

Effect of requirements specification on the quality of computer systems

Ammy Ashli Arizaga Umbo¹; Martin Alonso Pingo Valdiviezo²; Francisco Alonso Donayre Monteza³

^{1,1} *Universidad Tecnológica del Peru, Piura, Peru, u21313705@utp.edu.pe,
u21322350@utp.edu.pe, c29224@utp.edu.pe*

Abstract– Requirement’s engineering is a critical phase in the software life cycle, yet many projects still fail due to ambiguous, incomplete or weakly managed requirements. This systematic literature review analyses how the specification and management of software requirements influence the quality of information systems. The study compares formal and informal practices across different project contexts and identifies the metrics, attributes and standards used to assess their impact on reliability, maintainability, usability, efficiency and security/explainability. The research question was structured using the PICO framework and the search strategy followed PRISMA-S guidelines. Searches were executed in Scopus, Web of Science and EBSCO for the period 2020–2025, applying predefined inclusion and exclusion criteria. Out of 1,083 records, 38 primary studies were selected for detailed analysis. The results show that projects adopting explicit requirements engineering practices—such as structured templates, traceability, formal reviews and systematic change management—report fewer post-release defects, lower rework effort and higher project success than those relying on minimal or informal documentation. Non-functional requirements, particularly reliability, performance, security, usability and explainability, emerge as key drivers of architecture and long-term quality. Most studies employ at least five complementary indicators combining product, process and user-oriented measures, and recent works incorporate AI-based techniques (NLP, ensemble learning, LLMs) to support classification and quality assurance of requirements. The review concludes with practical guidelines for strengthening requirements engineering in both traditional and agile development settings.

Keywords– Software requirements specification; Software quality; Requirements engineering; Software requirements; Quality of developed systems.

Efecto de la especificación de requisitos en la calidad de los sistemas informáticos

Ammy Ashli Arizaga Umbo¹; Martin Alonso Pingo Valdiviezo²; Francisco Alonso Donayre Monteza³

^{1,)} *Universidad Tecnológica del Perú, Piura, Perú, u21313705@utp.edu.pe ,*

u21323350@utp.edu.pe, c29224@utp.edu.pe

Resumen— *La ingeniería de requisitos constituye una fase crítica del ciclo de vida del software; sin embargo, muchos proyectos siguen fracasando debido a requisitos ambiguos, incompletos o gestionados de manera informal. Esta revisión sistemática analiza cómo la especificación y gestión de requisitos de software influyen en la calidad de los sistemas informáticos. Se comparan prácticas formales e informales en distintos contextos de proyecto y se identifican las métricas, atributos y estándares utilizados para evaluar su impacto en la fiabilidad, mantenibilidad, usabilidad, eficiencia y seguridad/explicabilidad. La pregunta de investigación se formuló mediante el enfoque PICO y la estrategia de búsqueda siguió las directrices PRISMA-S. Se realizaron búsquedas en Scopus, Web of Science y EBSCO para el periodo 2020–2025, aplicando criterios de inclusión y exclusión predefinidos. De 1083 registros iniciales, se seleccionaron 38 estudios primarios para análisis detallado. Los resultados muestran que los proyectos que aplican una ingeniería de requisitos explícita —plantillas estructuradas, trazabilidad, revisiones formales y gestión sistemática de cambios— reportan menos defectos post-entrega, menor retrabajo y mayor probabilidad de éxito que aquellos con prácticas mínimas. Los requisitos no funcionales, en especial fiabilidad, rendimiento, seguridad, usabilidad y explicabilidad, se confirman como determinantes de la arquitectura y de la calidad a largo plazo. La mayoría de estudios emplea cinco o más indicadores complementarios y las investigaciones más recientes incorporan técnicas basadas en IA para apoyar la clasificación y aseguramiento de la calidad de los requisitos. La revisión concluye con lineamientos prácticos para fortalecer la ingeniería de requisitos en contextos de desarrollo tradicionales y ágiles.*

Palabras clave— *Especificación de requisitos de software; Calidad del software; Ingeniería de requisitos; Requisitos de software; Calidad de sistemas desarrollados.*

I. INTRODUCCIÓN

La especificación de requisitos de software es una fase clave del ciclo de vida de desarrollo, porque formaliza las necesidades y expectativas de los usuarios que orientan el diseño, la implementación y la evaluación de los sistemas. Una gestión adecuada de esta etapa mejora la transparencia, reduce ambigüedades y contribuye a elevar la calidad del producto final [1]. La calidad de software, entendida como el grado en que el producto satisface requisitos explícitos e implícitos, incluye atributos como fiabilidad, usabilidad, mantenibilidad y eficiencia, recogidos en estándares como ISO/IEC 25010. La literatura propone desde métricas para evaluar y mejorar los procesos de ingeniería de software [1], hasta la aplicación de estándares de medición [2] y técnicas como las tablas de decisión para fortalecer la definición de requisitos [3]. Asimismo, se ha evidenciado que errores, ambigüedades y cambios no controlados en los requisitos se asocian directamente con fallos de proyecto y baja calidad del sistema [4].

Sin embargo, la práctica profesional sigue mostrando limitaciones importantes. En numerosos proyectos, la baja calidad y la alta volatilidad de los requisitos generan sobrecostos, retrasos y defectos en los sistemas desarrollados, afectando su fiabilidad y sostenibilidad [4]. Además, muchos estudios se centran en contextos muy específicos —por ejemplo, programas de medición en organizaciones pequeñas [2] o técnicas puntuales como entrevistas y prototipos [3]— sin ofrecer una visión integrada del impacto de la especificación de requisitos sobre los distintos atributos de calidad del software. Esta fragmentación dificulta identificar patrones comunes, comparar enfoques y derivar lineamientos aplicables en diversos entornos de desarrollo.

Ante este panorama, se considera necesaria una revisión sistemática de literatura (RSL) que organice y sintetice la evidencia disponible sobre la relación entre especificación de requisitos y calidad de los sistemas informáticos. En la presente revisión sistemática se pretende analizar de manera crítica cómo la especificación y gestión de requisitos de software influye en la calidad de los sistemas informáticos, identificando y comparando enfoques formales e informales utilizados en distintos tipos de proyectos, así como las métricas, atributos y estándares empleados para evaluar su impacto en la fiabilidad, mantenibilidad, usabilidad, eficiencia y seguridad/explicabilidad del software, con el fin de proponer

lineamientos prácticos que orienten la mejora de la ingeniería de requisitos en contextos de desarrollo tradicionales y ágiles.

En tal sentido, el documento se organiza de la siguiente manera. La Sección II describe la metodología empleada para la RSL, incluyendo la formulación de la pregunta de investigación mediante PICO, la estrategia de búsqueda en Scopus, Web of Science y EBSCO, y los criterios de elegibilidad aplicados. La Sección III presenta los resultados, divididos en datos bibliométricos y datos de contenido organizados según las RQ1–RQ4. La Sección IV desarrolla la discusión crítica de los hallazgos y sus implicancias para la ingeniería de requisitos. Finalmente, la Sección V expone las conclusiones, recomendaciones para la práctica y posibles líneas de investigación futura.

