

processes

IMPACT
FACTOR
2.8

CITESCORE
5.1

Article

What Is the Process? A Metamodel of the Requirements Elicitation Process Derived from a Systematic Literature Review

Mauricio Hidalgo, Fernando Yanine, Rodrigo Paredes, Jonathan Frez and Mauricio Solar

Special Issue

New Trends in Smart Grid and Sustainable Energy Systems

Edited by

Dr. Fernando Yanine and Prof. Dr. Sarat Kumar Sahoo



<https://doi.org/10.3390/pr13010020>

Article

What Is the Process? A Metamodel of the Requirements Elicitation Process Derived from a Systematic Literature Review

Mauricio Hidalgo ^{1,2,*}, Fernando Yanine ^{1,†}, Rodrigo Paredes ^{1,†}, Jonathan Frez ^{3,†} and Mauricio Solar ^{2,†}

¹ Facultad de Ingeniería, Universidad Finis Terrae, Providencia 7501014, Chile; fyanine@uft.cl (F.Y.); raparede@uft.cl (R.P.)

² Departamento de Informática, Universidad Técnica Federico Santa María, Valparaíso 2390123, Chile; msolar@inf.utfsm.cl

³ Escuela de Informática y Telecomunicaciones, Universidad Diego Portales, Santiago 8370191, Chile; jonathan.frez@mail.udp.cl

* Correspondence: mhidalgo@uft.cl or mauricio.hidalgo@usm.cl

† These authors contributed equally to this work.

Abstract: Requirements elicitation is a fundamental process in software engineering, essential for aligning software products with user needs and project objectives. As software projects become more complex, effective elicitation methods are vital for capturing accurate and comprehensive requirements. Despite the variety of available elicitation methods, practitioners face persistent challenges such as capturing tacit knowledge, managing diverse stakeholder needs, and addressing ambiguities in requirements. Moreover, although elicitation is recognized as a core process for gathering and analyzing system objectives, there is a lack of a unified and systematic framework to guide practitioners—especially newcomers—through the activity. To address these challenges, we provide a comprehensive analysis of existing elicitation methods, aiming to contribute to better alignment between software products and project objectives, ultimately improving software engineering practices. We do so by performing a systematic literature review identifying crosscutting steps, common techniques, tools, and approaches that define the core activities of the elicitation process. We synthesize our findings into a metamodel that structures software elicitation processes. This review uncovers various elicitation methods—such as collaborative workshops, interviews, and prototyping—each demonstrating unique strengths in different project contexts. It also highlights significant limitations, including stakeholder misalignment and incomplete requirements capture, which continue to reduce the effectiveness of elicitation processes. Finally, our study seeks to contribute to understanding requirements elicitation methods by providing a comprehensive view of their current strengths and limitations through a metamodel enabling the structuring and optimization of elicitation processes.

Keywords: systematic literature review; software engineering; requirements elicitation; elicitation process; metamodel



Academic Editor: Olympia Roeva

Received: 29 November 2024

Revised: 19 December 2024

Accepted: 21 December 2024

Published: 25 December 2024

Citation: Hidalgo, M.; Yanine, F.; Paredes, R.; Frez, J.; Solar, M. What Is the Process? A Metamodel of the Requirements Elicitation Process Derived from a Systematic Literature Review. *Processes* **2025**, *13*, 20.

<https://doi.org/10.3390/pr13010020>

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, the success of software projects has become increasingly linked to the quality of practices to obtain requirements. Eliciting accurate, complete, and well-defined requirements is often the first step toward ensuring that a software system meets users' needs and expectations [1,2]. However, requirements elicitation (RE) remains one of the most challenging aspects of software engineering (SWE), as it demands an understanding of both the technical aspects of software design and the often implicit, complex expectations of stakeholders [3]. Capturing tacit knowledge, which is frequently embedded

within stakeholders' implicit expectations, poses a particularly complex challenge due to its unstructured nature and reliance on contextual interpretation [4]. Moreover, aligning diverse stakeholder groups requires not only effective communication techniques but also methods for mitigating ambiguities and goal conflicts [5,6]. As a result, this phase of software development has received significant attention, leading to the exploration of various methodologies and techniques to address these inherent difficulties. In response, numerous processes and techniques have been developed to enhance communication, understanding, and alignment between stakeholders and developers [7]. Despite these advancements, RE processes still face persistent obstacles, including challenges in capturing tacit knowledge, managing diverse stakeholder perspectives, and resolving ambiguities arising from conflicting requirements [8]. Given the fragmented landscape of RE studies, a systematic review provides an opportunity to consolidate and synthesize diverse practices and challenges [9], thereby establishing a coherent theoretical foundation to guide practitioners in selecting suitable approaches for specific contexts. These issues are further compounded as software systems grow in complexity and serve increasingly varied purposes. Traditional elicitation methods, such as interviews and questionnaires, often fall short of addressing these nuanced challenges in contemporary development contexts.

Consequently, researchers and practitioners have adapted or developed a range of techniques, from collaborative workshops to iterative prototyping, to enhance the elicitation process and make it more adaptable to diverse project requirements [7,10]. While these approaches provide valuable strategies, they also introduce unique limitations and contextual dependencies that reduce their effectiveness. Understanding the strengths and limitations of these methods is crucial for selecting or adapting the most suitable approach for each project context, particularly as projects scale or incorporate new technologies and involve varied stakeholder groups.

