

# Metodologías, Estándares y Herramientas para la Elicitación de Requisitos en la elaboración de proyectos de software

1<sup>st</sup> Sanmartín Tapia Omar  
*Universidad Nacional de Loja (UNL)*  
*Ingeniería Sistemas*  
 Loja, Ecuador  
 omar.sanmartin@unl.edu.ec

2<sup>nd</sup> Martínez Chamba Ángel  
*Universidad Nacional de Loja (UNL)*  
*Ingeniería Sistemas*  
 Loja, Ecuador  
 angel.martinez@unl.edu.ec

3<sup>rd</sup> Medina Morocho Cristian  
*Universidad Nacional de Loja (UNL)*  
*Ingeniería Sistemas*  
 Loja, Ecuador  
 cristian.e.medina@unl.edu.ec

4<sup>th</sup> González Chillogalli Diana  
*Universidad Nacional de Loja (UNL)*  
*Ingeniería Sistemas*  
 Loja, Ecuador  
 dggonzalesch@unl.edu.ec

5<sup>th</sup> Montaña Guamán Johanna  
*Universidad Nacional de Loja (UNL)*  
*Ingeniería Sistemas*  
 Loja, Ecuador  
 johanna.montano@unl.edu.ec

**Abstract**—Este artículo describe un estudio sobre el mapeo y revisión sistemática de literatura con el fin de identificar, analizar y clasificar los artículos publicados sobre el uso de metodologías, estándares y herramientas utilizados para la elicitación de requerimientos para la elaboración de proyectos de software. Por ello se utilizaron diferentes repositorios como: IEEE library, ScienceDirect y ACM Digital Library, donde se encontraron 26 artículos los cuales tiene relación con la investigación. Entre los principales resultados encontrados se detallan las metodologías ágiles haciendo mención a Scrum y Xp, del mismo modo se señalan las herramientas preferidas para el proceso de extracción de requisitos como lo son StarUML y Jira. Además, detallan artículos que contienen estándares que implementan en la elicitación como: IEEE 830-1984, ISO 14598-1 entre otros.

**Index Terms**—Elicitación de Requerimientos, Metodologías, Herramientas y Estándares

**Abstract**—This article describes a study on the mapping and systematic review of literature in order to identify, analyze and classify the articles published on the use of methodologies, standards and tools used for the elicitation of requirements for the development of software projects. Therefore, different repositories were used such as: IEEE library, ScienceDirect and ACM Digital Library, where 26 articles were found which are related to the research. Among the main results found, the agile methodologies are detailed, mentioning Scrum and Xp, as well as the preferred tools for the requirements extraction process, such as Star UML and Jira. In addition they detail articles that contain standards that implement in the elicitation as: IEEE 830-1984, ISO 14598-1 among others.

**Index Terms**—Requirements of elicitation, Methodologies, Tools and Standards

## I. INTRODUCCIÓN

En la actualidad el diseño de software es un proceso muy complejo ya que requiere solucionar problemas utilizando implementaciones tecnológicas, haciendo de este proceso un

trabajo intensamente intelectual, que compromete la creatividad y el juicio de las personas involucradas [1], lo cual acarrea una serie de problemas, por ejemplo, para que un ingeniero en el ámbito de desarrollo, aporte con una solución para un cliente, es necesario dedicación y tiempo para entender lo que realmente se desea construir. Por esta razón nace la Ingeniería de Requisitos, siendo esta un conjunto de actividades, tareas y procesos de desarrollo de sistemas cuyos objetivos principales son definir las características de un software, para que cumplan las necesidades de los clientes, gestionando las peticiones que se realicen durante la especificación de requisitos, para mantener una trazabilidad entre los requisitos y los productos de desarrollo.

Una parte fundamental de la Ingeniería de Requisitos es la Elicitación, ya que según [1] la elicitación de requerimientos es una de las principales tareas que debe llevarse a cabo para la correcta implementación de un desarrollo software, su incorrecta especificación genera costos innecesarios a lo largo del proyecto e inclusive, su completo fracaso. Contemporáneamente el crecimiento exponencial del desarrollo de software en diversos campos para diversos usos, hace que la presión por el aporte de la calidad en el desarrollo, sea prioridad para los desarrolladores de software, en ese contexto existen un sinnúmero de metodologías, estándares y herramientas que aportan exponencialmente la extracción de requisitos, su principales objetivo es que el proyecto cumpla de acuerdo a las necesidades de los Stakeholders, con los plazos establecidos porque nos ayudan a documentar un marco de desarrollo, bajo estándares e implementado herramientas desde el comienzo de proyecto para evitar fallos en etapas siguientes a la elicitación y que el producto entregable sea un fracaso. Simultáneamente cabe mencionar que existen pocas revisiones

sistemáticas de Literatura (RSL) orientadas a la elicitación de requerimientos, ya que se le da más peso a otras etapas del desarrollo sin prever que esta es una etapa fundamental ya que propone los cimientos del comienzo del proyecto, entre ellas se encuentra la propuesta por [2], que hace un análisis de diferentes artículos enfocados en la licitación para el desarrollo de aplicaciones web utilizando metodologías ágiles, sin embargo el documento aunque es claro y conciso en sus aportaciones se puede evidenciar que no se aborda en conjunto una contrastación de las diferentes metodologías existentes estándares y herramientas que existen no solo en el desarrollo ágil como lo mencionado por [2], si no que existen un sinnúmero de metodologías, que no precisamente pertenecen a este tipo de metodologías ágiles, del mismo modo herramientas y estándares.

Para finalizar, cabe señalar que el documento se encuentra estructurado en 5 secciones. En la Sección 2 se detalla el Método de Investigación, el cual engloba aspectos tales como: preguntas de investigación, proceso de búsqueda, definición de los criterios de inclusión y exclusión utilizados, cadenas de búsqueda y la evaluación de calidad. En la Sección 3 se aborda los resultados de la investigación. En la 4 las diferentes discusiones que se plantean con los artículos encontrados y finalmente las conclusiones de la presente RSL. Cabe señalar que el proceso de investigación utilizó herramientas como Parsif para crear un repositorio en el cual se pueda llevar un seguimiento detallado del trabajo expuesto en el presente, en cada una de sus partes.