II. METODOLOGÍA

Con el propósito de examinar sistemáticamente la producción científica sobre la influencia de la especificación de requisitos de software en la calidad de los sistemas desarrollados, se llevó a cabo una Revisión Sistemática de Literatura (RSL). Para estructurar la pregunta de investigación y orientar la estrategia de búsqueda se utilizó el enfoque PICO, lo que permitió definir de manera clara los elementos centrales del estudio y recuperar literatura relevante. Posteriormente, se siguieron los lineamientos de la declaración PRISMA, garantizando transparencia y rigurosidad en la identificación, selección e inclusión de los artículos. De acuerdo con lo anterior, a continuación, se presenta la construcción de la pregunta de investigación bajo el esquema PICO. Esta formulación permitió derivar subpreguntas específicas que guiaron el proceso de búsqueda y depuración de los estudios seleccionados.

TABLA I
DESCRIPCION DE LA PREGUNTA PICO

Tema de investigación: Influencia de la especificación de requisitos de software en la calidad de los sistemas desarrollados: una revisión sistemática de literatura.	
Pregunta general: ¿De qué manera la especificación y gestión de requisitos influye en la calidad del software en proyectos de desarrollo de sistemas informáticos, en comparación con prácticas no formales o débiles de especificación?	
Acrónimo y componente	Subpregunta
P: Especificación y gestión de requisitos	¿Cómo se ha definido y caracterizado la especificación/gestión de requisitos?
I: Calidad del software	¿En qué contextos/proyectos se estudia la relación RE ↔ calidad?
C: Proyectos de Software	¿Qué técnicas/estándares/herramientas se usan para mejorar la especificación?
O: Atributos de calidad	¿Qué indicadores varían con la calidad de la especificación y cuál es el tamaño del efecto reportado? (defectos post-entrega, retrabajo, fiabilidad, mantenibilidad, usabilidad, eficiencia, seguridad/explicabilidad, satisfacción del usuario, tiempo/costo)

En segundo lugar, la Tabla II resume la selección de términos clave para cada componente del modelo PICO, considerando requisitos de software, atributos de calidad y contexto de desarrollo.

TABLA II
DESCRIPCION DE PALABRAS CLAVE

Acrónimo y componente	Palabras clave
P: Especificación y gestión de requisitos	software requirements specification, requirements engineering, requirements management, requirements documentation
I: Calidad del software	software quality, system quality, software product quality, software reliability, software maintainability, software usability, software efficiency
C: Proyectos de Software	software project, software system, software development, information system
O: Atributos de calidad	software quality, reliability, maintainability, usability, efficiency, security, explainability, defects, rework, user satisfaction, project success

Cerrando el método PICO, se muestran en la Tabla III las ecuaciones de búsqueda aplicadas en las bases de datos seleccionadas (Scopus, Web of Science y EBSCO), integrando palabras clave en español e inglés mediante operadores booleanos, lo cual permitió recuperar literatura relevante para los fines de esta revisión [1].

Se realizó la búsqueda de estudios científicos utilizando ecuaciones diseñadas específicamente para cada base de datos: SCOPUS (EB1), Web of Science (EB2) y EBSCO(EB3). Dichas fórmulas incluyeron términos derivados del modelo PICO, combinados con operadores lógicos para maximizar la pertinencia y exhaustividad de los resultados [2].

TABLA III
DESCRIPCION DE LAS ECUACIONES DE BÚSQUEDA

Acrónimo	Ecuación de búsqueda
P	"software requirements specification" OR "requirements engineering" OR "requirements management" OR "requirements documentation"
I	"software quality" OR "system quality" OR "software product quality" OR "software reliability" OR "software maintainability" OR "software usability" OR "software efficiency"
C	"software project*" OR "software system*" OR "software development" OR "information system*"
O	
EB1 – SCOPUS:	"software requirements specification" OR "requirements engineering" OR "requirements management" OR "requirements documentation" AND "software quality" OR "system quality" OR "software product quality" OR "software reliability" OR "software maintainability" OR "software usability" OR "software efficiency" AND "software project*" OR "software system*" OR "software development" OR "information system*"
EB2 – WEB OF SCIENCE:	"software requirements specification" OR "requirements engineering" OR "requirements management" OR "requirements documentation" (Topic) AND "software quality" OR "system quality" OR "software product quality" OR "software reliability" OR "software maintainability" OR "software usability" OR "software efficiency" (Topic) AND "software project*" OR "software system*" OR "software development" OR "information system*"
EB3 – EBSCO:	("software requirements specification" OR "requirements

engineering" OR "requirements management" OR "requirements documentation") AND TX ("software quality" OR "system quality" OR "software product quality" OR "software reliability" OR "software maintainability" OR "software usability" OR "software efficiency" OR "ISO/IEC 25010") AND TX ("software project*" OR "software system*" OR "software development" OR "information system*")

Una vez recopilados los registros mediante las ecuaciones de búsqueda, se aplicaron criterios de elegibilidad con el fin de asegurar la pertinencia, actualidad y rigurosidad metodológica de los estudios seleccionados. La Tabla IV presenta los criterios de inclusión y exclusión para esta revisión sistemática, orientados a filtrar aquellos artículos que se alinearan con los objetivos de investigación centrados en la influencia de la especificación de requisitos de software en la calidad de los sistemas desarrollados [3].

TABLA IV DESCRIPCION DE LOS CRITERIOS DE ELEGIBILIDAD			
N	CRITERIOS DE INCLUSION	N	CRITERIOS DE EXCLUSION
CI1	Artículos que analicen atributos de calidad (fiabilidad, mantenibilidad, usabilidad, eficiencia) o métricas de calidad vinculadas a los requisitos.	CE1	Investigaciones que no detallen de forma precisa el enfoque metodológico empleado.
CI2	Inglés y español.	CE2	Estudios previos a 2020 (fuera del período de interés).
		CE3	Artículos sin acceso al texto completo.

Como resultado de la aplicación de las estrategias de búsqueda definidas, se localizaron 1083 registros distribuidos en tres bases de datos científicas: Scopus (n = 472), Web of Science (n = 16) y EBSCO (n = 595). Tras una depuración inicial, se identificaron y eliminaron 4 registros duplicados, quedando 1079 documentos únicos para su evaluación [4].

Durante la fase de selección preliminar, se excluyeron 893 publicaciones a partir de la revisión del título y resumen (abstract), eliminando principalmente aquellos estudios que no abordaban de manera explícita la especificación de requisitos de software ni su relación con los atributos de calidad del sistema (como fiabilidad, usabilidad, mantenibilidad o eficiencia (CI1) [5].

Posteriormente, se llevó a cabo una fase de filtrado secundario mediante la lectura de la introducción de los estudios preseleccionados. Esta etapa permitió identificar trabajos con enfoques no directamente vinculados con la investigación o que trataban aspectos tangenciales. Como resultado, se seleccionaron 186 estudios con texto completo disponible para su evaluación exhaustiva (CE3) [1].