To gain a comprehensive understanding of current RE practices, this paper conducts a systematic literature review (SLR) aimed at synthesizing and evaluating the state of RE processes in SWE. This review focuses on answering the following research questions:

- R.Q.1: What are the crosscutting steps and activities in requirements elicitation processes within software engineering?
- R.Q.2: What are the main challenges and opportunities associated with effectively executing requirements elicitation processes in software projects?
- R.Q.3: What are the elements required to define a metamodel for software requirements elicitation processes?

Answering these questions through an SLR enables a comprehensive understanding of the available methods, their most effective contexts, and potential areas for improvement [11]. Hence, we present an SLR on RE processes in SWE, aiming to identify key methods, challenges, and tools that support effective requirements capture. Our objectives are as follows:

- To identify crosscutting activities as foundational steps that span multiple phases of the requirements elicitation process, contextual factors that influence the requirements elicitation process, and essential techniques used in RE processes within SWE.
- To evaluate the main challenges and limitations associated with effective requirements elicitation in diverse project contexts.
- To propose a metamodel that structures and organizes the critical elements of elicitation processes, facilitating practical application.

Crosscutting activities include communication, validation, and iterative refinement. In fact, communication ensures that stakeholders remain engaged throughout all RE phases, while validation verifies that requirements align with stakeholder needs. Contextual factors

are environmental, organizational, or project-specific elements influencing the RE process; examples include project scale (e.g., small vs. large projects), stakeholder diversity (e.g., cultural or technical differences), and development methodology (e.g., agile or waterfall). Finally, the techniques include interviews and prototyping, among others.

Through this systematic review, we seek to consolidate existing practices and provide a comprehensive metamodel that guides practitioners in selecting and applying elicitation techniques suited to various project needs. For ease of reference, the structure of this paper is as follows. Section 2 provides the background of RE theory. Section 3 details the SLR methodology, including search strategies, selection criteria, and data extraction methods. Section 4 presents the results of the review, synthesizing the findings on the most prevalent processes, tools, and challenges. Based on our findings, we propose an initial metamodel for the RE process in Section 5, which is later refined in Section 6. Next, we analyze the validity threats to our SLR in Section 7 and discuss our findings in Section 8. We finish in Section 9, where we present the conclusions of this work and recommendations for future research. In addition, Appendix A provides summaries of the articles included in this work.

2. Requirements Elicitation

Requirements elicitation is the process of identifying, uncovering, gathering, and refining requirements for computer-based systems [12]. It is a fundamental activity focused on understanding and documenting the functionalities a software system must provide by obtaining detailed input from stakeholders [7]. This process involves a range of techniques and tools, both manual and automated, to effectively capture and formalize stakeholder needs [13,14]. Table 1 lists the most common techniques and tools for software requirements elicitation (SWRE). In this way, the techniques are chosen based on the product being developed, stakeholder characteristics, and the type of information needed [7]. In addition, automated tools and machine learning techniques are increasingly being adopted to handle the large volume of unstructured text documents from stakeholders, making the process more efficient and less error-prone [14,15].

From a theoretical perspective, Professor Sommerville [16] defined four activities for software requirements elicitation and analysis, as illustrated in Figure 1, presenting a high-level approach to a cyclical process. Although this model is widely recognized, RE is highly contextual. This underscores the importance of a systematic literature review (SLR) in defining a metamodel, as it allows for the incorporation of empirical studies that identify the crosscutting aspects applicable to various contexts.

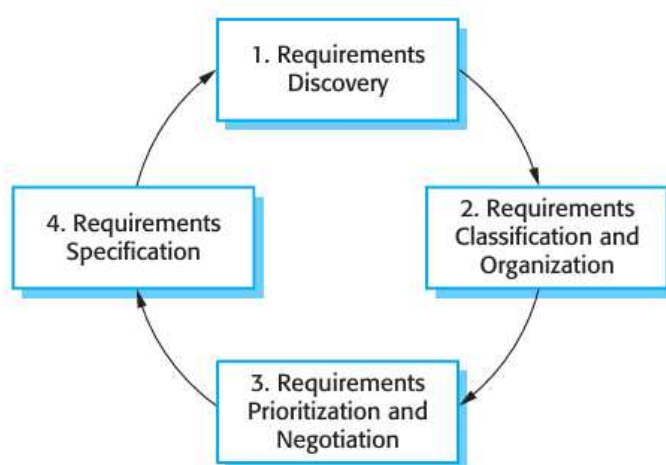


Figure 1. Professor Sommerville's requirements elicitation and analysis process [16].

Table 1. Software requirements elicitation techniques and tools.

Technique/Tool	Description
Brainstorming	Generating ideas and requirements through group discussion.
Document Analysis	Reviewing existing documents to identify relevant requirements.
Focus Group	Engaging a small group of stakeholders to discuss and refine requirements.
Interface Analysis	Analyzing interactions between systems to define requirements.
Interviews	Conducting structured conversations to gather requirements directly.
Observation	Studying end users in their environment to discover requirements.
Process Modeling	Diagramming workflows to understand requirements in context.
Prototype	Developing an early version of a system to refine requirements.
Requirements Workshops	Collaborative sessions to prioritize and define requirements.
Surveys/Questionnaires	Using forms to collect requirements from a broad audience.

