

Modelos de Elicitación de Requisitos Basados en Metodologías Ágiles: Una Revisión Sistemática de la Literatura

Anderson Ruiz Rojas¹

joseanderson.ruiz@unmsm.edu.pe

<https://orcid.org/0000-0002-8882-7816>

Universidad Nacional Mayor de San Marcos
Perú

Yudi Guzmán Monteza

yudi.guzman@unmsm.edu.pe

<https://orcid.org/0000-0001-5306-5295>

Universidad Nacional Mayor de San Marcos
Perú

RESUMEN

En este estudio se realiza una Revisión Sistemática de la Literatura (RSL) centrada en los modelos de elicitación de requisitos basados en metodologías ágiles. El objetivo es identificar patrones, prácticas exitosas y desafíos asociados a la elicitación de requisitos en entornos ágiles. La identificación y comprensión precisa de los requerimientos del sistema resulta crucial en el proceso de desarrollo de software, y en este panorama, la Elicitación de Requisitos (ER) desempeña un papel fundamental en este proceso. Las Metodologías Ágiles (MA), como Scrum, se han vuelto populares debido a su enfoque en la colaboración, entrega temprana y adaptabilidad a los cambios, ofreciendo ventajas significativas, como retroalimentación continua, la colaboración estrecha con los stakeholders y un enfoque orientado al valor de negocio. A pesar de la importancia de una ER adecuada, la tasa de éxito promedio de los proyectos de software sigue siendo baja. Los hallazgos derivados del análisis de frameworks y modelos para la ER basados en MA revelan un enfoque productivo en el uso de Scrum y procesos similares para gestionar de manera iterativa y colaborativa los requisitos del sistema. La integración de documentos de requisitos conjuntos (JRD) con Scrum destaca como una práctica común para mejorar la calidad y la colaboración. Además, se observa una adaptabilidad significativa relacionado a proyectos aplicando Big Data. La diversidad de modelos, desde Requirements Engineering for Software Architecture (RE4SA) hasta enfoques semiautomáticos para Requisitos de Calidad, destaca la variedad de estrategias para mejorar la eficacia de la ER en contextos ágiles. La validación práctica en entornos reales refuerza la aplicabilidad de estos enfoques. Estos hallazgos ofrecen percepciones valiosas para la mejora continua de la ER en el desarrollo ágil de software.

Palabras clave: elicitación de requisitos; ingeniería de requisitos; obtención ágil de requisitos; metodologías ágiles

¹ Autor principal

Correspondencia: joseanderson.ruiz@unmsm.edu.pe

Requirements Elicitation Models Based on Agile Methodologies: A Systematic Review of the Literature

ABSTRACT

In this study, a Systematic Review of the Literature (RSL) is carried out focusing on requirements elicitation models based on agile methodologies. The objective is to identify patterns, successful practices and challenges associated with the elicitation of requirements in agile environments. Accurately identifying and understanding system requirements is crucial in the software development process, and in this scenario, Requirements Selection (ER) plays a critical role in this process. Agile methodologies (MA), such as Scrum, have become popular due to their focus on collaboration, early delivery, and adaptability to change, offering significant advantages, such as continuous feedback, close collaboration with stakeholders, and a business value-oriented approach. Despite the importance of proper ER, the average success rate of software projects remains low. Findings derived from the analysis of AM-based frameworks and models for RE reveal a productive approach to using Scrum and similar processes to iteratively and collaboratively manage system requirements. Integrating Joint Requirements Documents (JRDs) with Scrum stands out as a common practice for improving quality and collaboration. In addition, there is a significant adaptability related to projects applying Big Data. The diversity of models, from Requirements Engineering for Software Architecture (RE4SA) to semi-automated approaches to Quality Requirements, highlights the variety of strategies to improve the effectiveness of RE in agile contexts. Practical validation in real-world environments reinforces the applicability of these approaches. These findings offer valuable insights for continuous improvement of ER in agile software development.

Keywords: elicitation of requirements; requirements engineering ; agile requirements procurement; agile methodologies

*Artículo recibido 20 noviembre 2023
Aceptado para publicación: 30 diciembre 2023*

INTRODUCCIÓN

En el entorno empresarial actual, donde los requisitos son cambiantes, la correcta identificación y comprensión de los requisitos del sistema se convierte en un factor crítico para el éxito en el desarrollo de software (Pressman & Maxim, 2019). La ER como proceso fundamental en este ámbito, desempeña un papel crucial al capturar las necesidades y expectativas de los stakeholders, garantizando así la entrega de un producto de calidad.

En los últimos años, las MA han revolucionado la forma en que se aborda el desarrollo de software, promoviendo la colaboración, la entrega temprana y continua de software funcional y la capacidad de adaptarse a los cambios (Marques & Da Cunha, 2019). Scrum, una de las metodologías ágiles más conocidas, destaca por su enfoque en ciclos iterativos, conocidos como Sprints, y una fuerte interacción con los clientes para obtener un feedback constante. Por otro lado, la Programación Extrema o Extreme Programming (XP) tiene como base cinco valores fundamentales: Simplicidad, Comunicación, Retroalimentación, Respeto y Coraje (Cadavid et al., 2013). XP facilita lanzamientos frecuentes en ciclos de desarrollo cortos, mejorando la productividad del equipo, y permitiendo la adopción ágil de nuevos requisitos del cliente al final de cada ciclo (Patilla Hubner et al., 2021).

Carrasco Gonzaga et al. (2019) señalan que en la ingeniería de software abundan diversas metodologías, modelos y herramientas, entre las cuales muchas, aunque en su momento contaron con una amplia aceptación y respaldo, han quedado obsoletas. Progresivamente, estas antiguas metodologías han sido sustituidas por enfoques más modernos y adaptables a las dinámicas actuales del desarrollo de software. Metodologías que poseen orientaciones y propósitos distintos; por ejemplo, Scrum se centra en los aspectos administrativos de un proyecto de software, mientras que XP va enfocado a manejar los aspectos técnicos (Salazar et al., 2018).