Finalmente, con la última fase de selección, se procedió a la lectura integral de los artículos restantes, lo cual permitió confirmar el cumplimiento de los criterios metodológicos y la relevancia de los resultados. En este proceso se excluyeron 148 artículos adicionales, entre ellos 42 por ausencia de descripción metodológica (CE1)[2] criterio que operó como

una evaluación de calidad para minimizar el sesgo de selección, 38 por idioma no permitido (CI2)[3], 44 por año de publicación anterior a 2020 (CE2)[4] y 24 por falta de acceso al texto completo (CE3) quedándonos con un total de 38 artículos seleccionados para RSL. Para garantizar la objetividad y reducir el riesgo de sesgo en esta selección, el proceso de cribado fue realizado de manera independiente por dos revisores. En los casos donde se presentaron desacuerdos o discrepancias sobre la elegibilidad de un estudio, estos se resolvieron mediante discusión y consenso, recurriendo a la mediación de un tercer revisor experto cuando fue necesario para alcanzar la decisión final.

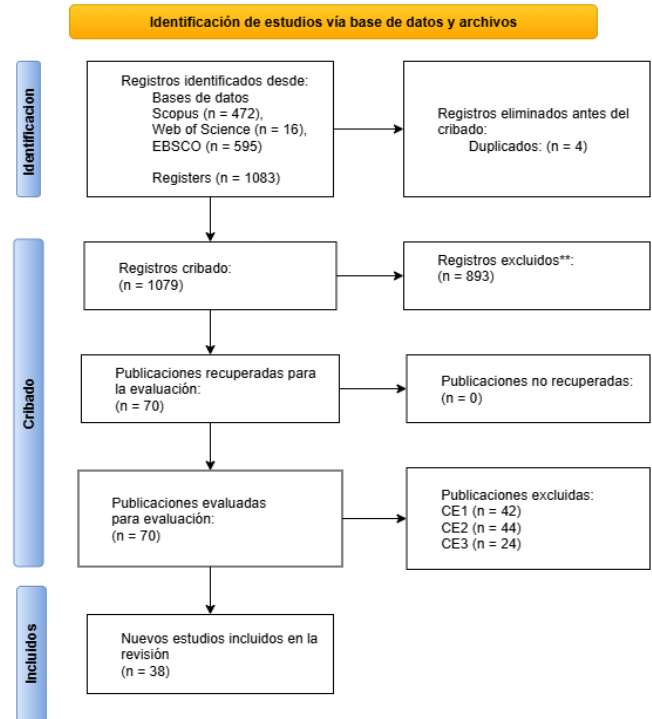


Fig. 1 Número de estudios vía base de datos

III. RESULTADOS

En esta sección se presentan los hallazgos de la revisión sistemática de literatura sobre el efecto de la especificación de requisitos en la calidad de los sistemas informáticos.

Primero se describen los datos bibliométricos, que caracterizan la producción científica en términos de volumen, evolución temporal y enfoques metodológicos. Luego se analizan los datos de contenido, organizados según las cinco preguntas de investigación (RQ1–RQ4).

DATOS BIBLIOMÉTRICOS

La revisión incluye 38 artículos publicados entre 2020 y 2025, todos vinculados a la especificación o gestión de requisitos y su relación con la calidad del software [7], [9].

La evolución temporal muestra una producción moderada en los primeros años (6 estudios en 2020, 2 en 2021, 3 en 2022 y 3 en 2023) y un incremento notable a partir de 2024, con 11 artículos en 2024 y 13 en 2025, lo que sugiere en esas 3 bases bibliográficas hay una consolidación reciente del tema como línea de investigación prioritaria en ingeniería de software [7], [1], [10], [12].

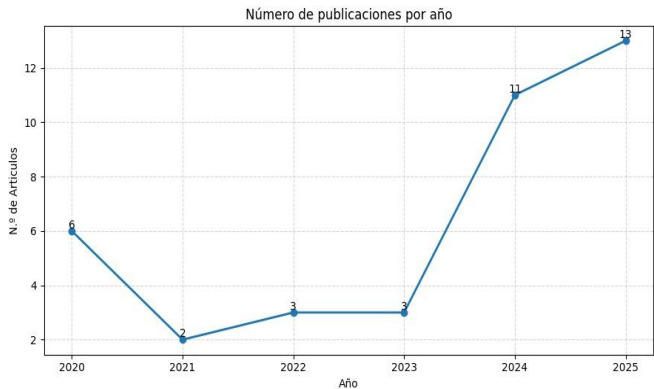


Fig. 2 Número de artículos por año incluidos en la RSL (2020–2025).

En cuanto al enfoque metodológico, se adopta un enfoque cuantitativo básico–descriptivo, complementado con estudios cualitativos y mixtos de carácter aplicado.

En total se identifican 13 estudios cuantitativos básicos, 10 cualitativos (2 básicos y 8 aplicados) y 15 mixtos (3 descriptivos y 12 aplicados–experimentales), centrados en evaluar métricas, modelos o herramientas para mejorar la especificación de requisitos y su impacto en la calidad del software [7], [8], [12].

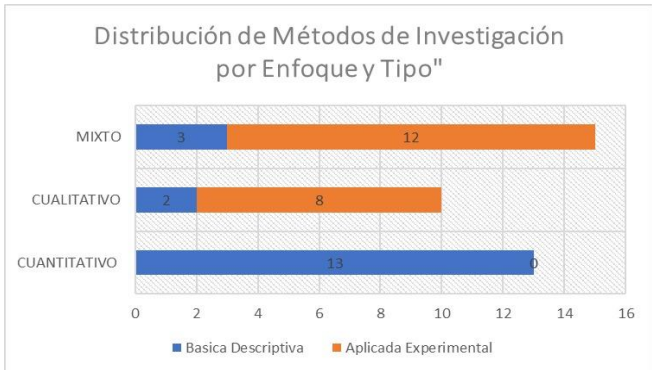


Fig. 3 Distribución de los estudios según enfoque (cuantitativo, cualitativo, mixto) y tipo (básico/aplicado)

La muestra de distribución porcentual de los temas abordados en los 38 estudios analizados. Predominan los trabajos centrados en IA/ML/LLM/NLP aplicados a requisitos con un 33 %, seguidos por aquellos que tratan de calidad/NFR y atributos de calidad con 22 % y métricas e indicadores para cuantificar la calidad o el proceso de RE con 18 %.

En menor proporción aparecen estudios sobre contextos ágiles con un 10 %), explicabilidad como requisito de calidad

12 %, así como trabajos sobre defectos y riesgo, dominios específicos (salud, automoción, e-commerce, África) y frameworks o modelos de procesos de RE, cada uno con porcentajes individuales inferiores pero complementarios al panorama general.