Finally, considering the contextual nature of the process, SWRE is recognized as an iterative process that evolves as the understanding of the requirements and the situation improves [17]. The selection of appropriate elicitation techniques is crucial and depends on situational knowledge [18,19]. Moreover, for large-scale projects, social networks and collaborative filtering can effectively identify and prioritize requirements, thereby helping manage the complexity and scale of stakeholder interactions [20]. To further enhance understanding of this process, systematic reviews contribute to comprehending the effectiveness of various elicitation techniques and improving the elicitation process itself [17–19].

3. Research Methodology: Systematic Literature Review

A systematic literature review (SLR) is a method used to evaluate and interpret all relevant information available for a specific research question, topic, or phenomenon of interest [21]. This process involves a comprehensive search to locate both published and unpublished works related to the subject, followed by a systematic integration and critique of the evidence [22,23]. Additionally, explicit methods, such as PRISMA guidelines, are used to minimize bias [24]. Given the rigor required for conducting an SLR, our protocol (depicted in Figure 2) is outlined in the following subsections. The SLR approach has already been used in SWE to prioritize SW requirements via use cases [25] and to engineer SW requirements in agile environments [26].

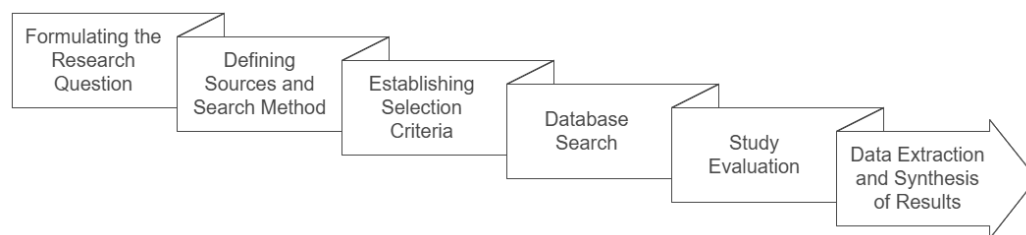


Figure 2. Systematic literature review protocol flow chart.

3.1. Research Questions

To establish the research questions presented in Section 1, we organized meetings with industry professionals, focusing the discussion on the following keywords: “requirements elicitation”, “software engineering processes”, “challenges in software projects”, “metamodels”, and “crosscutting activities”.

3.2. Defining Sources and Search Method

To achieve comprehensive coverage of the relevant literature, we carefully selected our sources and defined a rigorous search methodology, as illustrated in Figure 3.

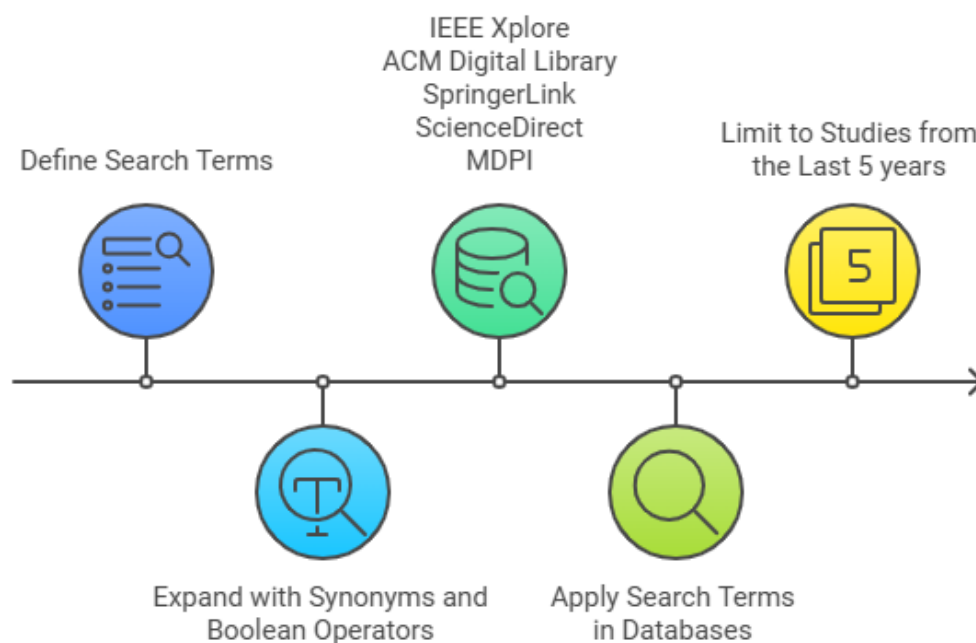


Figure 3. Chart defining the sources and search method. Image produced with Napkin AI assistance.

Firstly, we considered well-established document repositories recognized for their extensive collections in SWE, including IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, and MDPI, which provide access to a diverse array of journals, conference proceedings, and technical reports, allowing us to capture recent studies on the topic.

Secondly, we established a set of primary search terms based on the key themes in our research. The core terms included “requirements elicitation”, “software engineering processes”, “project challenges”, “metamodel”, and “crosscutting activities”. They were expanded with synonyms, related concepts, and Boolean operators (e.g., AND, OR) to refine and optimize the scope of the search results. For instance, we included terms like “requirements engineering” alongside “requirements elicitation” to broaden the scope and added “process limitations” to ensure comprehensive results related to challenges.

We applied the same search strings consistently across all repositories and limited our search to studies within the last 5 years to focus on contemporary insights.

3.3. Establishing Selection Criteria

In this stage, we established specific inclusion criteria to ensure a thorough and relevant selection of studies. The criteria are outlined as follows:

- Only empirical studies published between 1 June 2019, and 31 July 2024.
- Studies published in English to facilitate comprehension and accessibility.