Tabla 1

	Scrum	XP
	Coraje	Coraje
	Respeto	Respeto
Valores	Compromiso	
	Focus	
	Franqueza	

		Simplicidad
		Realimentación
		Comunicación
Roles	Product Owner	Cliente
	Scrum Master	Encargado de seguimiento
	Scrum Team	Programador
		Encargado de pruebas
Actividades y procesos	Product Backlog	Planeación
	Sprint Planning	Análisis
		Diseño
	Sprint	Codificación
		Pruebas unitarias
		Pruebas de aceptación
		Pruebas de integración
		Scrum diario
		Sprint Review Meeting
	Retrospectiva	
	Release Planning Meeting	
	Product Backlog	Historias de usuario
	Sprint Backlog	Tarjetas CRC
Artefactos		Task Cards
	Incremento del producto	
	Burn down chart	
	Burn up chart	
	Desarrollo iterativo	Casos de pruebas de aceptación
		Entregas pequeñas
		Pruebas
		Diseño simple
		Programación en parejas
		Desarrollo dirigido por pruebas
Prácticas y principios	Integración continua	Integración continua
		Propiedad colectiva del código
		Estándares de codificación
		Metáfora del sistema
		Ritmo sostenible
		Juego de planificación

Refactoring
Cliente en sitio
Paso sostenible

Control empírico de procesos
Colaboración
Bloque de tiempo asignado
Autoorganización
Priorización basada en el valor

Cuadro comparativo entre Scrum y XP

La Tabla 1 ofrece un análisis comparativo entre Scrum y XP en términos de actividades, procesos, artefactos prácticas y principios. Adaptado de (Janampa Patilla et al., 2021; Patilla Hubner et al., 2021). La importancia de una ER adecuada se refleja en su capacidad para reducir cambios y correcciones en los requisitos y en la retroalimentación efectiva al cliente acerca de la integridad y validez de los mismos (Durán, 2000; Robertson & Robertson, 1999). Sin embargo, a pesar de su relevancia, la tasa de éxito de los proyectos de ingeniería de software sigue siendo baja debido a diversas deficiencias técnicas y problemas sociales (Saeeda et al., 2020) s. Las estadísticas revelan una tasa de éxito promedio de alrededor de 42%, con la observación adicional de que los proyectos más pequeños tienen una probabilidad de éxito mucho mayor que los proyectos más grandes (Bierwolf, 2016).

Ante la dinámica del entorno empresarial actual y la alta tasa de fallos en sistemas generados al inicio de su ciclo de vida, las metodologías ágiles cobran aún más importancia para adaptarse de manera ágil y controlada a los cambios en los requisitos (Pressman & Maxim, 2019). Estas metodologías, al enfatizar el factor humano y el producto de software, ofrecen beneficios como feedback temprano, colaboración cercana con los stakeholders y enfoque de valor de negocio, lo que ayuda a asegurar la obtención efectiva y precisa de los requisitos del proyecto (Zuppa et al., 2019).

En este contexto, el presente trabajo tiene como objetivo realizar una RSL sobre "Modelos de Elicitación de Requisitos Basados en Metodologías Ágiles". Mediante esta revisión, se examinarán diversos estudios e investigaciones que abordan la ER en entornos ágiles, con el propósito de identificar patrones, prácticas exitosas y desafíos asociados. La comprensión de cómo las metodologías ágiles influyen en la

ER y cómo pueden ser optimizadas para maximizar sus beneficios se vuelve esencial para el éxito en el desarrollo de software.

A lo largo de la investigación, se explorarán los fundamentos teóricos de las metodologías ágiles y su relación con la elicitación de requisitos, revisando estudios previos y destacando conclusiones clave alcanzadas hasta la fecha. Se espera que esta investigación contribuya al avance del conocimiento en este campo y brinde una valiosa referencia para aquellos interesados en mejorar sus prácticas de ER en entornos ágiles, manteniéndose competitivos en la industria del desarrollo de software.

METODOLOGÍA DE INVESTIGACIÓN

La investigación realizada en este trabajo es una RSL y se ha adaptado el procedimiento propuesto por (Kitchenham & Charters, 2007), el cual involucra las siguientes fases:

- **Planificación de la revisión:** En esta fase, se plantean las preguntas de investigación y se define el protocolo de revisión.
- **Realización de la revisión:** En esta fase, se ejecuta el plan y se seleccionan los estudios primarios, de acuerdo con los criterios de inclusión y exclusión establecidos para el estudio.
- **Reporte de la revisión:** En esta fase se muestran las estadísticas y el análisis de los documentos encontrados y seleccionados, los detalles se explican en las siguientes secciones.

Planificación de la revisión

El propósito de esta investigación es abordar las Preguntas de Investigación (PI) establecidas con el fin de analizar las propuestas presentadas por otros investigadores en el ámbito en cuestión. Además, se busca identificar las MA que han sido incorporadas en los modelos propuestos para llevar a cabo la ER, y se pretende determinar cuáles de estas metodologías han sido efectivamente aplicadas en la industria.

Las PI que guiarán este estudio incluyen:

- **PI1:** ¿Qué metodologías ágiles han sido más utilizadas en los modelos propuestos?
- **PI2:** ¿Qué técnicas y artefactos se han utilizado en los frameworks y modelos de elicitación de requisitos basados en metodologías ágiles?
- **PI3:** ¿Qué prácticas y recomendaciones se han identificado en la literatura para mejorar la elicitación de requisitos en metodologías ágiles y asegurar la satisfacción de los stakeholders?

Tabla 2. Cadenas de búsqueda utilizadas en las bibliotecas digitales

Fuente	Cadena de búsqueda
Scopus	TITLE-ABS-KEY ((software) AND (agile) AND ("requirements engineering" OR "requirement engineering" OR "agile requirements elicitation" OR "agile requirements engineering" OR "agile requirements gathering" OR "agile requirements gathering techniques" OR "agile requirements techniques" OR "agile requirements methods" OR "agile requirements approaches" OR "agile requirements practices")) AND (TITLE-ABS-KEY ("model" OR "framework" OR "approach" OR "methodology" OR "technique" OR "process" OR "methods" OR "tools" OR "components" OR "artifacts")) AND (TITLE-ABS-KEY ("requirements elicitation" OR "requirements engineering" OR "requirements gathering" OR "requirements gathering techniques" OR "requirements techniques" OR "requirements methods" OR "requirements approaches" OR "requirements practices")))
Web of Science (WoS)	TS=((software) AND (agile) AND ("requirements engineering" OR "requirement engineering" OR "agile requirements elicitation" OR "agile requirements engineering" OR "agile requirements gathering" OR "agile requirements gathering techniques" OR "agile requirements techniques" OR "agile requirements methods" OR "agile requirements approaches" OR "agile requirements practices")) AND (TS= ("model" OR "framework" OR "approach" OR "methodology" OR "technique" OR "process" OR "methods" OR "tools" OR "components" OR "artifacts")) AND (TS=("requirements elicitation" OR "requirements engineering" OR "requirements gathering" OR "requirements gathering techniques" OR "requirements techniques" OR "requirements methods" OR "requirements approaches" OR "requirements practices")))

Tabla 3. Criterios de selección y exclusión

Criterios de selección	Criterios de exclusión
CS1: Estudios relacionados con propuestas de modelos, métodos y técnicas para la elicitación de requisitos basados en metodologías ágiles.	CE1: Estudios no orientados al ámbito de la investigación.
CS2: Responden directamente a las preguntas de investigación.	CE2: Fuentes de estudios diferentes de revistas indexadas.
	CE3: El idioma de estudio es diferente al inglés.
	CE4: Artículos de estudios que no se encuentren dentro del periodo 2018 al 5 julio de 2023.