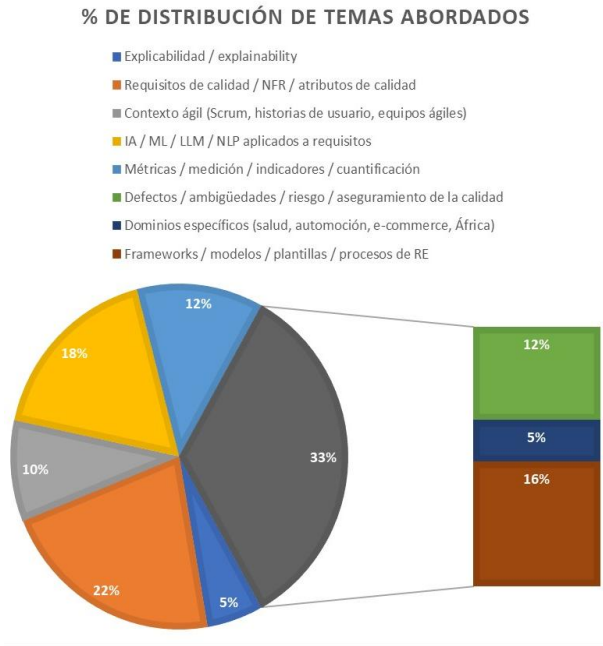


Fig. 4 Distribución de temas abordados

Los artículos proceden principalmente de revistas y conferencias indexadas en Scopus, complementadas por trabajos encontrados en Web of Science y EBSCO, todos relacionados con ingeniería de requisitos, calidad del software y medición de atributos de calidad [7], [11].

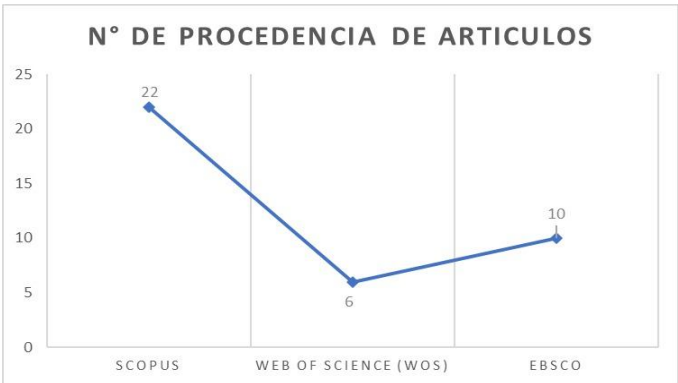


Fig. 5 Procedencia de los artículos

DATOS DEL CONTENIDO

A. ¿Cómo se ha definido y caracterizado la especificación/gestión de requisitos?

En los 38 estudios analizados, la especificación y gestión de requisitos se caracteriza principalmente como:

- Documentación estructurada (Sistema Suplementario de Retención (SRS), historias de usuario, plantillas, tablas de decisión, modelos de seguridad y privacidad) [9], [11], [28], [34], [36], [38], [39].
- Procesos y marcos de ingeniería de requisitos (RE) que integran fases, roles, artefactos y prácticas de aseguramiento de calidad [1], [15], [17], [18], [28], [35].
- Gestión explícita de requisitos de calidad (Requerimientos No Funcionales NFR/QRs) — incluyendo seguridad, privacidad, transparencia, explicabilidad y calidad de datos— como parte formal de la especificación [14], [17], [20], [23], [24], [25], [27], [30], [38].

En el conjunto de trabajos [7]–[38], la especificación se concibe como un conjunto de artefactos textuales o semi-formales que describen tanto funcionalidad como atributos de calidad. Varios estudios destacan el uso de SRS en lenguaje natural estructurado, historias de usuario con criterios de aceptación y plantillas sistemáticas para capturar requisitos funcionales y no funcionales [9], [11], [34], [36], [38], [39].

Otros trabajos enfatizan la formalización progresiva de requisitos mediante metamodelos, ontologías, tablas de decisión y lenguajes específicos de dominio, con el propósito de reducir ambigüedad, mejorar la trazabilidad y habilitar análisis automáticos [14], [17], [34], [35], [38], [39].

La gestión de requisitos se describe como un proceso continuo que abarca elicitación, análisis, documentación, validación, gestión de cambios y control de la volatilidad, articulado con modelos de madurez, marcos de seguridad, privacidad y explicabilidad [16], [19], [22], [25], [26], [27], [28], [30], [33], [35], [41].

En estudios sobre deuda técnica, madurez de requisitos de seguridad y transparencia, la calidad de la especificación se interpreta como un activo medible y gestionable a lo largo del ciclo de vida de desarrollo [19], [28], [35], [2].

En consecuencia, se resalta la gestión de requisitos integra de forma explícita los requisitos de calidad/no funcionales (seguridad, privacidad, transparencia, explicabilidad, calidad de datos), apoyados en marcos, catálogos o lenguajes específicos, lo que permite tratar estos atributos como parte central de la definición y control de la calidad del sistema desarrollado [16], [19], [22], [25], [26], [27], [28], [30], [33], [41].

B. ¿En qué contextos/proyectos se estudia la relación RE ↔ calidad?

La relación entre especificación/gestión de requisitos y calidad del software se aborda en una variedad de contextos:

- Organizaciones de desarrollo de software generales y PYMEs [1], [10], [29], [36], [2].

- Entornos ágiles y DevOps (Scrum, historias de usuario, integración continua) en aplicaciones web y móviles, incluyendo e-commerce y sistemas empresariales [9]–[11], [15], [18], [29], [37], [38].
- Dominios críticos como automoción (sistemas de percepción), salud, sistemas de información sanitaria y privacidad [22], [38], [31], [33], [41].
- Contextos regionales específicos, como el caso de empresas de África subsahariana, donde se analiza el impacto de problemas de RE en la calidad del software [4].
- Escenarios de adopción de IA/ML/LLMs para apoyar tareas de RE (clasificación, aseguramiento de la calidad, evaluación de riesgo y gestión de cambios) [7], [12]–[14], [20], [21], [23], [24], [32], [37], [40].

Distribución de contextos (n = 38)

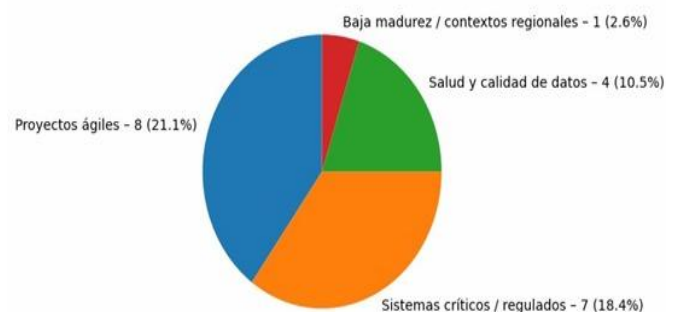


Fig. 6 Distribución de contextos de los artículos Incluidos

En conjunto, los estudios revisados muestran que la relación entre especificación/gestión de requisitos y calidad del software se ha analizado en contextos organizacionales y de dominio muy diversos. Pacheco, García, Calvo-Manzano y Reyes estudian una pequeña organización de desarrollo de software y evidencian cómo la formalización de la obtención de requisitos, basada en ISO/IEC/IEEE 15939, impacta en la calidad del proceso y de los productos generados [2],[7].