By prioritizing empirical research, we aimed to capture practical insights and challenges encountered in SWRE processes, reflecting the reality faced by practitioners, which would enable us to define a metamodel derived from these experiences.

3.4. Database Search

With the established keywords—namely, “requirements elicitation”, “software engineering processes”, “project challenges”, “metamodel”, and “crosscutting activities”—we conducted a systematic search across the selected document repositories (or databases, for brevity). By combining these keywords with logical operators (e.g., AND, OR), we performed the search to retrieve studies that directly address the research questions. This stage, as illustrated in Figure 4, was iterative, requiring periodic adjustments to the keywords and search strings to optimize both the relevance and comprehensiveness of the results.



Figure 4. Iterative stage of the database search. Image produced with the assistance of Napkin AI.

Each digital repository—that is, IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, and MDPI—was queried separately, and the retrieved studies were then compiled and filtered based on the inclusion and exclusion criteria, ensuring a refined and relevant dataset for further analysis.

3.5. Study Evaluation

To ensure a thorough and unbiased selection of studies, we guided our process through four key phases, as illustrated in Figure 5. These phases are described as follows:

- **Identification of Studies:** We conducted comprehensive searches of the chosen databases using predefined keywords and logical operators. This step ensured that we captured all relevant literature on the topic. Additionally, references from selected articles were reviewed to identify any further studies that might contribute to our synthesis.

- **Initial Screening:** We performed a preliminary review of titles and abstracts to exclude studies that did not meet our inclusion criteria. During this phase, duplicates were identified and removed, leaving unique records for the next phase.
- **Eligibility Assessment:** The full texts of all remaining studies were examined to confirm that they fully adhered to our inclusion criteria. At this stage, studies were assessed in greater detail for specific methodological aspects and relevance to ensure they aligned with our research objectives.
- **Final Inclusion in Synthesis:** Only studies that met all the inclusion criteria were retained. This phase allowed us to focus on high-quality studies with empirical insights that would meaningfully contribute to our review.

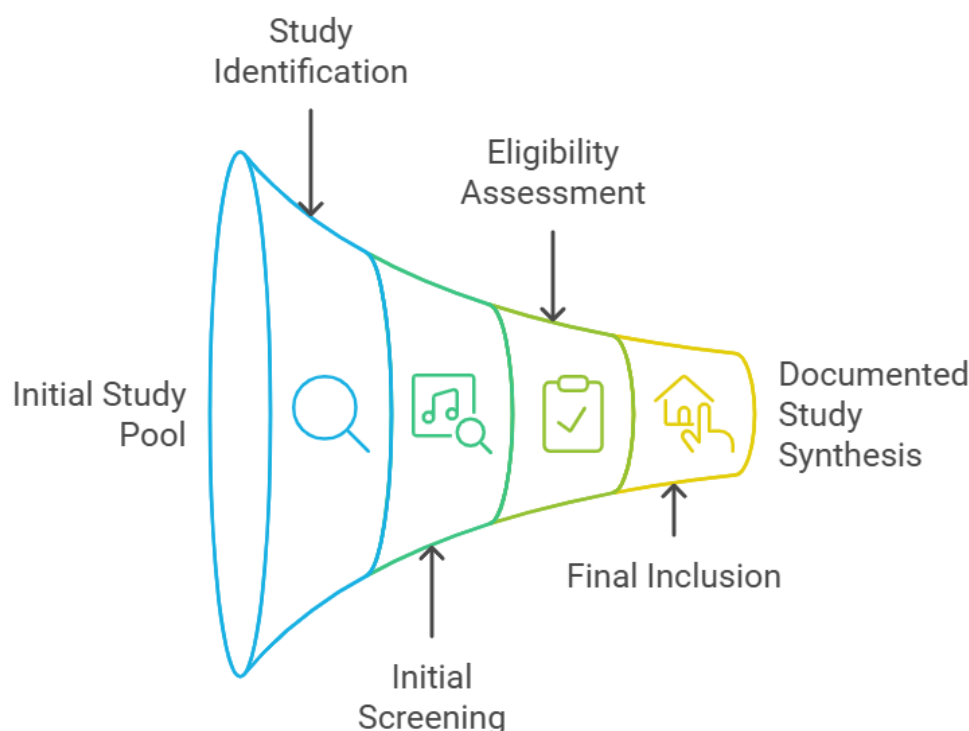


Figure 5. Evaluation phases of this study. Image produced with the assistance of Napkin AI.

By following these systematic steps, we ensured that our study selection process was focused on studies with the most relevance and reliability for our research objectives.

3.6. Data Extraction and Synthesis of Results

During this stage, we systematically extracted key data from each selected study to capture the primary findings. The synthesis of the results was then organized to align with our research objectives, providing a coherent analysis that could answer the research questions. The data synthesis was structured considering the following aspects:

- **Narrative Summary:** We provided an overview of each study to illustrate the overall trends and insights, highlighting key findings concerning the research questions.
- **Meta-Analysis:** We combined the findings via meta-analysis when quantitative data were sufficient and comparable, providing stronger evidence for certain patterns and trends.

This stage, illustrated in Figure 6, culminated in a structured synthesis that identified emerging patterns and potential gaps in current knowledge of SWRE.