## II. MÉTODO DE INVESTIGACIÓN

Se llevo a cabo una revisión sistemática de la literatura para responder a cuestiones concretas e identificar cómo ha sido abordada la investigación por otros autores. Esta revisión sistemática de la Literatura se realizó en tres partes:

- 1) Definición para la búsqueda: en donde definimos las preguntas de investigación, proceso de búsqueda, los criterios de inclusión y exclusión, y finalmente la cadena de búsqueda.
- 2) Ejecución de la búsqueda: aplicamos la cadena de búsqueda en las librerías seleccionadas, e iniciamos la evaluación de calidad con los resultados obtenidos.
- 3) Discusión de los resultados: en esta parte analizamos los resultados obtenidos en la Revisión.

### Preguntas de Investigación

A partir de la temática central denominada “Elicitación de Requisitos” se planteó seis preguntas de investigación clasificadas en preguntas para el mapeo sistemático (MQ) y la revisión sistemática (RQ):

- MQ1: ¿Cuántos estudios se publicaron a lo largo de los años sobre la Elicitación de Requisitos?
- MQ2: ¿Cuáles son las revistas y congresos que han publicado en el área?
- MQ3: ¿Cuáles son las técnicas más utilizadas, dentro de las metodologías encontradas para la elicitación de Requisitos?

- RQ1: ¿Cuáles son las metodologías para la elicitación de requisitos?
- RQ2: ¿Qué herramientas de software existen actualmente para la elicitación de requisitos?
- RQ3: ¿Cuáles son los estándares utilizados para la elicitación de requisitos?

### A. Proceso de búsqueda

Se determinaron un conjunto de términos basados en las preguntas de investigación para construir la cadena de búsqueda. Se utilizó el método Picoc propuesto para definir el ámbito de la RSL:

- **Población (P):** “Ingeniería de Requisitos”.
- **Intervención (I):** “Elicitación de Requisitos”.
- **Comparación (C):** “Metodologías, Herramientas, Estándares”
- **Resultados (O):** “metodologías”; “estándares”; “herramientas de software”.
- **Contexto (C):** “Proyectos de Software”

### B. Definición de los criterios de inclusión y exclusión

Para filtrar los contenidos y obtener mejores resultados, se especificaron 7 criterios de inclusión (IC):

- IC1: Los artículos deben estar relacionados con el área de la Ingeniería de Requisitos.
- IC2: Los artículos deben identificar metodologías para la elicitación de requisitos.
- IC3: Los artículos deben detallar los estándares utilizados para la elicitación de requisitos.
- IC4: Los artículos deben detallar herramientas de software para la elicitación de requisitos.
- IC5: Los artículos deben estar escritos en inglés
- IC6: Los artículos deben haber sido publicados en congresos OR revistas científicas.
- IC7: Artículos que hayan sido publicados desde el 2015 en adelante.

Y se especificaron 10 criterios de exclusión (EC):

- EC1: Artículos que estén relacionados con la ingeniería de requisitos.
- EC2: Artículos que no contienen criterios de inclusión.
- EC3: Artículos que no contienen palabras clave.
- EC4: Los artículos que no posean información sobre metodologías OR herramientas de software OR estándares en la Metodología de elicitación de requisitos.
- EC5: Artículos que hayan sido publicados antes el 2015
- EC6: Artículos que no estén enmarcados dentro del área de ingeniería.
- EC7: Artículos que no detallen estándares utilizados en la elicitación de requisitos.
- EC8: Los artículos que no se encuentran escritos en inglés.
- EC9: Artículos que no contienen información acerca de las preguntas de investigación.

- EC10: Artículos que no contengan información acerca de los autores o miembros de la investigación.

### C. Cadena de Búsqueda

En esta fase se definieron palabras clave a través del método Picoc, que permitieron junto con la sinonimia de cada término, establecer una cadena de búsqueda, usando operadores lógicos como: “OR” para los conceptos similares y “AND” para localizar los conceptos complementarios. Los estudios se obtuvieron de las siguientes fuentes de búsqueda:

- **IEEE Library**  
 (“Requirement Engineering”) AND  
 (“Requirement elicitation”) AND  
 (“Methodologies” OR “Software” OR “Tools” OR “Standard”)
- **ScienceDirect**  
 (“Requirement Engineering”) AND  
 (“Requirement elicitation”) AND  
 (“Methodologies” OR “Software” OR “Tools” OR “Standard”)
- **ACM Digital Library**  
 (“Requirement Engineering”) AND  
 (“Requirement elicitation”) AND  
 (“Methodologies” OR “Software” OR “Tools” OR “Standard”)

La calidad de estas fuentes garantiza la calidad del estudio y de la información que se obtiene.

### D. Evaluación de Calidad

Una vez definida la cadena de búsqueda es muy importante evaluar la calidad de los documentos seleccionados, por lo tanto, los datos extraídos se analizaron mediante un método cualitativo, lo que conduce a una síntesis descriptiva de resultados. Para evaluar la calidad de los artículos preseleccionados, se establecieron las siguientes preguntas de investigación:

- QA1: ¿El autor hace mención de al menos un estándar para la Elicitación de Requisitos?
- QA2: ¿El autor utiliza o hace mención de al menos una herramienta de software para la Elicitación de Requisitos?
- QA3: ¿El autor utiliza al menos una metodología para la Elicitación de Requisitos?

## III. RESULTADOS

Los resultados obtenidos de la RSL en los siguientes pasos:

- 1) Se ejecutó las cadenas de búsqueda en cada base de datos y se obtuvieron 332 documentos o artículos.
- 2) De los 332 documentos, se detectaron y eliminaron 9 artículos duplicados (2.71%), quedando 323 artículos por revisar.
- 3) Los 323 artículos fueron revisados y analizados en su título y resumen, tomando en consideración los criterios de inclusión y exclusión. Del total se eliminaron 237

artículos (71.38%) que son irrelevantes al objeto de estudio, además se descartaron porque su argumentación referente a la **Metodología de elicitación de requisitos** es débil y no da contestación a las preguntas de investigación planteadas.

En la Tabla I se detalla el proceso de selección de los estudios referentes a cada repositorio virtual IEEE, Science Direct, ACM, y del total de documentos se seleccionó 86 documentos (25.90%), para evaluarlos a través de los criterios de calidad.