Osama, Abdelsalam y Khedr analizan empresas de África subsahariana y concluyen que los problemas de ingeniería de requisitos son un factor crítico en la baja calidad de los sistemas y en el fracaso de los proyectos [4],[1].

De forma complementaria, Ofem, Isong y Lugayizi abordan proyectos de ingeniería de software en general, mostrando que la transparencia en los artefactos de requisitos (SRS) influye en la mantenibilidad y en la productividad de los involucrados [1],[8].

Otros trabajos se sitúan en dominios específicos como el desarrollo ágil de sistemas empresariales, automoción y salud

—por ejemplo, estudios sobre documentación de requisitos de calidad en entornos ágiles, sobre sistemas de percepción automatizada y sobre el lenguaje HDQL para traducir requisitos sanitarios en atributos de calidad— y confirman que el contexto (tipo de industria, criticidad del sistema y enfoque de desarrollo) condiciona la forma en que la especificación de requisitos se vincula con los distintos atributos de calidad del software [9]–[16].

C. ¿Qué técnicas/estándares/herramientas se usan para mejorar la especificación?

Los artículos identifican el uso combinado de plantillas de documentación, estándares de calidad, modelos de madurez, técnicas de revisión sistemática de requisitos y herramientas basadas en IA/NLP para mejorar la especificación y gestión de requisitos [7], [8], [13], [18], [20].

Como soporte documental, se emplean plantillas y guías de SRS para asegurar uniformidad y cubrir requisitos funcionales y no funcionales [1], [14]. Los estándares ISO/IEC 25010 e ISO/IEC/IEEE 15939 estructuran atributos de calidad del producto y procesos de medición, mientras que modelos como CMMI e ISO 9001 orientan la evaluación de la madurez y gestión de requisitos [7], [1], [9], [10].

A nivel procedimental, se proponen inspecciones, walkthroughs, prototipado y análisis de riesgo de requisitos, junto con trazabilidad, para identificar problemas en etapas tempranas [7], [9], [12], [17]. En el plano tecnológico, Los estudios aplican métodos de ML, NLP y LLMs para clasificar requisitos, detectar ambigüedades, identificar requisitos no funcionales y evaluar historias de usuario o cambios de requisitos [18], [19], [20], [24], [25], [26], [4].

En cuanto la distribución del número de artículos según la cantidad de métricas, atributos y estándares considerados en cada estudio. En el eje horizontal se representa el número de métricas/atributos/estándares (2 a 8) y en el eje vertical el número de artículos que utilizan cada cantidad. Se observa que la mayor concentración de trabajos se da en configuraciones de 5 y 7 elementos (7 artículos cada una), mientras que los extremos (2, 4 y 8 elementos) aparecen con menor frecuencia, lo que evidencia una preferencia por esquemas de evaluación de complejidad media en los estudios revisados, a partir del análisis conjunto de los 38 artículos incluidos en la RSL [7]–[41].



Fig. 7 Relación entre la cantidad de métricas, atributos y estándares utilizados

Se plasman plantillas estructuradas y estándares que reportan disminuciones en omisiones de requisitos y mejoras en la consistencia interna de la documentación, lo que reduce errores de interpretación y facilita el seguimiento de cambios [7], [1], [9]. Las experiencias con inspecciones y walkthroughs muestran que una parte relevante de los defectos de requisitos puede detectarse antes del diseño, evitando que se conviertan en fallos costosos en etapas posteriores [17], [9], [12].

En conjunto, estos resultados apoyan la conveniencia de incorporar técnicas estructuradas y automatizadas para lograr especificaciones más robustas y eficientes.

TABLA V
DESCRIPCION DE FAMILIA DE TECNICAS/ ESTANDARES/HERRAMIENTAS

Familia (técnicas/estándares/herramientas)	Elementos	Nº de artículos	% de presencia
IA / NLP / LLMs	Clasificación de requisitos, detección de ambigüedades, extracción de NFR, evaluación de historias de usuario, QA con prompts	11	28.9%
Estándares y modelos de calidad/madurez	ISO/IEC 25010, ISO/IEC/IEEE 15939, CMMI, ISO 9001, modelo de madurez para RE seguro	9	23.7%
Prácticas de revisión temprana	Inspecciones formales, walkthroughs, pair/peer documentation, prototipado	6	15.8%
Documentación estructurada	SRS/plantillas, guías de documentación, tablas de decisión, catálogos de calidad	6	15.8%
Trazabilidad, gestión de cambios y métricas	Matrices de trazabilidad, control de cambios, tableros y métricas operativas	4	10.5%
Alineación requisitos–pruebas	Metamodelos y patrones para vincular requisitos con casos de prueba y criterios de aceptación	2	5.3%

Se plasman plantillas estructuradas y estándares que reportan disminuciones en omisiones de requisitos y mejoras

en la consistencia interna de la documentación, lo que reduce errores de interpretación y facilita el seguimiento de cambios [7], [1], [9]. Las experiencias con inspecciones y walkthroughs muestran que una parte relevante de los defectos de requisitos puede detectarse antes del diseño, evitando que se conviertan en fallos costosos en etapas posteriores [17], [9], [12].

En conjunto, estos resultados apoyan la conveniencia de incorporar técnicas estructuradas y automatizadas para lograr especificaciones más robustas y eficientes.

D. ¿Qué indicadores varían con la calidad de la especificación y cuál es el tamaño del efecto reportado? (defectos post-entrega, retrabajo, fiabilidad, mantenibilidad, usabilidad, eficiencia, seguridad/explicabilidad, satisfacción del usuario, tiempo/costo)

Se utiliza un conjunto amplio de indicadores para evaluar la relación entre calidad de la especificación y calidad del software, que se pueden agrupar en:

- Indicadores de defectos y retrabajo: número de defectos por fase, defectos post-entrega, retrabajo, fallas de entrega [1], [10], [4], [33], [2].
- Indicadores de calidad de requisitos: cantidad de requisitos identificados, completitud, ambigüedad, consistencia, cobertura de NFR, madurez de prácticas, deuda técnica de requisitos, criterios ISO/IEC/IEEE 29148 [17], [19], [28], [29], [34], [35], [36], [38], [39].
- Indicadores de calidad del producto: atributos como fiabilidad, mantenibilidad, usabilidad, eficiencia, seguridad, privacidad, explicabilidad, calidad de datos y satisfacción del usuario [16], [19], [22], [25], [26], [27], [28], [30], [31], [33], [41].
- Indicadores de rendimiento de modelos y herramientas: exactitud, F1-score, precisión, recall, capacidad de detección de defectos, acuerdos con expertos [7], [12]–[14], [20], [21], [23], [24], [32], [37], [40].