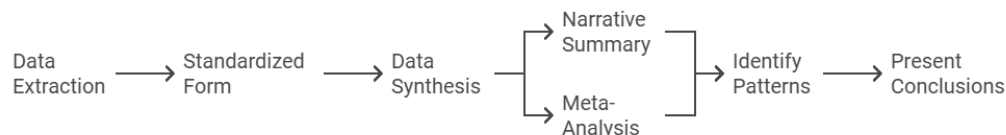


Figure 6. Synthesis process for data extraction and results. Image produced with Napkin AI assistance.

4. Results

For the search and selection process, the following query (adapted for each platform) was used to obtain the articles:

```

[Keywords: requirements elicitation] AND
[All: empirical] AND
[Full Text: software engineering processes] OR
[Full Text: challenges in software projects] OR
[Full Text: metamodel] OR
[Full Text: crosscutting activities]] AND
[E-Publication Date: (06/01/2019 TO 07/31/2024)]
  
```

Following the PRISMA guidelines (see Figure 7 for a simplified overview), the process was conducted in four steps:

1. The initial search retrieved a total of 1472 results: 492 from IEEE Xplore, 22 from the ACM Digital Library, 128 from SpringerLink, and 731 from MDPI.
2. Documents with 30% or more similarity were removed using a Python script to avoid duplicate or closely related studies.
3. Papers not directly related to “SW elicitation” or “SW requirements” were manually excluded (naturally, SW stands for software).
4. Finally, papers that did not report an empirical study (or a review of empirical experiences) on SWRE were excluded.

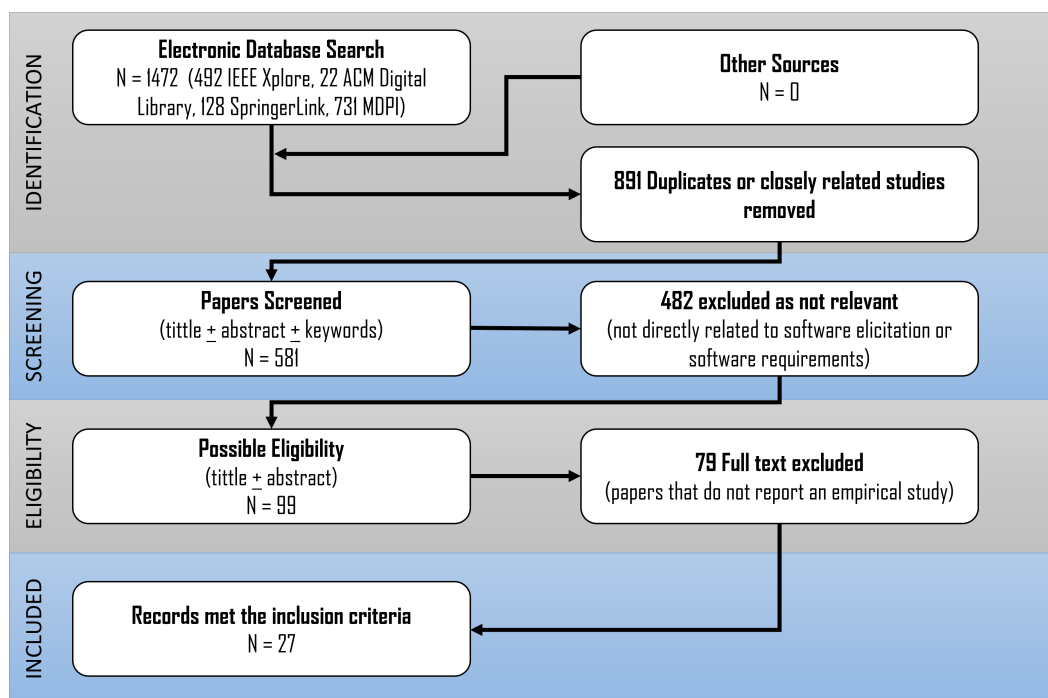


Figure 7. PRISMA simplified overview.

After completing these steps, a total of 27 papers were selected for inclusion in this study. A descriptive synthesis of these 27 articles is provided in Appendix A.

4.1. Crosscutting Steps and Activities in SWRE

To answer R.Q.1, this section synthesizes the findings of the SLR to identify the crosscutting steps and activities commonly present in SWRE processes. Crosscutting activities ensure consistency and adaptability throughout the RE process. These include documentation, team collaboration, iterative refinement, and the use of advanced tools, and are summarized as follows:

- **Documentation and Traceability:** Effective SWRE depends on well-structured documentation. Artifacts, such as UML diagrams and use cases, facilitate communication among teams and stakeholders [27,28].
- **Collaboration Across Teams:** In global software development, challenges such as cultural barriers and communication gaps necessitate structured frameworks for collaboration [29,30].
- **Iterative Refinement:** Most studies emphasize the iterative nature of SWRE. Requirements are revisited in cycles to accommodate evolving stakeholder needs and changes in project contexts [31,32].
- **Leveraging Technology and Tools:** Modern SWRE integrates advanced tools such as natural language processing for requirements analysis and domain-specific languages for rapid prototyping [33,34].

4.1.1. Quantitative Analysis

To further support these findings, a quantitative analysis of the selected studies was conducted. Figure 8 illustrates the distribution of elicitation methods used across the 27 selected papers.