Table I  
ARTÍCULOS REVISADOS

Base de Datos	Artículos				
	Encontrados	Duplicados	Revisados	Eliminados	Seleccionados
IEEE	25	0	25	20	5
Science Direct	238	5	233	167	66
ACM	69	4	65	50	15
<b>Total</b>	<b>332</b>	<b>9</b>	<b>323</b>	<b>237</b>	<b>86</b>

- 4) Cada una de las preguntas planteadas para el desarrollo del control de calidad tiene un puntaje de 1 si se califica con “Sí”, 0.5 si la respuesta es “Parcial” y 0 si la respuesta se evaluó con “No”.
- 5) Se procedió a hacer una evaluación de calidad de los 86 artículos seleccionados donde se valoraba cada artículo en función de las 3 preguntas de investigación, en el caso que acertara con una pregunta se sumaba un punto al artículo. Si un artículo tiene un puntaje igual o superior a 2, será seleccionado para extraer su información. Los artículos que cumplieron con los criterios de calidad fueron 26, es decir un (8.13%) del total inicial.
- 6) Para finalizar se procedió a extraer información relevante a las preguntas de investigación de cada artículo aceptado.

### A. Informe del mapeo sistemático

#### MQ1: “¿Cuántos estudios se han publicado a lo largo de los años sobre la Elicitación de Requisitos?”

En el conjunto de investigaciones de 2015 – 2019 en el área de Elicitación de Requisitos se han publicado 26 artículos, de los cuales 8 se divulgaron en 2016, 6 en 2015, 5 en 2019, 3 en 2017.

Esto demuestra que en el año 2016 se publicaron más artículos referentes a la Elicitación de Requisitos y que 2020 fue un año de sequía de divulgación de artículos en este campo de estudio.

**MQ2: “¿Cuáles son las revistas y congresos que han publicado en el área?”**, en la Fig. 2, se detalla el número de revistas y congresos publicados en el área de Elicitación de Requisitos, donde podemos destacar que la mayoría de artículos publicados corresponden a Revistas con 23, mientras que en las publicaciones en Congresos encontramos 5.

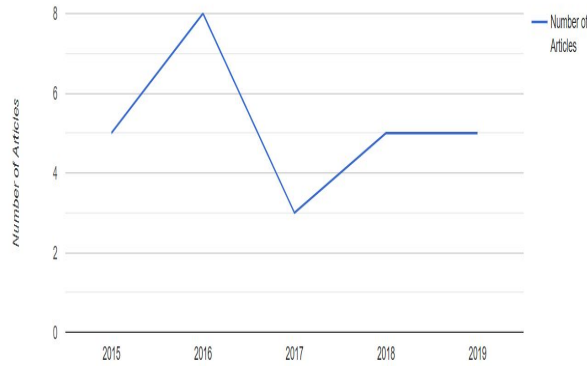


Figure 1. Publicaciones por año

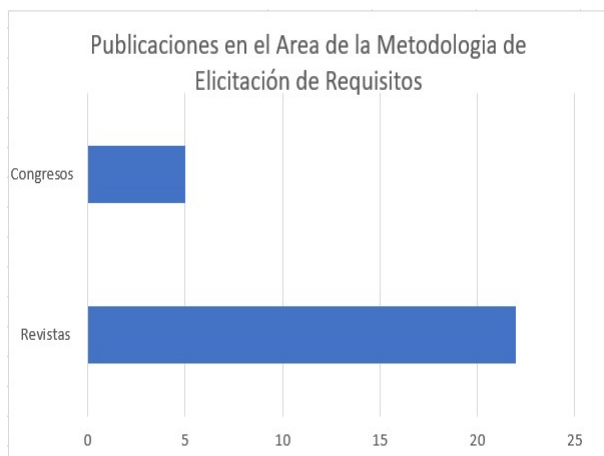


Figure 2. Lugar de publicación de artículos

**MQ3: “¿Cuáles son las técnicas más utilizadas, dentro de las metodologías encontradas para la elicitación de Requisitos?”**, las técnicas más utilizadas dentro de las metodologías de elicitación de requisitos son Joint Application Development(JAD), la Entrevista, Brainstorming y encuestas.

En la Tabla II se presenta los 26 artículos seleccionados donde se detalla la información por, el título, el año de publicación y su fuente de publicación (nombre del congreso o revista) para identificar posibles escenarios de publicación.

Table II  
ARTÍCULOS EVALUADOS

Título	Año / Revista / Congreso
Leveraging creativity in requirements elicitation within agile software development: A systematic literature review [3]	2019 The Journal of Systems and Software
Understanding software developers' cognition in agile requirements engineering [4]	2019 Science of Computer Programming
Validation to the Requirement Elicitation Framework via Metrics [5]	2015 SIGSOFT IEEE
Integrated product service offerings – Challenges in setting requirements [6]	2018 Journal of Cleaner Production
A systematic review on the relationship between user involvement and system success [7]	2015 Information and Software Technology
Chapter 7 - “Filling in the blanks”: A way to improve requirements management for better estimates [8]	2016 Software Quality Assurance
REASSURE: Requirements elicitation for adaptive socio-technical systems using repertory grid [9]	2017 Information and Software Technology
A systematic review of requirements change management [10]	2018 Information and Software Technology
Evaluating different i*-based approaches for selecting functional requirements while balancing and optimizing non-functional requirements: A controlled experiment [11]	2019 Information and Software Technology
Automated business rules and requirements to enrich product-centric information [12]	2019 Computers in Industry
A semi-automated approach for generating natural language requirements documents based on business process models [13]	2018 Information and Software Technology
Inviting Everyone to Play: Gamifying Collaborative Requirements Engineering [14]	2015 Fifth International Workshop on Empirical Requirements Engineering (EmpiRE) IEEE
A Requirements Elicitation Tool for Cloud-Based ERP Software Product Line [15]	2017 Proceedings of the 3rd Africa and Middle East Conference on Software Engineering IEEE
An Intelligent Planning Technique-Based Software Requirement Analysis [16]	2016 International Journal of Computational Science and Engineering IEEE
Chapter 1 - Requirements and Requirements Engineering This chapter serves to deliver general background knowledge about requirements and requirements engineering [17]	2018 Environment Modeling-Based Requirements Engineering for Software Intensive Systems
Enhancing requirements engineering for patient registry software systems with evidence-based components [18]	2017 Journal of Biomedical Informatics
A comparative study of software tools for user story management [19]	2015 Information and Software Technology
Grounded requirements engineering: An approach to use case driven requirements engineering [20]	2016 The Journal of Systems & Software

Early knowledge organization assisted by aspects [21]	2016 Science of Computer Programming
Automated design of multi-layered web information systems [22]	2016 The Journal of Systems and Software
10 - Requirement Management [23]	2016 Certifiable Software Applications
Bayesian Networks for Enhancement of Requirements Engineering: A Literature Review [24]	2016 Requirements Engineering IEEE
A novel data-driven graph-based requirement elicitation framework in the smart product-service system context [25]	2019 Advanced Engineering Informatics
Requirements Elicitation with Extended Goal Graph [26]	2016 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems
Knowledge management in requirement elicitation: Situational methods view [27]	2018 Computer Standards & Interfaces
Automated Requirements Extraction for Scientific Software [28]	2015 ICCS 2015 International Conference on Computational Science

### B. Extracción de la Información

Los criterios de selección de estudios establecen la pauta de extracción de información relevante. Por cada uno se sintetizó y clasificó a cada artículo para tener una visión clara de las preguntas RQ1 y RQ2 como se describe en la Tabla III y Tabla IV.