Algunos trabajos emplean métricas de defectos y retrabajo para evidenciar que problemas en la fase de requisitos repercuten en la calidad final: análisis de histogramas de incidencias muestran que una fracción significativa de los defectos se origina en RE [4], mientras que la adopción de procesos sistemáticos de elicitación y medición se asocia con disminuciones de defectos y de esfuerzo de retrabajo [10], [2].

En cuanto a la calidad de la especificación, se aplican métricas como número de requisitos elicitados, porcentaje de requisitos ambiguos, índices de completitud, niveles de madurez o valores de deuda técnica, así como chequeos de calidad según ISO/IEC/IEEE 29148 [17], [19], [28], [29], [30], [35], [36], [38], [39]. Estas métricas se utilizan tanto en estudios de mejora de procesos (pair documentation, guías de

QRs, modelos de madurez) como en propuestas de automatización (LLMs para evaluación de calidad).

Los atributos de calidad del producto sirven como marco conceptual para derivar indicadores: modelos de calidad que relacionan requisitos con fiabilidad, mantenibilidad, eficiencia, seguridad y explicabilidad [16], [19], [25], [26], [27], [30], [33]; lenguajes y marcos específicos para calidad de datos de salud y seguridad en automoción [22], [28], [31], [41]. Aunque muchos trabajos se centran en la definición de modelos, varios destacan explícitamente la necesidad de conectar estos atributos con métricas observables en la práctica.

Por último, los estudios que usan IA/ML/LLMs miden su impacto a través de métricas de rendimiento como exactitud y F1-score en tareas de clasificación de requisitos, detección de ambigüedades, identificación de defectos y evaluación de riesgo, así como mediante estudios de usuario con ingenieros de requisitos que valoran la utilidad de las herramientas [7], [12]–[14], [20], [21], [23], [24], [32], [37], [40].

En general, reportan mejoras claras en dichos indicadores cuando se introducen prácticas más sistemáticas o formales de especificación (por ejemplo, procesos de medición, plantillas estructuradas, modelos de madurez, guías de requisitos de calidad y técnicas automáticas de análisis), expresadas como reducciones de defectos y retrabajo, incrementos en la calidad de los requisitos o aumentos significativos en el desempeño de los modelos, aun cuando la mayoría no informa tamaños de efecto estadísticos estandarizados y presenta los resultados en términos comparativos frente a prácticas previas o enfoques base [10], [19], [4], [28], [29], [35]–[37], [2], [40], [41].

TABLA VI
DESCRIPCION DE CATEGORÍA DE INDICADORES

Categoría	Indicadores medidos	Tipo de medición	Efecto reportado en los estudios
Defectos y retrabajo	Defectos por fase, defectos post-entrega, retrabajo, fallas de entrega	Métricas de proceso histórico y registros de incidencias	Disminución de defectos y esfuerzo de retrabajo al usar prácticas formales
Calidad de requisitos	Completitud, ambigüedad, consistencia, cobertura NFR, deuda técnica, madurez	Evaluación de artefactos y estándares (ISO/IEC/IEEE 29148)	Mejora de claridad y calidad de especificaciones
Calidad del producto	Fiabilidad, mantenibilidad, usabilidad, eficiencia, seguridad, explicabilidad, satisfacción	Modelos de calidad derivados de requisitos	Mejor calidad percibida y mayor estabilidad del sistema
Rendimiento de herramientas	Accuracy, F1-score, precisión, recall, detección de defectos	Evaluación experimental de IA/ML/LLMs	Mejora significativa en identificación de errores y riesgos

IV. DISCUSIÓN

En esta revisión, la evidencia recopilada indica que la calidad del software no depende únicamente de las actividades de prueba, sino que está fuertemente condicionada por la forma en que se especifican y gestionan los requisitos desde las primeras etapas del proyecto. Estudios empíricos muestran que el uso de plantillas estructuradas, trazabilidad explícita y métricas de proceso en la obtención de requisitos reduce defectos post-entrega y retrabajo, mejorando los resultados globales del proyecto en organizaciones de software de pequeño tamaño [1], [10], [16], [18].

De manera complementaria, otros trabajos evidencian que la incorporación de métricas orientadas a la transparencia en la especificación de requisitos facilita el seguimiento, la evaluación de calidad y la toma de decisiones sobre los artefactos de requisitos [7], [8], [20]. En contextos de menor madurez se ha documentado que los problemas de ingeniería de requisitos constituyen el principal factor asociado a la baja calidad del software y al fracaso de proyectos en organizaciones de África subsahariana [12], [19], [22].

Además, el análisis de esta RSL muestra que más del 60 % de los estudios utilizan cinco o más métricas, atributos o estándares para evaluar la relación entre requisitos y calidad (por ejemplo, combinaciones de ISO/IEC 25010, modelos de madurez, conteo de defectos, esfuerzo de retrabajo y satisfacción del usuario), lo que permite caracterizar con mayor precisión el efecto de la especificación de requisitos sobre la fiabilidad, mantenibilidad y desempeño de los sistemas [7], [8], [12], [14], [20], [15].

En conjunto, estos trabajos sugieren que no solo es importante contar con métricas, sino disponer de un conjunto suficiente y coherente de indicadores que conecte explícitamente las propiedades de los requisitos con resultados observables en el producto y en el proyecto [7]–[8], [12], [14], [20], [25].

En contraste con revisiones centradas solo en requisitos funcionales, los estudios analizados subrayan el papel de los requisitos de calidad y de propiedades no funcionales como explicabilidad, seguridad, rendimiento o confiabilidad. Algunos trabajos tratan la explicabilidad como un requisito no funcional específico y discuten sus implicaciones para la especificación y evaluación de sistemas explicables, proponiendo recomendaciones para integrar esta propiedad en la ingeniería de requisitos [9], [20]. Otros estudios proponen modelos, catálogos y guías de documentación que vinculan directamente estos requisitos de calidad con métricas de evaluación y criterios de aceptación en diferentes dominios [13], [23], [27], [29].

Asimismo, la incorporación de técnicas automatizadas basadas en NLP, *ensemble learning* y LLMs para apoyar la clasificación, verificación y priorización de requisitos ha mostrado mejoras en la detección temprana de ambigüedades y defectos, en la cobertura de atributos de calidad y en la gestión de riesgos en dominios específicos como salud, automoción, comercio electrónico y contextos africanos [11],

[13], [17], [19], [24], [4], [28], [30], [32]. En esta línea, se reporta que la clasificación automática de requisitos mediante técnicas de *machine learning* contribuye a mejorar la organización y comprensión de los requisitos, facilitando su análisis posterior y reduciendo inconsistencias en la documentación [14], [4].

En conjunto, los resultados de los 38 artículos revisados refuerzan que fortalecer la ingeniería de requisitos —mediante procesos más formales, un uso sistemático de métricas y atributos de calidad, y el apoyo de herramientas inteligentes— constituye una estrategia efectiva para gestionar riesgos, optimizar recursos y elevar de forma sostenible la calidad global de los sistemas informáticos desarrollados [7]–[8], [11]–[15], [18]–[22], [25]–[30], [32].