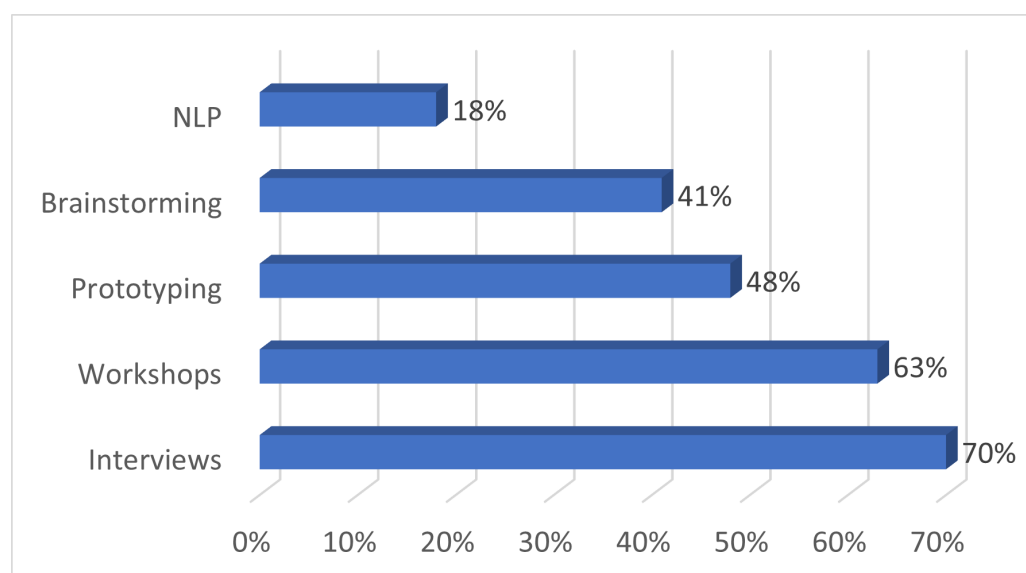


Figure 8. Distribution of elicitation methods across the selected studies.

The analysis shows that interviews and workshops are the most commonly cited methods, referenced in 70% and 63% of the studies, respectively. Techniques such as prototyping and brainstorming follow, being referenced in 48% and 41% of the papers. This quantitative breakdown highlights the prevalence of traditional methods while also identifying a growing interest in automated approaches, such as natural language processing (NLP), which

is referenced in 18% of the studies. These findings underline the importance of balancing traditional and modern techniques in diverse SW project contexts.

4.1.2. Analysis of Temporal Complexity and Scalability

In this study, the elicitation methods were evaluated from a qualitative perspective. However, quantitative metrics such as temporal complexity and scalability should also be considered when evaluating and comparing the various methods. These metrics provide a clearer view of the viability of each method in large-scale scenarios or when dealing with large teams or complex projects. Descriptions of how these aspects could be measured and compared for the methods studied are as follows:

- **Temporal Complexity:** This refers to the time required to carry out the requirements elicitation activities, such as interviews, workshops, or surveys. It can be measured by the total execution time for each technique or by evaluating the amount of resources (human and material) needed. Specific time units (minutes, hours) should be used for each task within the method.
- **Scalability:** This metric measures how the performance of the elicitation method changes as the project size or team size increases. Methods such as interviews may not scale well with large teams, while automated surveys or online collaboration platforms may offer better scalability. Techniques can be compared based on their ability to handle an increasing number of participants or requirements efficiently.

Example Metrics:

- For interviews:
 - Time per interview: 30 min;
 - Number of interviews needed for a team with five members: 10;
 - Total time spent: 5 h.
- For surveys:
 - Time to complete a survey: 15 min;
 - Number of surveys needed for 100 participants: 100;
 - Total time spent: 25 h.

4.2. Challenges and Limitations in SWRE

To answer R.Q.2, the SWRE process faces challenges like ambiguity and limited stakeholder involvement but also presents opportunities, such as enhancing explainability and integrating standards for scalability. Key challenges include the following:

- **Ambiguity and Miscommunication:** Natural languages dominate, leading to inconsistent interpretations [35].
- **Customer Involvement:** Limited technical knowledge among stakeholders can hinder effective elicitation [36,37].

Additionally, emerging opportunities for SWRE include the following:

- **Explainability as a Requirement:** Emphasizing explainability enhances trust in complex systems like ML models [38].
- **Integration of Standards:** Lightweight adaptations of standards ensures scalability for smaller organizations [39].

4.3. Section Summary

SWRE is a dynamic, iterative process that combines structured methods, modern tools, and continuous stakeholder collaboration to address challenges and drive innovation in software development practices. The crosscutting steps and activities in SWRE revolve

around structured stakeholder engagement, iterative refinement, effective documentation, and leveraging modern tools. While challenges such as ambiguity and limited customer involvement persist, opportunities for innovation—such as integrating explainability and adopting standards—highlight promising directions for future research and practice.

5. A Metamodel for Requirements Elicitation Processes

Considering R.Q.3, this section presents a metamodel that defines the essential elements and relationships within SWRE processes. The proposed metamodel synthesizes iterative interactions, stakeholder engagement, and artifact refinement to address the complexities of modern software development.

5.1. Elements of the Metamodel

The core components of the RE process and their roles are defined as follows:

1. **Stakeholders:** Stakeholders, including end users, clients, and technical teams, are the primary drivers of requirements. Their continuous involvement ensures that requirements reflect real-world needs and constraints.
2. **Requirements:** Requirements are categorized as follows:
 - **Functional Requirements:** Define specific system functionalities.
 - **Non-Functional Requirements:** Address quality attributes like performance, security, and usability.
 - **Domain-Specific Requirements:** Tailored to specific industries or applications [32,40].
3. **Techniques and Tools:** A range of techniques (e.g., interviews, workshops, and prototyping) and tools (e.g., domain-specific languages and UML) support the elicitation process by structuring and validating requirements [27,41].
4. **Artifacts:** Artifacts, such as use cases, sequence diagrams, and prototypes, serve as outputs of the elicitation process and are iteratively refined to align with evolving requirements [39].
5. **Contextual Factors:** Project size, regulatory requirements, team distribution, and other factors shape the choice of techniques, prioritization, and validation efforts [29,42].