Table III  
METODOLOGÍAS POR ARTÍCULO

Metodologías	Referencias
DESIGN THINKING	[3]
JAD	[7]
SCRUM	[4], [10], [18], [19], [23]
XP	[4], [18], [19], [23]
QUESTIONNAIRE SURVEYS	[7]
REASURE	[9]
RUP	[11]
DALTON	[12]
SDLC	[7]
CASCADA	[17]
MODELADO DE REQUISITOS	[17]
GROUNDDED	[20]
INGENIERÍA DE REQUISITOS ORIENTADA ASPECTOS	[21]
EKOE	[21]
MOCKUPTOME	[22]
MÉTODOS SITUACIONALES	[18], [24], [28]
DRUMS	[28]

Table IV  
HERRAMIENTAS SOFTWARE UTILIZADAS POR ARTÍCULO

Herramienta Software	Artículo
StarUML	[5]
Maiden Developing UX	[3]
Jira	[4], [10]
Quality Function Deployment (QFD)	[6]
Trello	[7]
Project Manager	[7]
OpenRepGrid	[9]
PaladinRm	[11]
WebRed Case Tool	[11]
Product Lifecycle Management (PLM)	[12]
Building Information Modeling (BIM)	[12]
Eclipse Modeling Framework (EMF)	[13]
Eclipse Graphical Modeling Framework (EGMF)	[13]
Code Hunt	[14]
Active Integration Architecture PL4x	[15]
Framework of algorithm IPRA	[16]
CIPROS	[17]
Planbox	[19]
tinyPM	[19]
ScrumDesk	[19]
VersionOne	[19]
Model Merge Tool (MMT)	[20]
The Ontology Web Language (OWL)	[21]
MockupToME tool	[22]
DOORS	[23]
RTM	[23]
Rational Requisite Pro	[23]
CARE	[24]
SRW	[24]
BOSH	[24]
Smart PPS	[25]
Extended Goal Graph (EGG)	[26]
SRMS	[27]

Table V  
ESTÁNDARES POR ARTÍCULO

Estándar	Artículo
ISO 14598-1	[8]
ISO 15939	[8]
ISO 25010	[8]
IEEE 830-1984	[18]
ISO/IEC/IEEE 29148	[18]
IEEE 1471	[21]
EIA 632	[23]

## IV. DISCUSIÓN

### A. Metodologías analizadas / utilizadas durante la elicitación de Requisitos

El artículo [3] proporciona un análisis de la metodología DESIGN THINKING, esta metodología posee cinco etapas determinantes para la elicitación las cuales son: definición, idealización, empatización, Idealización, testeo e implementación de un prototipo, sin embargo la trascendencia de mayor impacto es la creatividad, dentro del desarrollo de esta metodología, ya que ella representa un factor determinante para la elicitación, porque emplea la participación activa de los usuarios y la de los demás participantes, pues la perspectiva, aportada por cada integrante, es clave determinante para obtener requerimientos por ende facilita la implementación de una solución técnica para la parte del desarrollo de un proyecto de software. Por otra parte [3] recomienda, trabajar con metodologías ágiles como Scrum, si el objetivo fuera fusionar metodologías para convertirla en una híbrida, ya que esta facilita la entrega de productos terminados de forma incremental.

En el artículo [7] analiza la metodología JAD, cuando se busca obtener requisitos rápidamente, pues los requisitos en esta metodología se obtienen en un solo taller con todas las partes interesadas como es el cliente, líder y desarrollador.

En los artículos [4], [10], [19], [23] se describe la Metodología SCRUM, como una metodología ágil, en donde la elicitación de requisitos se lleva a cabo con un trabajo en equipo dinámico, por lo tanto la obtención de requisitos y la entrega del producto final no tarda mucho tiempo para ser entregado al cliente, pues esta metodología suprime la documentación necesaria y se enfoca solo en aquellas que son útiles para los clientes, los autores coinciden en que Scrum trabaja principalmente con historias de Usuario, por esto, [19] señala que es fundamentalmente importante definir bien las mismas, ya que de no hacerlo, los requisitos obtenidos no serán lo suficientemente completos para tener un producto final eficaz [10] recomienda la utilización de metodologías ágiles cuando se trata de requisitos cambiantes.

Los artículo [19], [23] analiza la metodología XP (Extreme Programing) definen a XP como una metodología Ágil que trabaja con historias de usuarios, con el objetivo de organizarlas, priorizarlas, y usarlas para la planificación, gestión y pruebas. [19] utiliza XP, pues lo señala como práctico para desarrollar la gestión sistemas software ya que está basada en valores y prácticas, en donde el cliente es tratado como un miembro del equipo junto con los desarrolladores, pues proporciona información para discutir los problemas del proyecto lo cual permite de forma ágil obtener los requisitos del usuario. Por otra parte [23] indica que el margen de error al programar un entregable en XP, es bajo, ya que la programación del software se hace en parejas, teniendo la oportunidad de revisión mutua e intercambio de

ideas y soluciones.

El artículo [7] utiliza una metodología denominada QUESTIONNAIRE SURVEYS para obtener requisitos de una empresa con un proyecto pequeño, y se basa en la aplicación de un cuestionario a los usuarios para obtener conclusiones del problema, formular tareas y requisitos para finalmente dar soluciones técnicas a los mismos.

El artículo [9], nos presenta una metodología denominada REASSURE (Obtención de requisitos para sistemas socio-técnicos adaptativos usando repertorio) aplicada en la obtención los requisitos de un hogar inteligente, y se basa en identificar las dimensiones del espacio del problema, obteniendo requisitos en el contexto, económico, ambiental y social del usuario, lo cual conlleva a un trabajo en conjunto con el cliente, no solo enmarcado a un sistema software si no al contexto personal del usuario.