CS3: Artículos de ciencias de la computación. **CE5:** Artículos duplicados.

La Tabla 3, contiene los criterios de inclusión (CI) y criterios de exclusión (CE) establecidos para la RSL.

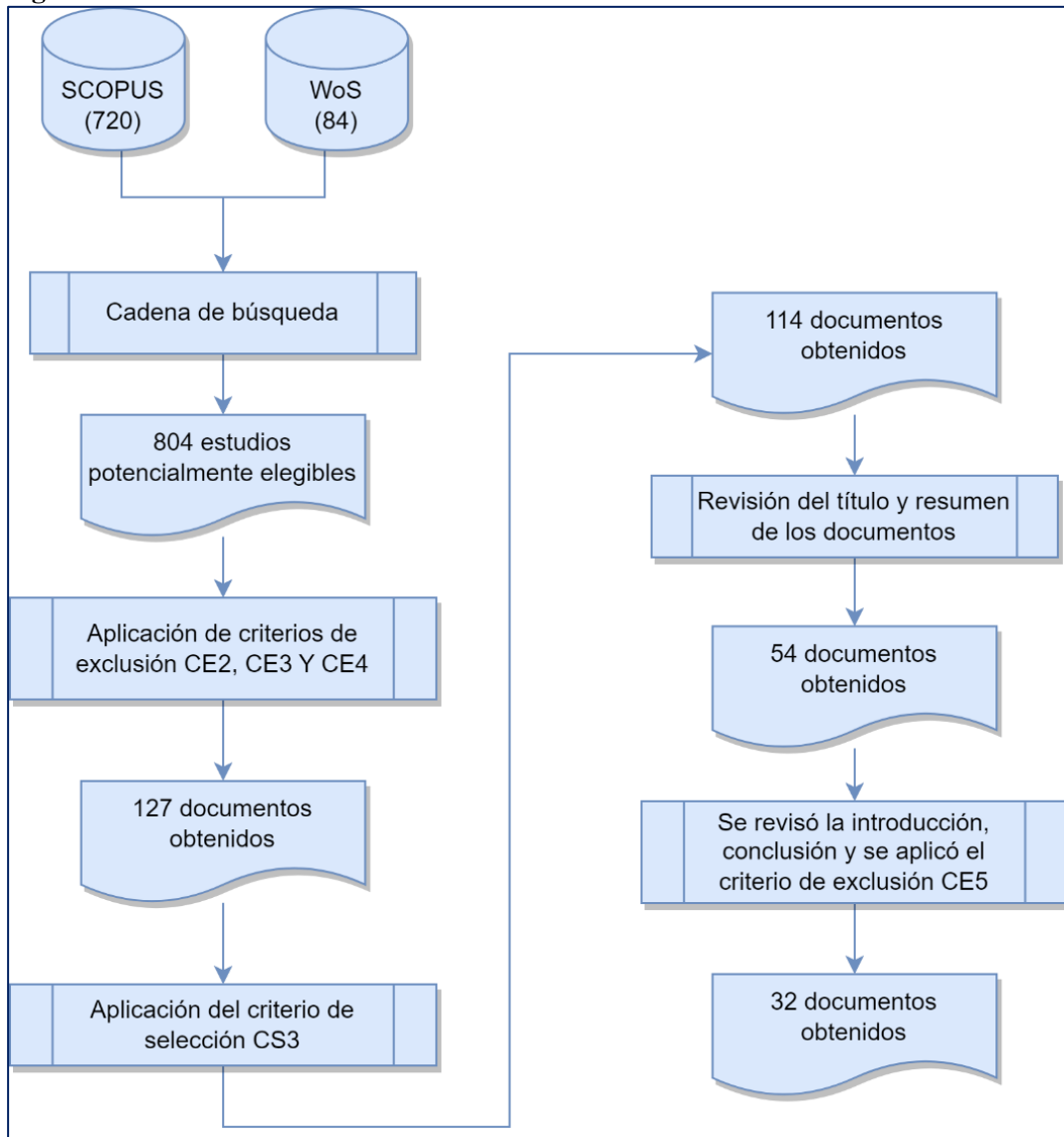
Realización de la revisión

Se llevó a cabo la ejecución de la cadena de búsqueda (ver Tabla 2) en los bancos de datos. En el caso de Scopus, se emplearon los términos de búsqueda TITLE-ABS-KEY para enfocarse en el título, resumen y palabras clave de los documentos. Como resultado, se obtuvo una lista de 720 documentos. Posteriormente, se aplicaron los criterios de exclusión CE2, CE3 y CE4, según se indica en la Tabla 3, lo cual redujo la lista a 70 artículos. Estos artículos fueron sometidos a un proceso de selección basado en el criterio CS3, resultando en 66 documentos para su posterior análisis.

Además, se utilizó la misma metodología de búsqueda empleando los términos TS en Web of Science (WoS), lo que generó una lista de 84 documentos. Después de aplicar los criterios de exclusión CE2, CE3 y CE4, la lista se redujo a 57 artículos. Finalmente, se aplicó el criterio de selección CS3, lo que redujo la lista a 48 documentos para su análisis.

Las listas obtenidas de Scopus y WoS se consolidaron en una única lista de 114 artículos. A partir de esta lista, se llevó a cabo una revisión exhaustiva del título y resumen de cada artículo, lo que permitió reducir la lista a 54 documentos para su análisis, como se muestra en la Tabla 4. Durante este proceso, se eliminaron 17 artículos duplicados, quedando un total de 32 artículos que serán objeto de un análisis riguroso. Este proceso se ilustra gráficamente en la Figura 1.