Considering the quality of refined artifacts, such as requirements documents, user stories, or models, qualitative evaluations provide valuable insights. However, quantitative metrics are also essential for objectively assessing the quality of these artifacts. To address this, we propose introducing similarity measures and convergence rates as objective metrics for evaluating the refinement process. These metrics are defined as follows:

- **Similarity Measures:** These metrics evaluate the degree to which the refined artifacts align with the original requirements [43]. For instance, cosine similarity can be applied to textual documents or models by comparing the frequency of terms, key concepts, or semantic structures between the original and refined artifacts. Higher similarity scores indicate that the refinement process has successfully preserved the core elements of the original requirements. The formula for cosine similarity is as follows:

$$\text{Similarity}(A, B) = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}}$$

where A_i and B_i are the components of the original and refined artifacts, respectively. Example: If the similarity between the original requirements document and the refined version is calculated as 0.85, it indicates that 85% of the terms and concepts are retained through the refinement process.

- **Convergence Rates:** The convergence rate quantifies how quickly the refinement process reaches a stable state, where subsequent changes to the artifacts become negli-

gible [44]. This can be assessed by tracking the number of revisions or modifications made to the artifacts over time. A faster convergence rate indicates an efficient refinement process, while a slower rate may highlight areas requiring additional attention. The formula for calculating the convergence rate is as follows:

$$\text{Convergence Rate} = \frac{\text{Number of Revisions}}{\text{Total Time Period}}$$

Example: A convergence rate of 0.2 revisions per day suggests that the refinement process is relatively slow and may require further optimization.

These objective metrics offer a quantitative basis for evaluating the quality and efficiency of the refinement process, complementing qualitative assessments and providing a more comprehensive view of the artifact quality.

5.2. Relationships in the Metamodel

The metamodel establishes relationships between stakeholders, requirements, and artifacts as follows:

1. Stakeholders interact with techniques to provide inputs.
2. Requirements guide the creation of artifacts using selected techniques and tools.
3. Contextual factors influence prioritization, stakeholder roles, and method selection.

5.3. Iterative Process Representation

Iterative process representation describes the cyclical nature of elicitation, validation, and refinement. The metamodel is inherently iterative, incorporating these cycles:

1. Elicitation: Using techniques to gather requirements from stakeholders.
2. Validation and Refinement: Iteratively updating and improving requirements and artifacts based on feedback.
3. Adaptation: Adjusting techniques and priorities to align with contextual changes.

The iterative nature of requirements elicitation, validation, and refinement can be mathematically represented using a Markov chain model. This model captures the probabilities of transitioning between the key phases of the elicitation process. The states of the Markov chain correspond to the primary activities (elicitation, validation, and refinement), while the transition probabilities reflect the likelihood of moving from one phase to another.

Let $S = \{E, V, R\}$ represent the states, where, E stands for elicitation, V for validation, and R for refinement. The transition probabilities between these states are given by the stochastic matrix P :

$$P = \begin{bmatrix} p_{EE} & p_{EV} & p_{ER} \\ p_{VE} & p_{VV} & p_{VR} \\ p_{RE} & p_{RV} & p_{RR} \end{bmatrix}$$

Each element p_{ij} represents the probability of transitioning from state i to state j . For example, p_{EV} denotes the probability of moving from elicitation to validation. Of course, the sum of probabilities in each row must equal 1, ensuring that the model adheres to the properties of a Markov chain. For illustration, an example transition matrix could be

$$P = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.6 & 0.1 \\ 0.1 & 0.4 & 0.5 \end{bmatrix}$$

This matrix suggests that the process is most likely to remain in the same phase, but transitions to other phases occur with smaller probabilities, reflecting the iterative

and feedback-driven nature of requirements elicitation. Figure 9 provides a state diagram visualizing the transitions between phases based on this example.

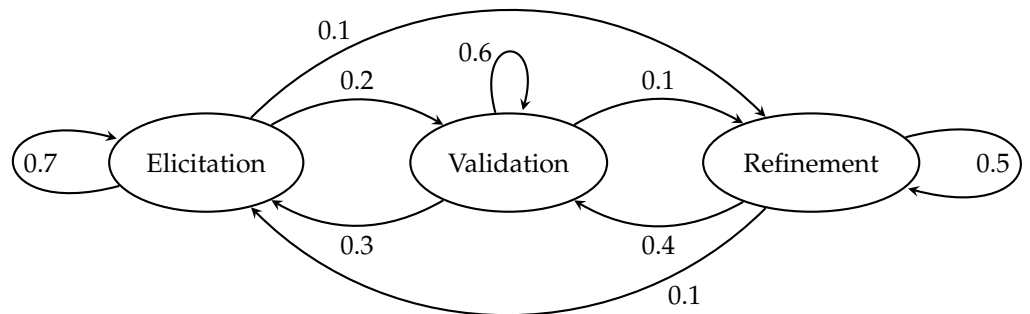


Figure 9. Example of a state diagram of the Markov model for iterative requirements elicitation.