El artículo [11], busca optimizar la obtención de requisitos haciendo uso de la metodología RUP (Proceso Unificado de Rational), con la cual se espera tener un sistema software correcto, completo, consistente y verificable. ya que en esta metodología existe una interacción continua con el usuario lo cual mejora la obtención de requisitos, pues ésta, se adapta a las necesidades del usuario, por otro lado, el estudio indica que los requisitos se trabajan con casos de uso y los productos entregables se desarrollan en diferentes iteraciones.

En el artículo [12], se utiliza la metodología DALTON (Data Linked Through Occurrences Network), para un diseño complejo de un software de una empresa de energía nuclear. Esta metodología aplica para proyectos complejos ya que la gestión de requisitos se vuelve compleja, en el ejemplo analizado en el artículo participan más 100 personas por varios años, por lo cual, en DALTON primero se establecen reglas del funcionamiento del sistema, y a partir de ellas se forman los requisitos.

El artículo [7], analiza el método SDLC (Ciclo de Vida del desarrollo del Sistema), el cual consta de 7 fases en su implementación, los cuales son: planeación del concepto, definición de requisitos, diseño, desarrollo y pruebas, puesta en marcha, operaciones y mantenimiento, y disposición, de modo que, en cada una de sus etapas se mantiene la participación activa del usuario, cuidadosamente planificada, pues para ello se requiere de tiempo y recursos tanto por parte del usuario como de los desarrolladores, [7] recalca la importancia de involucrar a los usuarios durante el ciclo de vida de desarrollo del software, en especial durante la obtención de requisitos, para captar con precisión sus necesidades, y durante las etapas finales como son diseño y pruebas, pues es donde esos requisitos se transforman en soluciones técnicas, esta participación activa conlleva al éxito del Sistema software.

En el artículo [15], se describe la metodología ERP basada en la nube, en donde la elicitación de requisitos se desarrolla de manera conjunta entre el ingeniero de requisitos y el dueño del producto, a través de una aplicación web, que los conecta por medio de una cuenta de usuario, esto les permite gestionar los requisitos, y generar una documentación de los mismos, de manera automatizada. En esta metodología no se hace mención de reuniones personales, y la elicitación de requisitos se genera por medio de una plataforma en la nube.

En el artículo [17], se definieron dos metodologías para la obtención de requisitos durante el desarrollo de un proyecto software, primero está la metodología del modelo de Cascada como un modelo lineal en el diseño de un software, esta metodología se utilizó para realizar mejoras en un sistema ya existente, la obtención de requisitos se definen mediante diálogos con el usuario para establecer las necesidades actuales del sistema; la segunda metodología analizada en este documento es la Metodología de Modelado de requisitos, que se desarrolla en 6 etapas, en las cuales se mantiene una constante comunicación con el usuario para obtener los requisitos completos y correctos, esta metodología separa los requisitos obligatorios de los opcionales para poder crear un sistema más flexible.

En el artículo [20], utilizó la metodología GROUNDED, para resolver problemas bastante complejos y novedosos en donde la mayoría de los requisitos no son obvios, pero sí pueden observarse en el mundo real. Esta metodología se desarrolla en 5 etapas: 1) Recopilar datos, 2) Transcribir videos, 3) Codificación abierta, 4) Codificación axial y 5) Codificación selectiva, en esta metodología, primeramente, se analizan las observaciones del mundo real, para describir el comportamiento del usuario al realizar una tarea, y, en segundo lugar, estas descripciones llegan a ser los requisitos del sistema, para consecuentemente ser transferidos a diagramas de casos de uso.

El artículo [21], propone la metodología de Desarrollo del Software Orientada a Aspectos, para facilitar la identificación, separación y clasificación de intereses del usuario, pues el levantamiento de requisitos en esta metodología se desarrolla de acuerdo a la perspectiva del cliente. Esta metodología trabaja ligada a diagramas de caso de uso y a programación en diferentes módulos, el artículo combina la metodología anterior, con la Metodología EKO, la cual, aplica un lenguaje natural al momento de obtener requisitos para facilitar la participación y entendimiento del cliente.

Según [22], analiza metodologías en Ingeniería basada en modelos, en donde, la construcción del software depende de tareas de diseño, estas tareas pueden conllevar meses en algunos casos, por los autores presentar la metodología MockupToME que incluye tareas respaldadas por técnicas de diseño (semi) automatizadas para algunos patrones de casos de uso.

En el artículo [27], plantea un desarrollo de software basado en la ingeniería de métodos, pues al ser los proyectos diferentes, se busca que metodología se adapta mejor al mismo, a esto el autor lo llama Métodos Situacionales, la misma que contiene varias técnicas de obtención de requisitos, como : [18], [24] opiniones, encuestas / cuestionarios, grupos focales, lluvia de ideas, etnografía, talleres de requisitos, prototipos, etc. Cada técnica tiene ciertas ventajas y desventajas, y ninguna de estas técnicas puede ser necesariamente ideal para todas las circunstancias. Estos métodos se seleccionan de acuerdo a la familiaridad de los métodos con los analistas de requisitos y participantes, preferencia de métodos, conformidad con la metodología adoptada para la elicitación y la mentalidad de los analistas, y relevancia para la situación.

En el artículo [28], se desarrolla la extracción de requisitos en proyectos de software científico. Para ello se hace uso de la metodología DRUMS (Domain Specific Requirements Modeling para for Scientific), En esta metodología los requisitos básicos definidos incluyen: Modelo, Método de cálculo, Assumption, definición de datos, proceso, interfaz, hardware, rendimiento y restricción. se aplica para requisitos en proyectos de cálculos matemáticos.

El artículo [14], se describe la metodología de Gamificación, enfocada principalmente en el potencial de la participación del usuario para mejorar la calidad de los requisitos, utilizando diferentes técnicas para lograr la cooperación de todos los participantes, en este artículo se aplica una técnica tomada de un juego denominado “ integración de la mecánica del juego en entornos de juego para aumentar la participación de la audiencia, la lealtad y diversión ”, esto, para animar a los usuarios a participar y contribuir en aplicaciones informáticas. En esta metodología los usuarios tienen un papel participativo en el juego, ya que intervienen activamente en la creación y verificación de requisitos, Durante el transcurso del juego, los requisitos son revisados y validados.