V. CONCLUSIÓN

En esta revisión sistemática se confirma que la calidad de los sistemas informáticos está profundamente condicionada por la forma en que se especifican y gestionan los requisitos, más allá de las actividades de prueba o de la tecnología utilizada. La evidencia de los 38 estudios analizados muestra que las organizaciones que aplican una ingeniería de requisitos explícita —con plantillas estructuradas, trazabilidad, revisiones formales y gestión sistemática de cambios— reportan menores niveles de defectos post-entrega, menos retrabajo y una mayor probabilidad de éxito del proyecto en comparación con aquellas que emplean prácticas informales o mínimas.

Un hallazgo relevante de esta RSL es el peso específico de los requisitos no funcionales. Cuando atributos como fiabilidad, rendimiento, seguridad, usabilidad o explicabilidad se modelan desde el inicio y se vinculan a la arquitectura y al diseño, los sistemas resultan más estables y mantenibles; por el contrario, cuando estos requisitos se dejan implícitos o se incorporan tarde, las correcciones suelen ser costosas y, en algunos casos, estructuralmente difíciles de revertir. Esto refuerza la necesidad de tratar los requisitos de calidad como ciudadanos de primer orden dentro de la ingeniería de requisitos, y no como anexos secundarios.

En términos de medición, el análisis de los artículos evidencia un patrón: alrededor del 60 % de los estudios emplea cinco o más métricas, atributos o estándares para evaluar la relación entre requisitos y calidad, combinando indicadores de producto (defectos, fiabilidad, mantenibilidad), de proceso (retrabajo, esfuerzo, tiempos de entrega) y de experiencia del usuario (satisfacción, aceptación). Los trabajos que utilizan solo dos o tres indicadores tienden a ofrecer visiones parciales y tienen menos capacidad para explicar el impacto real de la especificación de requisitos sobre el desempeño del sistema.

A partir de ello, esta revisión sugiere que un conjunto mínimo razonable para proyectos de desarrollo debería incluir al menos cinco indicadores complementarios, alineados con

modelos como ISO/IEC 25010 y con las necesidades del contexto de desarrollo.

Finalmente, los resultados muestran que la combinación de procesos formales de ingeniería de requisitos con técnicas de apoyo basadas en IA —como NLP, modelos de ensemble o LLMs para clasificación, detección de ambigüedades y priorización— potencia la capacidad de los equipos para anticipar riesgos y mejorar la cobertura de requisitos de calidad.

En conjunto, la RSL permite recomendar a los profesionales que: (1) institucionalicen prácticas formales de especificación y gestión de requisitos; (2) integren explícitamente los requisitos no funcionales desde el inicio del ciclo de vida; y (3) definan una estrategia de medición que utilice múltiples métricas coherentes. Bajo estas condiciones, los sistemas desarrollados tienen una probabilidad considerablemente mayor de alcanzar niveles de calidad sostenibles en contextos tanto tradicionales como ágiles

REFERENCIAS

- [1] P. Ofem, B. Isong, and F. Lugayizi, “Métricas para evaluar y mejorar la transparencia en ingeniería de software: un estudio empírico y un modelo de mejora,” 2024. doi: 10.1007/s42979-024-03471-3.
- [2] C. Pacheco, I. Garcia, J. A. Calvo-Manzano, and M. Reyes, “Medición y mejora de la obtención de requisitos de software en una organización de software de tamaño pequeño: una implementación liviana del proceso de medición ISO/IEC/IEEE 15939:2017,” Springer-Verlag London Ltd., 2022. doi: 10.1007/s00766-022-00394-4.
- [3] J. Grzybowski, L. Młyńczak, M. Sokołowska, and M. Matowicki, “Elaboración de requisitos de software con tablas de decisión,” Facultad de Ingeniería de Transporte y Aviación, Universidad Tecnológica de Silesia, 2024. doi: 10.20858/sjsutst.2024.125.6.
- [4] E. Osama, M. Abdelsalam, and A. Khedr, “El efecto de la calidad y la volatilidad de los requisitos en el éxito de los proyectos de sistemas de información,” 2020. doi: 10.14569/IJACSA.2020.0110950
- [5] Q. Wang, J. Liao, M. Lapata, and M. Macleod, “PICO entity extraction for preclinical animal literature,” Systematic Reviews, vol. 11, art. 209, 2022. doi: 10.1186/s13643-022-02009-5.
- [6] M. L. Rethlefsen, S. Kirtley, S. Waffenschmidt, et al., “PRISMA-S: an extension to the PRISMA Statement for Reporting Literature Searches in Systematic Reviews,” Systematic Reviews, 2021. doi: 10.1186/s13643-020-01542-z.
- [7] P. Vasudevan, S. Reddivari, “The Role of Generative AI Models in Requirements Engineering: A Systematic Literature Review,” ACM International Conference Proceeding Series, 2025. doi: 10.1145/3696673.3723053.
- [8] U. K. Durrani, M. Akpinar, H. Bektas, M. Saleh, “Impact of Artificial Intelligence on Software Engineering Phases and Activities (2013-2024): A Quantitative Analysis Using Zero-Truncated Poisson Model,” IEEE Access, 2025. doi: 10.1109/ACCESS.2025.3499125.
- [9] W. Behutiye, P. Seppänen, P. Rodríguez, M. Oivo, “Documentation of Quality Requirements in Agile Software Development,” ACM International Conference Proceeding Series, 2020. doi: 10.1145/3383219.3383245.
- [10] D. García-López, M. Segura-Morales, E. Loza-Aguirre, “Improving the quality and quantity of functional and non-functional requirements obtained during requirements elicitation stage for the development of e-commerce mobile applications: an alternative reference process model,” IET Software, 2020. doi: 10.1049/iet-sen.2018.5443.
- [11] W. Behutiye, M. Oivo, P. Rodríguez, “Quality Requirement Documentation Guidelines for Agile Software Development,” IEEE Access, 2021. doi: 10.1109/ACCESS.2022.3187106.
- [12] L. Giamattei, A. Guerriero, R. Pietrantuono, S. Russo, “Causal Reasoning in Software Quality Assurance: A Systematic Review,” Information and Software Technology, 2025. doi: 10.1016/j.infsof.2024.107599.
- [13] R. Izhar, K. Cosh, L. Ramingwon, S. Ramingwong, S. N. Bhatti, “An Efficient Methodology for the Categorization of Software Requirements Using Natural Language Processing and Similarity Analysis,” IEEE Access, 2025. doi: 10.1109/ACCESS.2025.3568504.
- [14] S. Lubos, A. Felfernig, T. N. T. Tran, D. Garber, M. El Mansi, S. Polat Erdeniz, V.-M. Le, “Leveraging LLMs for the Quality Assurance of Software Requirements,” IEEE International Conference on Requirements Engineering (RE), 2024. doi: 10.1109/RE59067.2024.00046.
- [15] S. Anwer, L. Wen, M. U. Hassan, Z. Wang, A. A. Al-Awady, Y. A. A. Ali, “BERDD: A Behaviour Engineering-Based Approach for Requirements Defects Detection,” IEEE Access, 2024. doi: 10.1109/ACCESS.2024.3365352. 11
- [16] L. Chazette, W. Brunotte, T. Speith, “Exploring Explainability: A Definition, a Model, and a Knowledge Catalogue,” IEEE International Conference on Requirements Engineering (RE), 2021. doi: 10.1109/RE51729.2021.00025.
- [17] E. Serna, A. Serna, “Process and Progress of Requirement Formalization in Software Engineering,” Ingeniare. Revista chilena de ingeniería, 2020. doi: 10.4067/S0718-33052020000300411.
- [18] Y. Fitrissia, R. Ibrahim, “A Framework for Capturing Quality Requirements by Integrating the Requirement Engineering Elements in Agile Software Development Methods,” International Journal of Advanced Computer Science and Applications (IJACSA), 2024. doi: 10.14569/IJACSA.2024.0150920.
- [19] D. Gobov, O. Zuieva, “Software Quality Attributes in Requirements Engineering,” International Journal of Information Technology and Computer Science (IJITCS), 2025. doi: 10.5815/ijitcs.2025.04.04.
- [20] F. Talha, T. Tahir, T. Nadeem, “A Semiautomated Approach for Detecting Ambiguities in Software Requirements Using SpanBERT and Named Entity Recognition,” Software: Practice and Experience, 2025. doi: 10.1002/smr.70041.
- [21] M. Binkhonain, R. Alfayez, “Are prompts all you need? Evaluating prompt-based Large Language Models (LLMs) for software requirements classification,” IEEE Access, 2025. doi: 10.1109/ACCESS.2025.3571871.
- [22] T. Haddad, P. Kumarapeli, S. de Lusignan, S. Barman, S. Khaddaj, “Healthcare Data Quality Language (HDQL): Streamlining the Translation of Healthcare System Requirements into Software Quality Attributes,” Studies in Health Technology and Informatics, 2025. doi: 10.3233/SHTI250456.
- [23] C. Kumar, P. S. Khan, M. Srinivas, S. Prakash, R. S. Rathore, “Ensemble Learning for Software Requirement-Risk Assessment: A Comparative Study of Bagging and Boosting Approaches,” Journal of Intelligent Manufacturing, 2025. doi: 10.1007/s10844-025-00939-3.
- [24] Y. Kong, et al., “Collaboration with Generative AI to Improve Requirements Change,” Computer Standards & Interfaces, 2025. doi: 10.1016/j.csi.2025.104013.
- [25] J. Droste, H. Deters, M. Obaidi, J. Klünder, K. Schneider, “Framing what can be explained – an operational taxonomy for explainability needs,” Requirements Engineering, 2025. doi: 10.1007/s00766-025-00440-x.
- [26] R. Ramasamy, T. Y. Lim, “Exploring Non-Functional Requirements in Requirements Engineering: A Systematic Review,” IEEE International Conference on Computational Intelligence (ICCI), 2024. doi: 10.1109/ICOCO62848.2024.10928262.
- [27] H. Alqaraawi, S. von Keitz, B. Kroll, A. Vogelsang, K. Geihs, “Explainability as a non-functional requirement: challenges and recommendations,” Requirements Engineering, 2020. doi: 10.1007/s00766-020-00333-1.
- [28] A. M. Saeed, M. Niazi, M. Alshayeb, S. Mahmood, S. Zafar, “A maturity model for secure requirements engineering,” Computers & Security, 2020. doi: 10.1016/j.cose.2020.101852.
- [29] N. Qamar, N. Sabahat, A. Mosavi, “Evaluating the Impact of Pair Documentation on Requirements Quality in Agile Software Development,” IEEE Access, 2025. doi: 10.1109/ACCESS.2025.3550123.

