección. 2) Método de verificación. 3) Verificar escenarios o descripciones de casos de uso, este proceso se descompone en varias fases: i) generación del gráfico de requisitos; ii) preparación del documento del

SRS; iii) revisión, actualización y finalización del documento del SRS en diferentes sitios de la GSD; iv) validación de requisitos en diferentes sitios de la GSD; v) validación de requisitos entre diferentes sitios de la GSD. En [32] [31] [20] [16] presentan información de un método que considera el enfoque para interpretar y detallar el análisis semántico de la especificación de requisitos en NL, son novedosos porque los diseñadores de software pueden refinar aún más la información recopilada y pueden desarrollar modelos orientados a objetos sólidos. En [21] [27] describen la utilización de un método de ingeniería inversa, que capturará la especificación en un esquema XML y la retendrá en la especificación UML y finalmente, situará la especificación en una forma estándar de documento SRS.

Los [41] [40] [39] presentan información sobre el método del juicio de los expertos, que es la codificación del conocimiento tácito en medidas probabilísticas asociadas con el nivel de cumplimiento de los requisitos de los programas informáticos, esta describe dos fases, en la primera fase se enfoca en la realización de un marco conceptual de un modelo tácito a explícito y la segunda fase se centra en la transferencia de conocimiento, también describe la realización de una investigación empírica, que evalúa la usabilidad de la plantilla basada en herramientas desde la perspectiva del usuario novato, y la técnica repetitiva se propone como un puente entre lo formal y lo informal en la especificación de requisitos.

De esta forma se determinó que el método basado en prototipados es el que más prioridad ha tenido por los autores en la SRS, ya que realiza un maquetado de pantallas en base a los requisitos permitiendo realizar diagramas de UML, así también, por un mal diseño se pueden encontrar inconsistencias esto llevaría al fracaso del proyecto, para evitar esto, [35] [14] [20] [9] [17] [30] [25] presenta tres bases que ayudan a disminuir las inconsistencias de los diagramas

### *C. Problemas encontrados en la especificación de requisitos*

En [5] [7] [9] [11] [14] [18] [21] [24] [28] [30] [32] [33] [35] [36] [39][37][41] hace referencia a uno de los problemas más comunes en la SRS que son las ambigüedades: i) Ambigüedad léxica. ii) Ambigüedad sintáctica o estructural. iii) Ambigüedad semántica iv) Ambigüedad pragmática v) Ambigüedad de la sintaxis, que son ocasionados por la omisión de información, la mala interpretación y la incompatibilidad e inconsistencia lo que pone en duda la calidad de la SRS. Los [4] [10] [19] [20] [23] [25] [33] presenta información de la falta de modelos, métodos y herramientas para evaluar la suficiencia de información sobre calidad en la especificación de requisitos; obteniendo como resultado el uso de técnicas informales que conlleva a requisitos ambiguos. Estos problemas resultan en una implementación incorrecta y reelaboración de ellos mismo.

En resumen, uno de los mayores problemas que existen en la SRS es la ambigüedad [5] [7] [9] [11] [14] [18] [21] [24] [28] [30] [32] [33] [35] [36] dados por la omisión de información y mala interpretación por la falta de modelos, métodos y

herramientas; obteniendo como resultado el uso de técnicas informales lo que pone en duda la calidad de los SRS.