Figura 1. Proceso de desarrollo de la RSL



Reporte de la revisión

Tendencias de las publicaciones

El proceso de la RSL arrojó un total de 804 posibles estudios, de los cuales se seleccionaron 32 de acuerdo con los criterios de selección y exclusión, como se detalla en la Tabla 3. Estos estudios fueron analizados para abordar las preguntas de investigación planteadas. En la Tabla 4 se presenta la cantidad de estudios seleccionados en las fuentes de datos de Scopus y WoS. Además, la Tabla 5 muestra la distribución de los estudios en diversas revistas científicas indexadas.

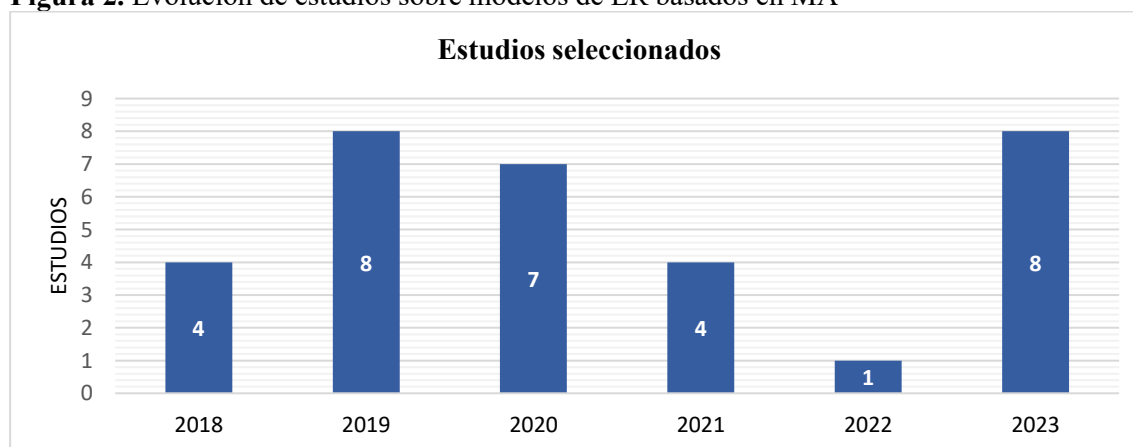
Tabla 4. Estudios potenciales y estudios seleccionados

Fuente	Estudios potenciales	Estudios seleccionados
Scopus	66	31
WoS	48	1
TOTAL	114	32

Tabla 5. Distribución de estudios potenciales y estudios seleccionados por Journal

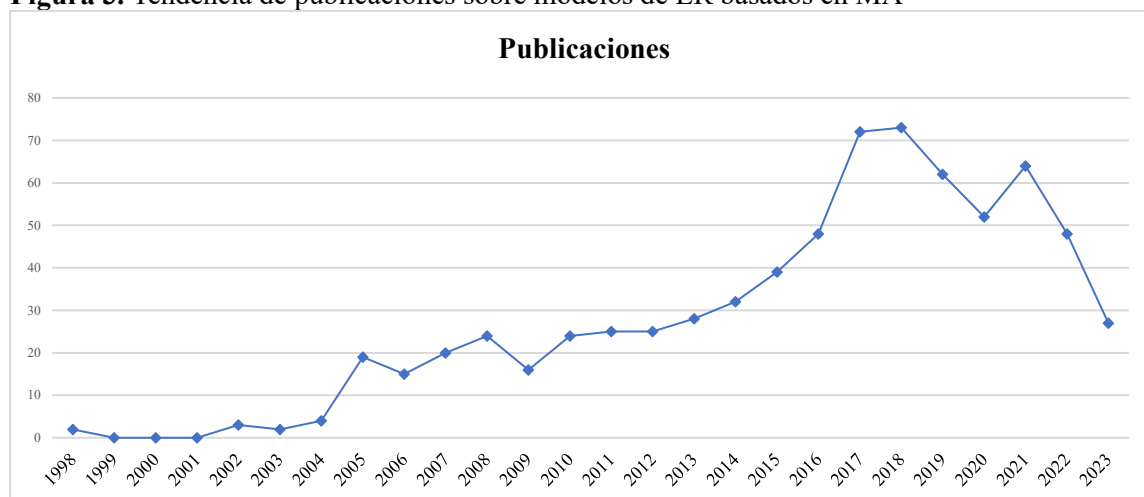
Fuente	Estudios potenciales	Estudios seleccionados
Elsevier	31	9
IEEE	19	6
Springer	25	7
Wiley	2	1
MDPI	6	2
Otros	31	7
TOTAL	114	32

La Figura 2 muestra la cantidad de estudios relacionados con el propósito de estudio de esta RSL, comprendido entre los años 2018 y 2023. Estos 32 estudios representan diversas contribuciones y aportes en el ámbito de la ER basada en MA.

Figura 2. Evolución de estudios sobre modelos de ER basados en MA

Por otro lado, al emplear la sentencia de búsqueda en Scopus (julio 2023), se han identificado las tendencias de las publicaciones relacionadas con modelos de ER basados en MA, con un total de 720 publicaciones (ver Figura 1). Es evidente que las investigaciones en este campo se remontan al menos al año 1998, tal como se muestra en la Figura 3.

Figura 3. Tendencia de publicaciones sobre modelos de ER basados en MA



Hallazgos encontrados durante la RSL

En esta sección se responden a las PI propuestas en el ítem 2.1 Planificación de la revisión.

MA generalmente utilizadas en la ER

Los resultados se presentan en tablas que contienen el aspecto a analizar, su cantidad, fuentes que lo referencian y la frecuencia que este aspecto es referenciado.

Tabla 6. Tipos de aportes en ER basados en MA

Tipo de Aporte	Referencia	Total
Frameworks	Elghariani et al. (2018); Hesham et al. (2021); Saeeda et al. (2020); Sherif et al. (2023); Watanabe et al. (2021)	5
Modelos	Marques & Da Cunha (2019); Medeiros et al. (2018); Melegati et al. (2019); Muhammad et al. (2023); Nasiri et al. (2023); Oriol et al. (2020); Rivero et al. (2019); Spijkman et al. (2021); Tatale & Chandra Prakash (2020)	9
Otros	Alrezaamiri et al. (2019); Casillo et al. (2022); De Vries & Opperman (2023); Dingsøyr et al. (2018); Franch et al. (2023); Hafeez et al. (2021); Hess et al. (2019a); Jia et al. (2019); Kudo et al. (2023); Liebel & Knauss (2023); Medeiros et al. (2020); Mihelič et al. (2023); Noel et al. (2018); Nowroozi et al. (2020); Rojas & Macías (2019); Salleh & Nohuddin (2019); Schon et al. (2020); Younas et al. (2020)	18

El análisis detallado de los frameworks y modelos propuestos para la ER basados en MA, revela una diversidad de enfoques y estrategias empleadas por diferentes investigadores. Estos abordajes buscan

mejorar la eficacia de la ER en entornos ágiles, donde la adaptabilidad y la colaboración son fundamentales. La Tabla 6 proporciona una visión general de estas investigaciones, clasificándolas según los tipos de aportes: Frameworks, Modelos y Otros.