- [30] L. Chazette, W. Brunotte, T. Speith, "Explainable software systems: from requirements analysis to system evaluation," *Requirements Engineering*, 2022. doi: 10.1007/s00766-022-00393-5.
- [31] K. M. Habibullah, H. Heyn, G. Gay, J. Horkoff, E. Knauss, M. Borg, A. Knauss, H. Sivencrona, P. J. Li, "Requirements and software engineering for automotive perception systems: an interview study," *Requirements Engineering*, 2024. doi: 10.1007/s00766-023-00410-1.
- [32] A. Alhaizaey, M. Al-Mashari, "Automated Classification and Identification of Non-Functional Requirements in Agile-Based Requirements Using Pre-Trained Language Models," *IEEE Access*, 2025. doi: 10.1109/ACCESS.2025.3570359.
- [33] T. Hovorushchenko, D. Medzaty, Y. Voichur, M. Lebiga, "Method for forecasting the level of software quality based on quality attributes," *Journal of Intelligent and Fuzzy Systems*, 2023. doi: 10.3233/JIFS-222394.
- [34] K. Großer, A. S. Ahmadian, M. Rukavitsyna, Q. Ramadan, J. Jürjens, "Benchmarking requirement template systems: comparing appropriateness, usability, and expressiveness," *Requirements Engineering*, 2024. doi: 10.1007/s00766-024-00427-0.
- [35] J. Perera, E. Tempero, Y.-C. Tu, K. Blincoe, "Modelling the quantification of requirements technical debt," *Requirements Engineering*, 2024. doi: 10.1007/s00766-024-00424-3.
- [36] X. Franch, C. Palomares, C. Quer, P. Chatzipetrou, T. Gorschek, "The state-of-practice in requirements specification: an extended interview study at 12 companies," *Requirements Engineering*, 2023. doi: 10.1007/s00766-023-00399-7.
- [37] G. Airlangga, "Enhancing Software Requirements Classification with Semisupervised GAN-BERT Technique," *Journal of Electrical and Computer Engineering*, 2024. doi: 10.1155/2024/4955691.
- [38] T. N. Kudo, R. de F. Bulcão-Neto, V. V. Graciano Neto, A. M. R. Vincenzi, "Aligning requirements and testing through metamodeling and patterns: design and evaluation," *Requirements Engineering*, 2023. doi: 10.1007/s00766-022-00377-5.
- [39] J. Grzybowski, L. Młyńczak, M. Sokołowska, M. Matowicki, "Software Requirements Elaboration with Decision Tables," *Requirements Engineering*, 2024. doi: 10.1007/s00766-024-00438-x.
- [40] M. A. Umar, K. Lano, "Advances in automated support for requirements engineering: a systematic literature review," *Software Quality Journal*, 2024. doi: 10.1007/s10664-024-10515-3.
- [41] M. Peixoto, C. Silva, J. Araújo, T. Gorschek, A. Vasconcelos, J. Vilela, "Evaluating a privacy requirements specification method by using a mixed-method approach: results and lessons learned," *Requirements Engineering*, 2022. doi: 10.1007/s00766-022-00388-2