#### *D. Software/herramientas utilizadas por los investigadores para la especificación de requisitos*

En [38] detalla el uso de la herramienta TEMOS (Textual Modelling System), que fue diseñado e implementado como una aplicación multihilo del lado del cliente. Su mapeo a los fragmentos UML por el usuario, esta proporciona un procesamiento de texto automático y mapeo automático de fragmentos. Debido a los requisitos específicos de los dominios, maneja los términos en un glosario editable. Utiliza recursos en línea pero puede ser utilizada sin acceso a internet. En [5] [37] detallan la herramienta BOILERPLATE que es utilizado en herramientas para transformar semiautomáticamente el NL en requisitos semiformal, llamado DODT. La herramienta puede reducir el esfuerzo de transformación y para mejorar la calidad de la ingeniería de requisitos. Además, EARS boilerplate es uno de las mejores herramientas que se han estado utilizando para superar la problemática del LN. Además, el texto estándar es eficaz para reducir la ambigüedad y haciéndolos más adaptables al análisis automatizado. En [36] describe la herramienta PROMIRAR (producto para la gestión requisitos implícitos que utilizan el razonamiento basado en la analogía), es un software que identifica y gestiona la especificación de requisitos. Esto implican un método manual para extraer oraciones límites del documento de requisitos y reemplazándolos adicionalmente en tablas, imágenes y figuras en el formato correspondiente. En [34] menciona una herramienta de soporte (SVSTool) que va de la mano con el principio de especificación de visualización de software denominada ingeniería de requisitos (especificación) para la validación del factor humano (RESHFv), que tiene como objetivo reducir la brecha entre los procesos de especificación de requisitos con psicología humana al involucrar activamente al factor humano, durante la validación de la especificación. En el proceso de RESHFv, se describe que la SRS se valida con los cinco atributos de las cualidades de la especificación correcta, verificable, comprensible e inequívoco que es el resultado esperado del marco RESHFv. En [30] menciona las herramientas automatizadas SREE, QuARS, RESI, ARM, NAI, NL2OCL y WSD, etc., utilizados para detectar y resolver ambigüedades. El enfoque está dado en la aplicación de W6H en los requisitos obtenidos para manejar el problema de la ambigüedad. Tanto la obtención de requisitos como la superación de brechas, son tareas desafiantes generalizadas. Utilizando el enfoque estructurado de patrones W6H podemos abordar estos problemas, planteando una serie de interrogantes que permiten eliminar la ambigüedad provocada por los mismos requisitos inicialmente. En [29] detalla la herramienta Quality Checker que mide la calidad del documento SRS basado en requisitos de calidad de la oración (RSQ) y Calidad de documentos de requisitos (RDQ), la evaluación es mediante el reconocimiento de los marcadores y llamar la atención sobre las oraciones que contienen errores potenciales y ambigüedades. El documento se utiliza durante la revisión y sesiones donde el

resultado apoya a los revisores en términos de detección de inexactitudes, ambigüedades y lingüística sin consistencias en el documento. En [24] [26] detalla la herramienta Magnum Statemate Certifier, que es capaz de verificar la corrección y la evolución del modelo, también con una simulación de panel. Y las representaciones están respaldadas por herramientas de software que proporcionan potentes pruebas de verificación en los niveles inferiores de diseño del sistema, siguiendo los niveles de especificación más altos, mediante pruebas matemáticas automatizadas. De hecho, la lógica del sistema implementada mediante un lenguaje formal, permite utilizar herramientas para la verificación del cumplimiento del modelo especificaciones, la verificación automática de inconsistencias en la lógica de funcionamiento, como la lógica de los conflictos, el indeterminismo, estados inaccesibles, y para comprobar la accesibilidad a los estados por condiciones específicas, de acuerdo con los requisitos. En [23] detalla la herramienta XML\_DocTracker que puede generar el SRS del esquema XML, así como también puede detectar documentos con cambios en el mismo debido al factor de trazabilidad que está incrustado dentro de la herramienta. En el resultado del caso de estudio aplicando esta herramienta ha sido satisfactoria pues genera un nuevo SRS automáticamente. Esto muestra una tasa de éxito del 100% de generación manual a automática de SRS, es una alternativa adecuada para los ingenieros de requisitos. En [18] menciona una herramienta estadística llamada SPSS, la cual es el soporte y necesidad, para reducir el lenguaje natural dentro de la identificación de escenarios de evolución, que requerirán un proceso más formal; apoyando la evolución de características con dependencias (requiere y excluye) en el modelo de características; y lidiar con la evolución de fusiones de casos de uso, que forman parte de la ER. En [21] se detalla que PROMETHEUS es una herramienta que aplica un conjunto de reglas que combinan el escenario especificado con su información relevante, da resumen de objetivos para generar la descripción abstracta y Luego, usa la descripción abstracta para generar un gráfico DOT script de origen, que Graphviz puede usar para generar un gráfico de representación del diagrama de actividad, Los autores de desarrollaron esta herramienta para de esta manera con el método de prototipado se pudiese comprender de una mejor manera la recolección de información para la SRS. En [20] detalla la herramienta openNLP la cual proporciona etiquetas de partes del habla (POS) muy relevantes. Estas partes del etiquetado del habla del SRS son bastante útil para una mayor identificación de elementos orientados a objetos como: clases, objetos, atributos, relaciones, etc. Esta herramienta puede ser utilizada para para la verificación del cumplimiento del modelo de especificaciones, la verificación automática de inconsistencia en la lógica de funcionamiento y aumentar la calidad de requisitos evitando estructuras complejas e inconsistencia en los requisitos. En [15] [12] detallan las herramientas llamadas NLARE (A Natural Language Processing Tool for Automatic Requirements) y X-JRDL en donde utiliza un requisito funcional como un conjunto de palabras en el que las palabras pueden formar uno de los tres elementos siguientes: actor (realiza la acción en el enunciado), función (indica qué acción debe realizarse),