#### *B. Herramientas Software utilizadas en la Elicitación de Requisitos*

En el artículo [5] se utiliza la herramienta startUML la cual es un sofisticado software de modelamiento destinado a admitir modelos ágiles y concisos.

En , se utiliza Maiden Developing UX para la creación de prototipos que permite plasmar las ideas en una interfaz gráfica para contrastarlas más fácilmente y de manera visual.

En los artículos [4] y [10] se utiliza Jira que es una herramienta en línea para la administración de tareas de un proyecto, el seguimiento de errores e incidencias y para la gestión operativa de proyectos, está disponible para varios sistemas operativos

En el documento [5], se utiliza la herramienta Visual Paradigm para la gestión de Requisitos incluido el análisis de

texto y además admite 14 tipos de diagrama UML.

Según [6], se utiliza la herramienta QFD (Quality Function Deployment) para traducir las necesidades del cliente en parámetros cuantitativos para lograr calidad en el diseño de sistemas.

Según el artículo [7], se usaron 2 herramientas la primera Trello para gestionar proyectos con una colaboración sencilla, divertida, visualiza datos e informes en tiempo real y ayuda a simplificar tareas, la segunda herramienta Project Manager para visualizar los datos y obtener informes en tiempo real de todos los proyectos del equipo centralizando su método en la comunicación entre los participantes del proyecto para además ayuda a gestionar y visualizar el trabajo en equipo.

Según [9], se utiliza la herramienta OpenRepGrid para analizar datos de cuadrícula de un repertorio, crear gráficos, calcular índices y generar informes.

En [11], se utiliza la herramienta PaladinRm para describir el comportamiento y para definir relaciones entre los requisitos con estructuras gráficas. Además, este artículo de igual manera utiliza la herramienta WebRed Case Tool para ayudar al diseñador en las primeras fases del proceso de desarrollo web.

En el artículo [12], se utilizan las herramientas PLM y BIM. PLM se utiliza para gestionar información a lo largo del ciclo de vida de un sistema desde su inicio hasta su eliminación. BIM se utiliza para crear y gestionar proyectos centralizando su información en los recursos digitales.

Según [13], se utilizan las herramientas EMF (Eclipse Modeling Framework) que es un marco de modelado utilizado para construir herramientas y aplicaciones basadas en un modelo de datos ya estructurados mientras que EGMF (Eclipse Graphical Modeling Framework) es un marco de aplicación que sirve para crear editores gráficos esto con un enfoque de arquitectura basada en modelos para el dominio de editores gráficos.

En [14], se utiliza Code Hunt el cual es un juego basado en la nube, donde los jugadores codifican en todos los niveles para adquirir y mejorar habilidades en lenguajes de programación su uso destaca para motivar la colaboración y el intercambio de conocimientos entre programadores para así incentivar y potenciar la tarea de la Ingeniería de Requisitos.

En el documento [15], se hace mención de PL4x (Active Integration Architecture) que es un software de incorporación que permite la integración de datos bidireccionales y con ello el acoplamiento de procesos entre diversos módulos como Teamcenter y SAP Business, Oracle EBS u otras aplicaciones empresariales, entre las ventajas de la herramienta tenemos que es adaptable a los requisitos comerciales, conjunto a la presentación de información y procesos Product Lifecycle Management (PLM) para usuarios y roles específicos.

En el artículo [16] encontramos Framework of Algorithm

Intelligent Planning-based Requirement Analysis (IPRA) una herramienta para aprender modelos de acción con efectos condicionales y efectos probabilísticos con el fin de aplicar este algoritmo para obtener el dominio de planificación completo y la especificación de los requisitos de forma automática usando los métodos de planificación automatizada y una técnica de aprendizaje automático para traducir los requisitos de software.

Según [17], encontramos CIPROS la cual es una lista que nos ayuda a los desarrolladores a evaluar los requisitos de los sistemas existentes y refuerza el informe de las descripciones de los sistemas de software. CIPROS es el resultado de un análisis cualitativo del texto.

En [19], se utilizan diversas herramientas como Planbox que es una plataforma de gestión ágil que cubre una gran parte de los requisitos funcionales y a su vez gestiona las historias de usuarios, tinyPM que es una herramienta ligera para una colaboración ágil que incluye gestión de productos, trabajo atrasado, panel de tareas, historias de usuarios, wiki e integraciones, scrumDesk que se usa para la gestión de Proyectos bajo la metodología Scrum, desde la creación de las historias de usuario hasta la realización de los incrementos, y por último VersionOne que gestiona proyectos, rastrea y mantiene registros de todas las fases además admite metodologías como Scaled Agile Framework (SAFe), Enterprise Scrum, Kanban, DAD, LeSS o una metodología híbrida.

En el artículo [20], se utiliza MMT (Model Merge Tool) qué nos sirve para editar y examinar modelos de aplicación en un módulo o capa diferente de usuarios.

En [21], se utiliza OWL (The Ontology Web Language) que es una herramienta para procesar el contenido de la información, lo que la convierte en el lenguaje más conocido para la descripción semántica de las ontologías, hay dos tipos principales de propiedades que OWL representas como relaciones: "Propiedades de los objetos" y "Propiedades de los datos". OWL se integra con la metodología "Early Knowledge Organization assisted by Aspects" (EKOA)

En [22], se utiliza MockupToME tool que es una herramienta para diseñar maquetas con el fin de describir interacciones del cliente/propietario de un producto software, además dispone de un catálogo de plantillas Mockups TOME tool se usa con la metodología MockupToME Method.

En el artículo [23], pudimos encontrar 3 herramientas que permiten la adquisición de requisitos, el establecimiento de la aptitud, la generación de informes y la documentación de apoyo a la ingeniería de requisitos, las cuales son DOORS desarrollada para la gestión de requisitos con un almacenamiento estructurado con una variedad de herramientas para el análisis y la automatización de diferentes pasos, además cuenta con su lenguaje llamado DXL (DOORS eXtended Language), RTM (Requirements Management Tool) utilizada para el apoyo automatizado con



la gestión automatizada de requisitos, Rational Requisite Pro dedicada al aspecto de la trazabilidad, proporciona tipos de requisitos definibles por el usuario además de crear proyectos configurables con plantillas de documentos.