Entre los frameworks analizados, resalta el uso frecuente de Scrum o procesos similares como componentes fundamentales (Hesham et al., 2021; Saeeda et al., 2020). Investigadores como Saeeda et al. (2020) han concebido frameworks que combinan documentos de requisitos conjuntos (Join Requirements Document – JRD) y procesos de Scrum para gestionar de manera efectiva los requisitos del sistema. La validación práctica de este enfoque en una empresa de desarrollo de software en Noruega respalda su eficacia y aplicabilidad en entornos ágiles del mundo real.

Otro enfoque interesante se encuentra en el estudio de Hess et al. (2019), que se centra específicamente en la ER en proyectos de Big Data. Este framework busca mejorar la calidad de los requisitos mediante la integración de técnicas de recolección de requisitos, mapeo mental y la metodología Scrum. La estructura secuencial de cuatro fases proporciona una guía sistemática desde la recolección inicial hasta la ejecución bajo Scrum, destacando la adaptabilidad a contextos específicos.

En cuanto a los modelos propuestos, se observa una diversidad de enfoques que abordan tanto requisitos funcionales como no funcionales en contextos ágiles. Por ejemplo, Spijkman et al. (2021) presenta el modelo RE4SA, que se enfoca en alinear requisitos funcionales y arquitectura funcional. Aunque no aborda aspectos no funcionales, proporciona un marco y métricas para equipos de desarrollo ágil.

En otro enfoque, el estudio de Oriol et al. (2020) introduce un modelo semiautomático respaldado por una herramienta para generar y documentar Requisitos de Calidad. Este enfoque, centrado en artefactos como Alertas de Calidad y Patrones de Requisitos de Calidad, destaca la importancia de la documentación efectiva para garantizar la calidad en proyectos de software.

Marques & Da Cunha (2019) propone el modelo ARES, un enfoque para definir Especificaciones de Software en entornos regulados. Su aplicación exitosa en proyectos académicos y en el desarrollo de un prototipo de Sistema Embebido en Tiempo Real para Aviones sugiere su viabilidad y adaptabilidad en entornos críticos.

Además de estos modelos específicos, se encuentran contribuciones que, si bien no presentan un modelo o framework formal, se basan en técnicas, artefactos, modelos y metodologías ágiles bien establecidos

como Scrum, XP, Lean, User Experience (UX), Design Thinking, entre otros. Estos aportes proporcionan herramientas y procesos que se centran en abordar la ER, respaldados por estudios comparativos de casos reales de la industria. Además, algunos enfoques incorporan el uso de modelos de Machine Learning, añadiendo una capa de innovación tecnológica para mejorar la efectividad del proceso de ER. Estas contribuciones demuestran la amplitud y la sofisticación de los esfuerzos para mejorar la práctica de la ER en el ámbito ágil. (Alrezaamiri et al., 2019; Casillo et al., 2022; De Vries & Opperman, 2023; Dingsøyr et al., 2018; Franch et al., 2023; Hafeez et al., 2021; Hess et al., 2019; Jia et al., 2019; Kudo et al., 2023; Liebel & Knauss, 2023; Medeiros et al., 2020; Mihelič et al., 2023; Noel et al., 2018; Nowroozi et al., 2020; Rojas & Macías, 2019; Salleh & Nohuddin, 2019; Schon et al., 2020; Younas et al., 2020)

Técnicas y artefactos utilizados en los Frameworks y Modelos de ER basados en MA

De los estudios analizados, se observa escasa referencia directa a una metodología ágil en particular. No obstante; se identificaron trabajos que utilizaron Scrum o procesos iterativos similares a Scrum, Sprints y Product backlog (Hess et al., 2019; Marques & Da Cunha, 2019; Melegati et al., 2019; Oriol et al., 2020; Rivero et al., 2019).

En la RSL, se han identificado diversas técnicas y artefactos utilizados por investigadores en el campo de la elicitación de requisitos. Estas herramientas desempeñan un papel fundamental en la captura efectiva de los requisitos del proyecto. Algunos de los métodos y artefactos más comunes mencionados en los estudios revisados incluyen Documentos de Requisitos Conjuntos, Mapas Mentales, Entrevistas, procesos de Scrum, la técnica MoSCoW, Historias de Usuario e Historias de Usuario de Software, entre otros (ver Tabla 7). Estos enfoques ofrecen un conjunto diverso de herramientas para abordar los desafíos en la elicitación y definición de requisitos en proyectos variados.

Tabla 7. Técnicas y artefactos utilizados en los Frameworks y Modelos de ER basados en MA

Técnicas/ Artefactos	Objetivo	Referencia
Documentos de requisitos conjuntos (JRD)	Gestionar requisitos, fomentando la colaboración y la mejora de la calidad.	Saeeda et al. (2020)
Procesos de Scrum o similares	Gestionar de manera iterativa y colaborativa los requisitos del sistema.	Marques & Da Cunha (2019); Melegati et al. (2019); Oriol et al.

		(2020); Rivero et al. (2019); Saeeda et al. (2020)
Priorización de Requisitos	Ayuda a los equipos de desarrollo a identificar y priorizar los requisitos más importantes y que generen valor al cliente y al negocio.	Hesham et al. (2021) Spijkman et al. (2021)
Técnica de Inyección de Conocimiento	Esta técnica es utilizada para inyectar el conocimiento implícito en los argumentos en los requisitos del usuario.	Elghariani et al. (2018)
Mapa de Argumentos	Es utilizado para visualizar los argumentos y su relación con los requisitos del usuario.	Elghariani et al. (2018)
Mapeo mental / Mapas mentales	Visualizar y estructurar la información durante la recolección de requisitos y representar gráficamente los requisitos recopilados y mejorar la comprensión de estos.	Hesham et al. (2021) Hess et al. (2019) Saeeda et al. (2020)
Product Backlog	Gestionar y priorizar los requisitos del sistema.	Akbar et al. (2018); Marques & da Cunha (2019); Oriol et al. (2020); Rivero et al. (2019); Tatale & Chandra Prakash (2020) (Akbar et al. (2018); Elghariani et al. (2018); Marques & Da Cunha (2019); Medeiros et al. (2018); Melegati et al. (2019); Muhammad et al. (2023); Nasiri et al. (2023b); Oriol et al. (2020); Rivero et al. (2019); Tatale & Chandra Prakash (2020)
Historias de Usuario e Historias de Usuario de Software	Describir funcionalidades desde la perspectiva del usuario.	(2018); Melegati et al. (2019); Muhammad et al. (2023); Nasiri et al. (2023b); Oriol et al. (2020); Rivero et al. (2019); Tatale & Chandra Prakash (2020)
Alertas de Calidad y Patrones de Requisitos de Calidad	Documentar y gestionar requisitos de calidad en proyectos de software.	Oriol et al. (2020)
Catálogo de Patrones de Requisitos de Calidad	Respaldar la generación y documentación de Requisitos de Calidad.	Saeeda et al. (2020)