o detalle (indica las condiciones en las que se espera que se actúe). Estas herramientas pretenden mejorar las características de la especificación de requisitos funcionales del software de manera que el usuario se sienta cómodo al momento de la implementación y creación de SRS para la implementación y automatización de sistemas.

Como se ha visto a lo largo de la presentación de herramientas se recomienda la Quality Checker [29], evalúa la calidad de oraciones y documentos, resaltando aquellas que contienen errores potenciales y ambigüedades, de esta manera el apoyo es directo para los ingenieros en requisitos en términos de detección de inexactitudes, ambigüedades y lingüísticas

## V. CONCLUSIONES

Se seleccionaron 40 artículos y los resultados muestran que existe gran variedad de metodologías, pero entre las más utilizadas se encuentra ARSENAL que tuvo mayor impacto dentro de los investigadores por sus fases: i) Obtención de vocabulario, ii) las reglas semánticas de negocios, iii) Partiendo de un lenguaje definido como semi formal o también conocido como lenguaje natural (NL), permitiendo disminuir las ambigüedades en la SRS. La revisión muestra que hay un gran número de estudios relacionados con SRS en los que se utilizan diferentes métodos y herramientas para la extracción de la información concreta, así mismo se indican los problemas al momento de extraer la información a partir del NL. Por último, el SRS es un documento formal en donde se especifican los requisitos basados en NL y los modelos para el desarrollo de un sistema software, el mismo que es entendible para la parte interesada (Stakeholder)