En el documento [24], se encontraron diferentes herramientas de software específicamente 3 las cuales son CARE que es una herramienta de ingeniería de software asistida por ordenador, SRW (System Requirement Web) representa las relaciones entre los requisitos, BOSH ofrece una cadena de herramientas para ingeniería de versiones, la implementación de Software y para gestionar el ciclo de vida de aplicaciones.

En el artículo [25], se utiliza la herramienta Smart SPP en donde los datos generados por el usuario y los datos detectados por el producto se recogen durante la fase de utilización, en la que los posibles requisitos se pueden obtener fácilmente de una manera de creación de valor con conocimiento del contexto.

En el artículo [26], el autor utiliza EGG (Extended Goal Graph) como herramienta de apoyo para la reunión de obtención de requisitos y soluciones gracias al descubrimiento de conflictos para promover la discusión en la búsqueda de nuevas soluciones y criterios a través del experimento.

EGG tiene éxito en la extracción de premisas y restricciones que los interesados sostienen implícitamente detrás de los requisitos.

En el documento [27], el autor utiliza SMRS (Situational Requirement Method System) para extraer con facilidad el método utilizado en la fase de obtención de requisitos que sea más adecuado para su situación de proyecto además se pueden crear nuevos métodos de acuerdo con su experiencia previa, SRMS cuenta con dos tipos de actores: El usuario y el administrador.

### *C. Estándares utilizados para la elicitación de requisitos*

El artículo [8], se usa el estándar ISO 14598-1 este proporciona un marco de trabajo para evaluar la calidad de todo tipo de producto software e indica los requisitos para los métodos de medición y el proceso de evaluación, proporcionando métricas y requisitos para los procesos de evaluación, a través de 6 etapas. El artículo [8], se usa el estándar ISO 15939 define un proceso de medición del software aplicable a todas las disciplinas de ingeniería y de gestión relacionados con el software. El proceso de medición de software en esta norma se describe a través de un modelo que define las actividades del proceso de medición, que son necesarios para especificar adecuadamente la información solicitada. El artículo [8], se usa el estándar ISO 25010 es un modelo de calidad compuesto de 8 características que se relacionan con las propiedades estáticas del software y las propiedades dinámicas del sistema informático.

El artículo [18] se usa el estándar IEEE 830-1984 para la especificación de los requisitos del software, y se hace uso del estándar ISO/IEC/IEEE 29148 define la construcción de un buen requisito, proporciona atributos y características

de los requisitos y analiza la aplicación iterativa y recursiva de los procesos de requisitos a lo largo del ciclo de vida. Este documento proporciona orientación adicional en la aplicación de procesos de gestión e ingeniería de requisitos para actividades relacionadas con requisitos. Estos estándares describen métodos de obtención de requisitos de software y proporcionan especificaciones de requisitos de software. Sin embargo, son genéricos y poco específicos. Por esta razón se propone un nuevo método para apoyar el proceso de ingeniería de requisitos este se basa en evidencia para requisitos de elicitación, en el cual se crea una lista de requisitos basados en evidencias llamado lista de verificación.

El artículo [21], se usa el estándar IEEE 1471 aborda las actividades de creación, análisis y mantenimiento de arquitecturas de sistemas intensivos en software, y el registro de dichas arquitecturas en términos de descripciones arquitectónicas. Se establece un marco conceptual para la descripción arquitectónica. En el modelo propuesto representa preocupación por los requisitos atómicos, es decir, requisitos que no pueden separarse de manera semántica. Este modelo resulta interesante para los interesados, existe una relación indirecta entre la estructura del modelo y los interesados, ya que la estructura del modelo identifica a los interesados los cuales establecen los tratos específicos para la creación, organización y análisis de la estructura.

El artículo [23], se usa el estándar EIA 632 define los procesos utilizados en la ingeniería de un sistema. Estos procesos son fundamentales para la ejecución de un programa de ingeniería de sistemas en un proyecto de desarrollo de productos. Esto incluye procesos de gestión técnica, adquisición y suministro, diseño de sistemas, realización de productos y evaluación técnica. Permite la adquisición de requisitos, establecimiento de trazabilidad, ingeniería de requisitos de soporte de generación de informes y documentación. Una deficiencia es que se presenta los requisitos de manera genérica por ello se junta varias técnicas de elicitación de requisitos (entrevista, prototipos).

## V. CONCLUSIONES

El presente trabajo investigativo encontró como resultado 26 artículos y después de ser analizadas cuidadosamente se encontraron 21 metodológicas, entre las más destacadas se encuentran las del manifiesto ágil como Scrum, XP como lo señala [4], [10], [19], [23] sin embargo cabe señalar que existen más metodologías señaladas en los artículos como Design Thinking, ERP Basado en la nube, Ingeniería de requisitos orientada a aspectos entre otras ya mencionadas en la parte de discusiones, el implemento de estas metodologías determina un papel importante en la elicitación de requisitos ya que a través de ellas se detallan la manera en que se puede desarrollar un proyecto de software dándole énfasis a la elicitación y especificación de requisitos para prever que el proyecto desarrollado no sea un completo fracaso.

En lo referente a las herramientas de software existen una gran variedad dedicadas a diferentes partes del proceso de elicitación de requisitos, donde pudimos encontrar herramientas encargadas a la creación de prototipos, creación de diagramas como Maiden Developing UX, Paladin RM, OpenRepGrid y StarUML, aparte de esos temas pudimos encontrar programas dedicados a gestionar la obtención de requisitos exclusivamente requisitos de creación de software como lo son ScrumDesk, Jira, Visual Paradigm, EMF, CIPROS, Planbox, versionOne y por otra parte encontramos herramientas para la gestión de requisitos de proyectos en general pero que se pueden adaptar para software como es el caso de PLM, BIM, Rational Requisite Pro, RTM (Requirements Management Tool). Otras herramientas obtenidas fueron Code Hunt el cual es una ayuda al proceso de la elicitación mediante la colaboración y el intercambio de conocimientos.

En lo referente a los estándares de software se pudo determinar que menos del 50% de los artículos seleccionados hacen uso de los estándares de software, además se pudo encontrar estándares como ISO 14598-1, ISO 15939, ISO 25010, IEEE 830-1984, ISO/IEC/IEEE 29148, IEEE 1471, EIA 632 los cuales son mayormente usados en proyectos de gran tamaño o importancia como es en la industria médica, educación, ciencia, banca y finanzas, etc.