Especificación RE4SA y RE4SA-Agile	Alinear requisitos funcionales y arquitectura funcional en equipos de desarrollo ágil.	Spijkman et al. (2021)
Inmersión superficial y profunda	Técnica que se utiliza para procesar los requisitos a través de fases de inmersión superficial y profunda para su elicitación. Estas pruebas se realizan desde la perspectiva del usuario final y se centran en validar si el sistema satisface las necesidades del usuario y si es fácil de usar. Son esenciales para garantizar que el software cumpla con las expectativas y necesidades del usuario antes de su implementación	(Saeeda et al., 2020)
Pruebas de Aceptación del Usuario	Generar pruebas de aceptación de manera eficiente y efectiva al considerar todas las combinaciones posibles de los parámetros de entrada.	Medeiros et al. (2018); Melegati et al. (2019); Muhammad et al. (2023); Tatale & Chandra Prakash (2020)
Pruebas combinatorias	Generar y documentar Requisitos de Calidad en el backlog del proyecto.	Tatale & Chandra Prakash (2020)
Modelo semiautomático respaldado por herramienta	Facilitar la gestión y mantenimiento de la trazabilidad y transparencia de los requisitos a lo largo de todo el ciclo de vida del proyecto. Además, posibilitan la representación visual de los requisitos mediante herramientas gráficas, ofreciendo una visualización clara y comprensible de las funcionalidades del sistema, que puede incluir prototipos, mockups, diagramas y otros recursos visuales.	Akbar et al. (2018)
Herramientas de gestión de requisitos y artefactos visuales.	Implican la interacción directa entre el equipo de desarrollo y los stakeholders o usuarios del sistema, con el objetivo de comprender sus necesidades y expectativas. Se realizan preguntas abiertas y cerradas para obtener	Medeiros et al. (2018); Melegati et al. (2019); Muhammad et al. (2023); Rivero et al. (2019); Spijkman et al. (2021)
Entrevistas		Melegati et al. (2019); Spijkman et al. (2021)

información detallada sobre los requisitos del sistema.

Talleres participativos para la captura de requisitos involucrar la participación activa de los stakeholders y miembros del equipo en sesiones estructuradas diseñadas para identificar, discutir y documentar los requisitos del sistema. Spijkman et al. (2021)

Prácticas y Recomendaciones para Mejorar la ER en MA

La tabla 8 resume las prácticas y recomendaciones identificadas para mejorar la ER en MA, destacando la importancia de la colaboración, adaptabilidad y consideración de aspectos específicos del proyecto para asegurar la satisfacción de los stakeholders.

Tabla 8. Prácticas y Recomendaciones para Mejorar la ER en MA

Práctica/ Recomendación	Objetivo	Referencia
Fomentar la comunicación y colaboración	Favorecer la interacción continua con los stakeholders y todo el equipo de desarrollo para comprender y adaptarse a sus necesidades del cliente.	Hesham et al. (2021); Hess et al. (2019); Marques & Da Cunha (2019); Medeiros et al. (2018); Melegati et al. (2019); Saeeda et al. (2020); Spijkman et al. (2021)
Adopción de técnicas iterativas, incrementales, colaborativas y artefactos visuales.	Incorporar procesos iterativos e incrementales, como Sprints, para facilitar la adaptación a cambios y asegurar entregas continuas, así como herramientas colaborativas, mapeo mental, mockups, prototipado, user stories, diagramas, entre otros.	Elghariani et al. (2018); Hesham et al. (2021); Hess et al. (2019); Marques & Da Cunha (2019); Medeiros et al. (2018); Melegati et al. (2019); Nasiri et al. (2023); Rivero et al. (2019); Saeeda et al. (2020); Spijkman et al. (2021)
Priorización Efectiva de Requisitos	Implementar fases de priorización para abordar primero los requisitos más críticos o valiosos que aporten mayor valor al cliente y al negocio.	Hesham et al. (2021); Hess et al. (2019); Medeiros et al. (2018)

Consideración Específica de Requisitos de Calidad	Centrarse en Requisitos de Calidad con herramientas como Alertas de Calidad y Patrones de Requisitos de Calidad.	Akbar et al. (2018)
Validación Práctica en Entornos Reales	Validar en entornos de desarrollo reales para evidenciar su eficacia y aplicabilidad.	Elghariani et al. (2018); Rivero et al. (2019); Saeeda et al. (2020)
Realizar pruebas de aceptación de usuario	Utilizar pruebas de aceptación como una forma de validar los requisitos directamente con los usuarios finales, lo que ayuda a garantizar que el software cumpla con las expectativas del cliente.	Hesham et al. (2021); Medeiros et al. (2018); Melegati et al. (2019)

CONCLUSIONES

La investigación subraya la importancia de la adaptabilidad y la colaboración constante en la ER en entornos ágiles. La diversidad de enfoques, desde la integración de metodologías ágiles hasta la atención específica a requisitos de calidad, resalta la necesidad de ajustar las prácticas a contextos particulares.

La colaboración intensiva con los stakeholders se posiciona como un pilar fundamental para mantenerse alineado con las expectativas cambiantes y asegurar la entrega de productos que satisfacen las necesidades reales.

La validación práctica de los frameworks y modelos propuestos en entornos reales refuerza la eficacia y la aplicabilidad de las estrategias de ER en MA. La implementación exitosa en empresas de desarrollo de software respalda la utilidad de estos enfoques en escenarios altamente cambiantes. Esta validación proporciona una base sólida para la confianza en las prácticas propuestas y destaca su relevancia en la mejora de procesos de ágiles.