## REFERENCIAS

- [1] C. Ribeiro and D. Berry, "The prevalence and severity of persistent ambiguity in software requirements specifications: Is a special effort needed to find them?," *Science of Computer Programming*, vol. 195, p. 102472, sep 2020.
- [2] M. Bösch, R. Bogusch, A. Fraga, and C. Rudat, "Bridging the gap between natural language requirements and formal specifications," in *CEUR Workshop Proceedings*, vol. 1564, CEUR-WS, 2016.
- [3] U. Anuar, S. Ahmad, and N. A. Emran, "A simplified systematic literature review: Improving Software Requirements Specification quality with boilerplates," in *2015 9th Malaysian Software Engineering Conference, MySEC 2015*, pp. 99–105, Institute of Electrical and Electronics Engineers Inc., may 2016.
- [4] J. Camacho, K. Campos, P. Cedillo, B. Coronel, and A. Bermeo, "Forensics Analysis on Mobile Devices: A Systematic Mapping Study," in *Advances in Intelligent Systems and Computing*, vol. 884, pp. 57–72, Springer Verlag, nov 2019.
- [5] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Systems and Software*, vol. 80, pp. 571–583, apr 2007.
- [6] J. Medeiros, M. Goulao, A. Vasconcelos, and C. Silva, "Towards a model about quality of software requirements specification in agile projects," in *Proceedings - 2016 10th International Conference on the Quality of Information and Communications Technology, QUATIC 2016*, pp. 236–241, Institute of Electrical and Electronics Engineers Inc., jan 2017.
- [7] F. W. Neiva, J. M. N. David, R. Braga, and F. Campos, "Towards pragmatic interoperability to support collaboration: A systematic review and mapping of the literature," *Information and Software Technology*, vol. 72, pp. 137–150, apr 2016.
- [8] J. Popay, L. Arai, and N. Britten, "Guidance on the conduct of narrative synthesis in systematic reviews: A product from the ESRC Methods Programme," 2006.
- [9] Y. Priyadi, A. Djunaidy, and D. Siahaan, "Requirements Dependency Graph Modeling on Software Requirements Specification Using Text Analysis," in *2019 1st International Conference on Cybernetics and Intelligent System, ICORIS 2019*, pp. 221–226, Institute of Electrical and Electronics Engineers Inc., aug 2019.
- [10] T. Hovorushchenko and O. Pomorova, "Methodology of evaluating the sufficiency of information on quality in the software requirements specifications," in *Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT 2018*, pp. 370–374, Institute of Electrical and Electronics Engineers Inc., jul 2018.
- [11] C. Guerra-Garcia, I. Caballero, M. Cardenas-Juarez, and R. Samano-Robles, "Adding quality in the user requirements specification: A first approach," in *2016 11th International Conference on Digital Information Management, ICDIM 2016*, pp. 79–84, Institute of Electrical and Electronics Engineers Inc., 2016.
- [12] Y. Al-Kasabera, W. Alzyadat, A. Alhroob, S. Al Showarah, and A. Thunibat, "An automated approach to validate requirements specification," *Compusoft*, vol. 9, pp. 3578–3585, feb 2020.
- [13] L. Van Rooijen and H. Hamann, "Requirements specification-by-example using a multi-objective evolutionary algorithm," in *Proceedings - 2016 IEEE 24th International Requirements Engineering Conference Workshops, REW 2016*, pp. 3–9, Institute of Electrical and Electronics Engineers Inc., jan 2017.
- [14] Y. Matsumoto, S. Shirai, and A. Ohnishi, "A Method for Verifying Non-Functional Requirements," in *Procedia Computer Science*, vol. 112, pp. 157–166, Elsevier B.V., jan 2017.
- [15] R. Hassani and Y. El Bouzekri El Idrissi, "A framework to succeed the requirement specification document of IT projects in Morocco," in *ACM International Conference Proceeding Series*, vol. Part F1482, pp. 138–144, Association for Computing Machinery, 2019.
- [16] D. Šenkýř and P. Kroha, "Problem of incompleteness in textual requirements specification," in *ICSOF 2019 - Proceedings of the 14th International Conference on Software Technologies*, pp. 323–330, SciTePress, 2019.
- [17] M. S. Murtazina and T. V. Avdeenko, "The detection of conflicts in the requirements specification based on an ontological model and a production rule system," in *CEUR Workshop Proceedings*, vol. 2416, pp. 63–73, CEUR-WS, 2019.
- [18] A. O. J. ale Sabriye and W. M. N. W. Zainon, "A framework for detecting ambiguity in software requirement specification," in *ICIT 2017 - 8th International Conference on Information Technology, Proceedings*, pp. 209–213, Institute of Electrical and Electronics Engineers Inc., oct 2017.
- [19] S. Ghosh, D. Elenius, W. Li, P. Lincoln, N. Shankar, and W. Steiner, "ARSINAL: Automatic requirements specification extraction from natural language," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9690, pp. 41–46, Springer Verlag, 2016.
- [20] M. Mohanan and P. Samuel, "Software requirement elicitation using natural language processing," in *Advances in Intelligent Systems and Computing*, vol. 424, pp. 197–208, Springer Verlag, 2016.
- [21] Y. Abushark, J. Thangarajah, T. Miller, M. Winikoff, and J. Harland, "Requirements specification in the prometheus methodology via activity diagrams," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pp. 1247–1248, International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2016.
- [22] R. P. D. Oliveira and E. S. D. Almeida, "Requirements Evolution in Software Product Lines: An Empirical Study," in *Proceedings - 2015 9th Brazilian Symposium on Software Components, Architectures and Reuse, SBCARS 2015*, pp. 1–10, Institute of Electrical and Electronics Engineers Inc., oct 2015.
- [23] V. H. Subburaj and J. E. Urban, "Descartes-Agent: Verifying Formal Specifications Using the Model Checking Technique," in *Proceedings - 2nd IEEE International Conference on Robotic Computing, IRC 2018*, vol. 2018-Janua, pp. 392–398, Institute of Electrical and Electronics Engineers Inc., apr 2018.
- [24] S. N. Pedersen, M. E. Christensen, and T. J. Howard, "Robust design requirements specification: a quantitative method for requirements development using quality loss functions," *Journal of Engineering Design*, vol. 27, pp. 544–567, aug 2016.
- [25] N. Ali and R. Lai, "A method of software requirements specification and validation for global software development," *Requirements Engineering*, vol. 22, pp. 191–214, jun 2017.