#### REFERENCES

- [1] D. S. Cohn Muroy, "Análisis de la transparencia en la elicitación de requerimientos al combinar historias de usuario y casos de uso,"
- [2] Y. Mamani Coaquira, "Seguimiento de requisitos en proyectos de software basado en el enfoque de prácticas ágiles mediante una aplicación web," 2014.
- [3] A. Aldave, J. M. Vara, D. Granada, and E. Marcos, "Leveraging creativity in requirements elicitation within agile software development: A systematic literature review," *Journal of Systems and Software*, vol. 157, p. 110396, 2019.
- [4] J. Jia, X. Yang, R. Zhang, and X. Liu, "Understanding software developers' cognition in agile requirements engineering," *Science of Computer Programming*, vol. 178, pp. 1 – 19, 2019.
- [5] S. N. Bhatti, M. Usman, and A. A. Jadi, "Validation to the requirement elicitation framework via metrics," *SIGSOFT Softw. Eng. Notes*, vol. 40, p. 1–7, Sept. 2015.
- [6] S. Nilsson, E. Sundin, and M. Lindahl, "Integrated product service offerings – challenges in setting requirements," *Journal of Cleaner Production*, vol. 201, pp. 879 – 887, 2018.
- [7] M. Bano and D. Zowghi, "A systematic review on the relationship between user involvement and system success," *Information and Software Technology*, vol. 58, pp. 148 – 169, 2015.
- [8] L. Buglione, A. Abran, M. Daneva, and A. Herrmann, "Chapter 7 - "filling in the blanks": A way to improve requirements management for better estimates," in *Software Quality Assurance* (I. Mistrik, R. Soley, N. Ali, J. Grundy, and B. Tekinerdogan, eds.), pp. 151 – 176, Boston: Morgan Kaufmann, 2016.
- [9] S. Dey and S.-W. Lee, "Reassure: Requirements elicitation for adaptive socio-technical systems using repertory grid," *Information and Software Technology*, vol. 87, pp. 160 – 179, 2017.
- [10] S. Jayatilake and R. Lai, "A systematic review of requirements change management," *Information and Software Technology*, vol. 93, pp. 163 – 185, 2018.
- [11] J. Zubcoff, I. Garrigós, S. Casteleyn, J.-N. Mazón, J.-A. Aguilar, and F. Gomariz-Castillo, "Evaluating different i\*-based approaches for selecting functional requirements while balancing and optimizing non-functional requirements: A controlled experiment," *Information and Software Technology*, vol. 106, pp. 68 – 84, 2019.
- [12] V. Fortineau, T. Paviot, and S. Lamouri, "Automated business rules and requirements to enrich product-centric information," *Computers in Industry*, vol. 104, pp. 22 – 33, 2019.
- [13] B. Aysolmaz, H. Leopold, H. A. Reijers, and O. Demirörs, "A semi-automated approach for generating natural language requirements documents based on business process models," *Information and Software Technology*, vol. 93, pp. 14 – 29, 2018.
- [14] N. Unkelos-Shpigel and I. Hadar, "Inviting everyone to play: Gamifying collaborative requirements engineering," in *2015 IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE)*, pp. 13–16, 2015.
- [15] M. A. A. Elmoniem, E. S. Nasr, and M. H. Gheith, "A requirements elicitation tool for cloud-based erp software product line," in *Proceedings of the 3rd Africa and Middle East Conference on Software Engineering, AMECSE '17*, (New York, NY, USA), p. 1–6, Association for Computing Machinery, 2017.
- [16] C. Liu, "An intelligent planning technique-based software requirement analysis," *Int. J. Comput. Sci. Eng.*, vol. 13, p. 285–295, Jan. 2016.
- [17] Z. Jin, "Chapter 1 - requirements and requirements engineering this chapter serves to deliver general background knowledge about requirements and requirements engineering," in *Environment Modeling-Based Requirements Engineering for Software Intensive Systems* (Z. Jin, ed.), pp. 3 – 11, Oxford: Morgan Kaufmann, 2018.
- [18] D. Lindorfer and U. Mansmann, "Enhancing requirements engineering for patient registry software systems with evidence-based components," *Journal of Biomedical Informatics*, vol. 71, pp. 147 – 153, 2017.
- [19] S. Dimitrijević, J. Jovanović, and V. Devedžić, "A comparative study of software tools for user story management," *Information and Software Technology*, vol. 57, pp. 352 – 368, 2015.
- [20] D. Würfel, R. Lutz, and S. Diehl, "Grounded requirements engineering: An approach to use case driven requirements engineering," *Journal of Systems and Software*, vol. 117, pp. 645 – 657, 2016.
- [21] E. Barra and J. Morato, "Early knowledge organization assisted by aspects," *Science of Computer Programming*, vol. 121, pp. 34 – 54, 2016. Special Issue on Knowledge-based Software Engineering.
- [22] F. P. Basso, R. M. Pillat, T. C. Oliveira, F. Roos-Frantz, and R. Z. Frantz, "Automated design of multi-layered web information systems," *Journal of Systems and Software*, vol. 117, pp. 612 – 637, 2016.
- [23] J.-L. Boulanger, "10 - requirement management," in *Certifiable Software Applications 1* (J.-L. Boulanger, ed.), pp. 239 – 282, Elsevier, 2016.
- [24] I. M. Águila and J. Sagrado, "Bayesian networks for enhancement of requirements engineering: A literature review," *Requir. Eng.*, vol. 21, p. 461–480, Nov. 2016.
- [25] Z. Wang, C.-H. Chen, P. Zheng, X. Li, and L. P. Khoo, "A novel data-driven graph-based requirement elicitation framework in the smart product-service system context," *Advanced Engineering Informatics*, vol. 42, p. 100983, 2019.
- [26] N. Kushiro, T. Shimizu, and T. Ehira, "Requirements elicitation with extended goal graph," *Procedia Computer Science*, vol. 96, pp. 1691 – 1700, 2016. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 20th International Conference KES-2016.
- [27] D. Mishra, S. Aydin, A. Mishra, and S. Ostrovska, "Knowledge management in requirement elicitation: Situational methods view," *Computer Standards Interfaces*, vol. 56, pp. 49 – 61, 2018.
- [28] Y. Li, E. Guzman, K. Tsiamoura, F. Schneider, and B. Bruegge, "Automated requirements extraction for scientific software," *Procedia Computer Science*, vol. 51, pp. 582 – 591, 2015. International Conference On Computational Science, ICCS 2015.