A pesar de los avances identificados, la baja tasa de éxito promedio de los proyectos de software resalta la necesidad continua de investigaciones y mejoras en la ER en MA. La identificación de este desafío apunta hacia futuras oportunidades de investigación que podrían abordar deficiencias técnicas y sociales, contribuyendo así a la evolución y mejora continua de prácticas ágiles para el desarrollo de software.

Como trabajos futuros se tiene previsto la implementación concreta de un modelo de ER basado en las MA, tomando como referencia los patrones y prácticas exitosas identificadas en esta investigación. Este

modelo estará diseñado para abordar los desafíos específicos asociados a la captura de requisitos en entornos caracterizados por cambios constantes.

REFERENCIAS BIBLIOGRÁFICAS

Akbar, M. A., Sang, J., Khan, A. A., Amin, F. E., Nasrullah, Hussain, S., Sohail, M. K., Xiang, H., & Cai, B. (2018). Statistical Analysis of the Effects of Heavyweight and Lightweight Methodologies on the Six-Pointed Star Model. *IEEE Access*, 6, 8066–8079.

<https://doi.org/10.1109/ACCESS.2018.2805702>

Alrezaamiri, H., Ebrahimnejad, A., & Motameni, H. (2019). Software requirement optimization using a fuzzy artificial chemical reaction optimization algorithm. *Soft Computing*, 23(20), 9979–9994.

<https://doi.org/10.1007/s00500-018-3553-7>

Bierwolf, R. (2016). Project excellence or failure? Doing is the best kind of learning. *IEEE Engineering Management Review*, 44(2), 26–32. <https://doi.org/10.1109/EMR.2016.2568745>

Cadavid, A. N., Fernández Martínez, J. D., & Morales Vélez, J. (2013). *Revisión de metodologías ágiles para el desarrollo de software A review of agile methodologies for software development*.

Carrasco Gonzaga, M. K., Ocampo Pazos, W. J., Ulloa Meneses, L. J., & Azcona Esteban, J. (2019). Metodología híbrida de desarrollo de software combinando XP y Scrum. *Mikarimin*, 5(2), 109–116.

Casillo, F., Deufemia, V., & Gravino, C. (2022). Detecting privacy requirements from User Stories with NLP transfer learning models. *Information and Software Technology*, 146, 106853.

<https://doi.org/10.1016/j.infsof.2022.106853>

De Vries, M., & Opperman, P. (2023). Improving active participation during enterprise operations modeling with an extended story-card-method and participative modeling software. *Software and Systems Modeling*, 22(4), 1341–1368. <https://doi.org/10.1007/s10270-023-01083-8>

Dingsøyr, T., Moe, N. B., Fægri, T. E., & Seim, E. A. (2018). Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering*, 23(1), 490–520. <https://doi.org/10.1007/s10664-017-9524-2>

Durán, T. A. (2000). *Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información*. Universidad de Sevilla.

- Elghariani, K., Kama, N., Mohd Azmi, N. F., & Abu bakar, N. A. (2018). Implicit thinking knowledge injection framework for Agile requirements engineering. *International Journal of Advanced Computer Science and Applications*, 9(11), 141–146. <https://doi.org/10.14569/ijacsa.2018.091121>
- Franch, X., Palomares, C., Quer, C., Chatzipetrou, P., & Gorschek, T. (2023). The state-of-practice in requirements specification: an extended interview study at 12 companies. *Requirements Engineering*, 28(3), 377–409. <https://doi.org/10.1007/S00766-023-00399-7/TABLES/18>
- Hafeez, Y., Asghar, S., Arif, B., & Ali, S. (2021). Role of situational method engineering to improve visual information systems in agile distributed environment. *Multimedia Tools and Applications*, 80(6), 8881–8908. <https://doi.org/10.1007/s11042-020-09896-1>
- Hesham, A., Emam, O. E., & Salah, M. (2021). Enhanced Framework For Big Data Requirement Elicitation. *International Journal of Advanced Computer Science and Applications*, 12(8), 134–143. <https://doi.org/10.14569/IJACSA.2021.0120816>
- Hess, A., Diebold, P., & Seyff, N. (2019). Understanding information needs of agile teams to improve requirements communication. *Journal of Industrial Information Integration*, 14, 3–15. <https://doi.org/10.1016/j.jii.2018.04.002>
- Janampa Patilla, H., Gómez Enciso, E., Juárez Pulache, J. C., Lozano Rodríguez, J. L., Solórzano Huallanca, E., & Meneses Conislla, Y. (2021). Modelo de Gestión de Desarrollo de Software Ágil mediante Scrum y Kanban sobre la Programación Extrema. *RISTI: Revista Ibérica de Sistemas e Tecnologias de Informação*, ISSN-e 1646-9895, N°. Extra 43, 2021, Págs. 450-466, 43, 450–466.
- Jia, J., Yang, X., Zhang, R., & Liu, X. (2019). Understanding software developers' cognition in agile requirements engineering. *Science of Computer Programming*, 178, 1–19. <https://doi.org/10.1016/j.scico.2019.03.005>
- Kitchenham, B. A., & Charters, S. M. (2007). Guidelines for Performing Systematic Literature Reviews in Software Engineering (version 2.3). *Technical Report, Keele University and University of Durham*.
- Kudo, T. N., Bulcão-Neto, R. de F., Neto, V. V. G., & Vincenzi, A. M. R. (2023). Aligning requirements and testing through metamodeling and patterns: design and evaluation. *Requirements Engineering*, 28(1), 97–115. <https://doi.org/10.1007/s00766-022-00377-5>