- [26] C. Stary, "Requirements elicitation and specification using the S-BPM paradigm," in *CEUR Workshop Proceedings*, vol. 1796, CEUR-WS, 2017.
- [27] H. Aman and R. Ibrahim, "XML-DocTracker: Generating Software Requirements Specification (SRS) from XML Schema," in *ICISS 2016 - 2016 International Conference on Information Science and Security*, Institute of Electrical and Electronics Engineers Inc., mar 2017.
- [28] A. Nordin, N. H. Ahmad Zaidi, and N. A. Mazlan, "Measuring software requirements specification quality," in *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, pp. 123–128, Universiti Teknikal Malaysia Melaka, 2017.
- [29] P. Rodrigues, J. P. S. Da Silva, M. Ecar, G. T. Guedes, S. V. Menezes, and E. M. Rodrigues, "Empirical evaluation of formal method for requirements specification in agile approaches," in *ACM International Conference Proceeding Series*, pp. 416–423, Association for Computing Machinery, jun 2018.
- [30] A. Piccolo, V. Galdi, F. Senesi, and R. Malangone, "Use of formal languages to represent the ERTMS/ETCS system requirements specifications," in *Electrical Systems for Aircraft, Railway and Ship Propulsion, ESARS*, vol. 2015-May, IEEE Computer Society, may 2015.
- [31] T. Hassan and S. Hassan, "Design semantic framework for software requirement specification," in *2016 6th International Conference on IT Convergence and Security, ICITCS 2016*, Institute of Electrical and Electronics Engineers Inc., nov 2016.
- [32] A. O. J. Sabriye and W. M. N. Wan Zainon, "An approach for detecting syntax and syntactic ambiguity in software requirement specification," *Journal of Theoretical and Applied Information Technology*, vol. 96, pp. 2275–2284, apr 2018.
- [33] C. Arun, S. Karthick, G. Sivan Muthu Krishnan, and S. Dutta, "A tool for analyzing software requirements document quality," *International Journal of Recent Technology and Engineering*, vol. 8, pp. 274–279, jul 2019.
- [34] A. K. Gupta and A. Deraman, "A framework for software requirement ambiguity avoidance," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 6, pp. 5436–5445, 2019.
- [35] G. Reggio, M. Leotta, F. Ricca, and D. Clerissi, "DUSM: A Method for Requirements Specification and Refinement Based on Disciplined Use Cases and Screen Mockups," *Journal of Computer Science and Technology*, vol. 33, pp. 918–939, sep 2018.
- [36] M. Caron, F. S. Bäumer, and M. Geierhos, "Back to basics: Extracting software requirements with a syntactic approach," in *CEUR Workshop Proceedings*, vol. 2075, CEUR-WS, 2018.
- [37] T. Hovorushchenko and O. Pomorova, "Information technology of evaluating the sufficiency of information on quality in the software requirements specifications," in *CEUR Workshop Proceedings*, vol. 2104, pp. 555–570, CEUR-WS, 2018.
- [38] F. Binti Zainuddin, R. Bin Abdullah Arshah, and R. Binti Mohamad, "Software visual specification for requirement specification validation," in *ACM International Conference Proceeding Series*, pp. 66–71, Association for Computing Machinery, feb 2018.
- [39] U. Anuar, S. Ahmad, and N. A. Emran, "An empirical investigation on a tool-based boilerplate technique to improve software requirement specification quality," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 12, pp. 397–401, 2018.
- [40] O. Emebo, D. Olawande, and A. Charles, "Promirar: Tool for identifying and managing implicit Requirements in SRS documents," in *Lecture Notes in Engineering and Computer Science*, vol. 2237, pp. 108–112, Newswood Limited, 2018.
- [41] S. Ahmad, U. Anuar, and N. A. Emran, "A tool-based boilerplate technique to improve SRS quality: An evaluation," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, no. 2-7, pp. 111–114, 2018.
- [42] D. Šenkýř and P. Kroha, "Patterns of Ambiguity in Textual Requirements Specification," in *Advances in Intelligent Systems and Computing*, vol. 930, pp. 886–895, Springer Verlag, 2019.
- [43] D. Šenkýř and P. Kroha, "Patterns in textual requirements specification," in *ICSOF 2018 - Proceedings of the 13th International Conference on Software Technologies*, pp. 197–204, SciTePress, 2018.
- [44] S. W. Ali, Q. A. Ahmed, and I. Shafi, "Process to enhance the quality of software requirement specification document," in *2018 International Conference on Engineering and Emerging Technologies, ICEET 2018*, vol. 2018-Janua, pp. 1–6, Institute of Electrical and Electronics Engineers Inc., apr 2018.