- Liebel, G., & Knauss, E. (2023). Aspects of modelling requirements in very-large agile systems engineering. *Journal of Systems and Software*, *199*, 111628.
<https://doi.org/10.1016/j.jss.2023.111628>
- Marques, J., & Da Cunha, A. M. (2019). ARES: An Agile Requirements Specification Process for Regulated Environments. *International Journal of Software Engineering and Knowledge Engineering*, *29*(10), 1403–1438. <https://doi.org/10.1142/S021819401950044X>
- Medeiros, J., Vasconcelos, A., Silva, C., & Goulão, M. (2018). Quality of software requirements specification in agile projects: A cross-case analysis of six companies. *Journal of Systems and Software*, *142*, 171–194. <https://doi.org/10.1016/j.jss.2018.04.064>
- Medeiros, J., Vasconcelos, A., Silva, C., & Goulão, M. (2020). Requirements specification for developers in agile projects: Evaluation by two industrial case studies. *Information and Software Technology*, *117*, 106194. <https://doi.org/10.1016/j.infsof.2019.106194>
- Melegati, J., Goldman, A., Kon, F., & Wang, X. (2019). A model of requirements engineering in software startups. *Information and Software Technology*, *109*, 92–107.
<https://doi.org/10.1016/J.INFSOF.2019.02.001>
- Mihelič, A., Hovelja, T., & Vrhovec, S. (2023). Identifying Key Activities, Artifacts and Roles in Agile Engineering of Secure Software with Hierarchical Clustering. *Applied Sciences (Switzerland)*, *13*(7), 4563. <https://doi.org/10.3390/app13074563>
- Muhammad, A., Siddique, A., Mubasher, M., Aldweesh, A., & Naveed, Q. N. (2023). Prioritizing Non-Functional Requirements in Agile Process Using Multi Criteria Decision Making Analysis. *IEEE Access*, *11*, 24631–24654. <https://doi.org/10.1109/ACCESS.2023.3253771>
- Nasiri, S., Adadi, A., & Lahmer, M. (2023a). Automatic generation of business process models from user stories. *International Journal of Electrical and Computer Engineering*, *13*(1), 809–822.
<https://doi.org/10.11591/IJECE.V13I1.PP809-822>
- Nasiri, S., Adadi, A., & Lahmer, M. (2023b). Automatic generation of business process models from user stories. *International Journal of Electrical and Computer Engineering*, *13*(1), 809–822.
<https://doi.org/10.11591/IJECE.V13I1.PP809-822>

- Noel, R., Riquelme, F., Lean, R. Mac, Merino, E., Cechinel, C., Barcelos, T. S., Villarroel, R., & Munoz, R. (2018). Exploring collaborative writing of user stories with multimodal learning analytics: A case study on a software engineering course. *IEEE Access*, 6, 67783–67798. <https://doi.org/10.1109/ACCESS.2018.2876801>
- Nowroozi, A., Teymouri, P., Ramezanifarkhani, T., Besharati, M. R., & Izadi, M. (2020). A Crisis Situations Decision-Making Systems Software Development Process with Rescue Experiences. *IEEE Access*, 8, 59599–59617. <https://doi.org/10.1109/ACCESS.2020.2981789>
- Oriol, M., Martínez-Fernández, S., Behutiye, W., Farré, C., Kozik, R., Seppänen, P., Vollmer, A. M., Rodríguez, P., Franch, X., Aaramaa, S., Abhervé, A., Choraś, M., & Partanen, J. (2020). Data-driven and tool-supported elicitation of quality requirements in agile companies. *Software Quality Journal*, 28(3), 931–963. <https://doi.org/10.1007/s11219-020-09509-y>
- Patilla Hubner, J., Peralta Sotomayor, K., & Vásquez Luis Miguel, P. (2021). Modelo de desarrollo de software de la programación extrema sobre Scrum para gestión de software ágil. *RISTI: Revista Ibérica de Sistemas e Tecnologias de Informação*, ISSN-e 1646-9895, N°. Extra 39, 2021, Págs. 611-626, 39, 611–626.
- Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner's Approach 9th Edition*.
- Rivero, J. M., Grigera, J., Distanto, D., Montero, F., & Rossi, G. (2019). DataMock: An Agile Approach for Building Data Models from User Interface Mockups. *Software and Systems Modeling*, 18(1), 663–690. <https://doi.org/10.1007/S10270-017-0586-9>
- Robertson, S., & Robertson, J. (1999). *Mastering the requirements process*. Addison-Wesley.
- Rojas, L. A., & Macías, J. A. (2019). Toward collisions produced in requirements rankings: A qualitative approach and experimental study. *Journal of Systems and Software*, 158, 110417. <https://doi.org/10.1016/j.jss.2019.110417>
- Saeeda, H., Dong, J., Wang, Y., & Abid, M. A. (2020). A proposed framework for improved software requirements elicitation process in SCRUM: Implementation by a real-life Norway-based IT project. *Volume 32, Issue 7*, 32(7). <https://doi.org/10.1002/smr.2247>
- Salazar, J. C., Casallas, Á. T., Linares, J. C., Lozano, A., & Valbuena, Y. L. (2018). Scrum versus XP: similitudes y diferencias. *Tecnología Investigación y Academia*, 6(2), 29–37.

<https://revistas.udistrital.edu.co/index.php/tia/article/view/10496>

- Salleh, N. M., & Nohuddin, P. N. E. (2019). Comparative study between lean six sigma and lean-agile for quality software requirement. *International Journal of Advanced Computer Science and Applications*, 10(12), 212–218. <https://doi.org/10.14569/ijacsa.2019.0101230>
- Schon, E. M., Thomaschewski, J., & Escalona, M. J. (2020). Lean User Research for Agile Organizations. *IEEE Access*, 8, 129763–129773. <https://doi.org/10.1109/ACCESS.2020.3009101>
- Sherif, E., Helmy, W., & Galal-Edeen, G. H. (2023). Proposed Framework to Manage Non-Functional Requirements in Agile. *IEEE Access*, 11, 53995–54005. <https://doi.org/10.1109/ACCESS.2023.3281195>
- Spijkman, T., Molenaar, S., Dalpiaz, F., & Brinkkemper, S. (2021). Alignment and granularity of requirements and architecture in agile development: A functional perspective. *Information and Software Technology*, 133, 106535. <https://doi.org/10.1016/j.infsof.2021.106535>
- Tatale, S., & Chandra Prakash, V. (2020). Enhancing acceptance test driven development model with combinatorial logic. *International Journal of Advanced Computer Science and Applications*, 11(10), 268–278. <https://doi.org/10.14569/IJACSA.2020.0111036>
- Watanabe, Y., Washizaki, H., Fukazawa, Y., Honda, K., Taga, M., Matsuzaki, A., & Suzuki, T. (2021). Data-Driven Persona Retrospective Based on Persona Significance Index in B-to-B Software Development. *International Journal of Software Engineering and Knowledge Engineering*, 31(1), 117–146. <https://doi.org/10.1142/S0218194021500029>
- Younas, M., Jawawi, D. N. A., Shah, M. A., Mustafa, A., Awais, M., Ishfaq, M. K., & Wakil, K. (2020). Elicitation of Nonfunctional Requirements in Agile Development Using Cloud Computing Environment. *IEEE Access*, 8, 209153–209162. <https://doi.org/10.1109/ACCESS.2020.3014381>
- Zuppa, F., Silva, J., Carraro, J. M., Gaitán, G., Decuzzi, G., Wilkinson, H., & Paez, N. (2019). *Desarrollo de Software Ágil en 10Pines*.