5.4. Metamodel Representation and Advantages

Figure 10 visualizes the proposed metamodel and its iterative relationships.

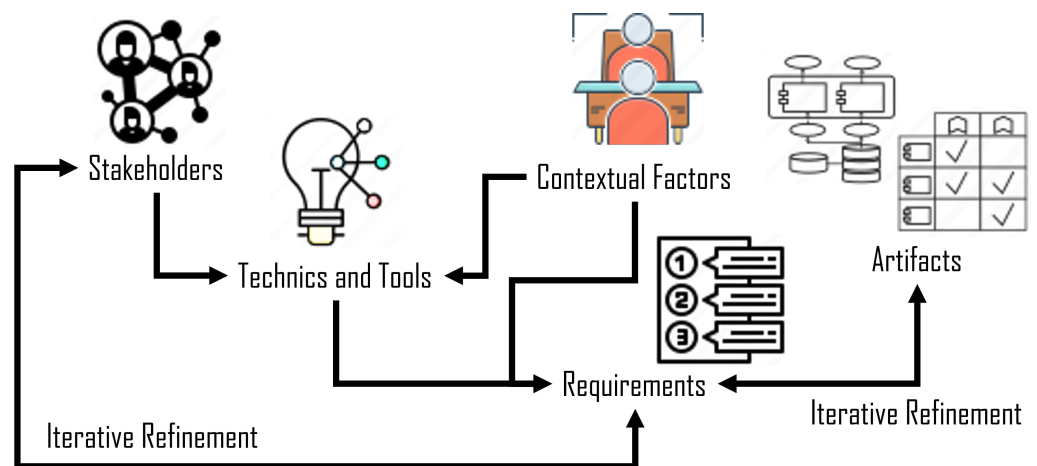


Figure 10. Software requirements elicitation process metamodel.

The metamodel encapsulates the iterative and adaptive nature of RE via the integration of stakeholders, contextual factors, and iterative refinement, providing a robust framework to navigate the complexities of software development. The advantages of the metamodel are as follows:

1. **Adaptability:** Suitable for varying project contexts, from small- to large-scale developments [39].
2. **Flexibility:** Incorporates feedback loops to address dynamic stakeholder needs [31].
3. **Traceability:** Links requirements to artifacts, improving communication and alignment between teams [38].

6. Metamodel Validation and Refinement

The proposed SWRE process metamodel was reviewed by six domain experts with diverse backgrounds in SWE and requirements analysis. Each expert provided detailed feedback based on their experience and understanding of the elicitation process. Their evaluations are summarized as follows:

- Expert 1—Senior Requirements Engineer (15 years of experience):
 1. Highlighted the clear separation of activities, such as stakeholder identification, requirements gathering, and validation, within the metamodel.

2. Suggested emphasizing feedback loops to address iterative requirements refinement in agile projects.
- Expert 2—Academic Researcher in Requirements Engineering (10+ published papers):
 1. Praised the comprehensive representation of key stages in the elicitation process.
 2. Recommended expanding the representation of tools and techniques, such as interviews and prototyping, within the diagram.
 - Expert 3—Project Manager in Software Development (12 years of experience):
 1. Commended the inclusion of stakeholder management as a central element.
 2. Suggested clarifying the handoff between RE and system modeling to ensure seamless integration.
 - Expert 4: Business Analyst with a focus on Enterprise Software (8 years of experience):
 1. Appreciated the alignment of the metamodel with industry-standard practices.
 2. Proposed adding a decision-making node to highlight prioritization and trade-off analysis in elicitation.
 - Expert 5—Agile Coach and Scrum Master (7 years of experience):
 1. Recognized the adaptability of the metamodel for agile frameworks.
 2. Suggested that iterative interactions between stakeholders and development teams could be more prominently visualized.
 - Expert 6—Software Architect (8 years of experience):
 1. Endorsed the modular structure of the metamodel, which facilitates customization for specific project contexts.
 2. Recommended incorporating risk assessment and mitigation strategies during RE.

These expert reviews validate the clarity, comprehensiveness, and adaptability of the metamodel, emphasizing its alignment with industry standards and its suitability for diverse development contexts, including agile and enterprise-level projects.

6.1. Strengths Identified

The following key strengths were consistently highlighted by the experts:

- A clear structure that separates essential activities, such as stakeholder management, requirements gathering, and validation.
- A comprehensive representation of stages within the elicitation process, ensuring applicability across various methodologies.
- A modular and adaptable design that allows customization to meet the specific needs of agile and enterprise environments.

6.2. Common Recommendations

The experts provided several aligned suggestions to further enhance the utility and practicality of the metamodel:

- Incorporate feedback loops: Emphasize iterative refinement processes to reflect the dynamic nature of agile projects.
- Expand on tools and techniques: Add explicit representations of elicitation techniques, such as interviews, workshops, and prototyping.
- Highlight risk management: Integrate risk assessment and mitigation strategies to address uncertainties during RE.
- Clarify stage transitions: Define the handoff between RE and system modeling to ensure seamless integration.

- Include decision-making nodes: Add elements to address prioritization and trade-offs, reflecting real-world challenges in requirements analysis.

6.3. Refined Metamodel

The synthesis of the strengths and recommendations provided a roadmap for refining the metamodel, ensuring it is both practical and aligned with modern SWE practices. The refined metamodel is illustrated in Figure 11.

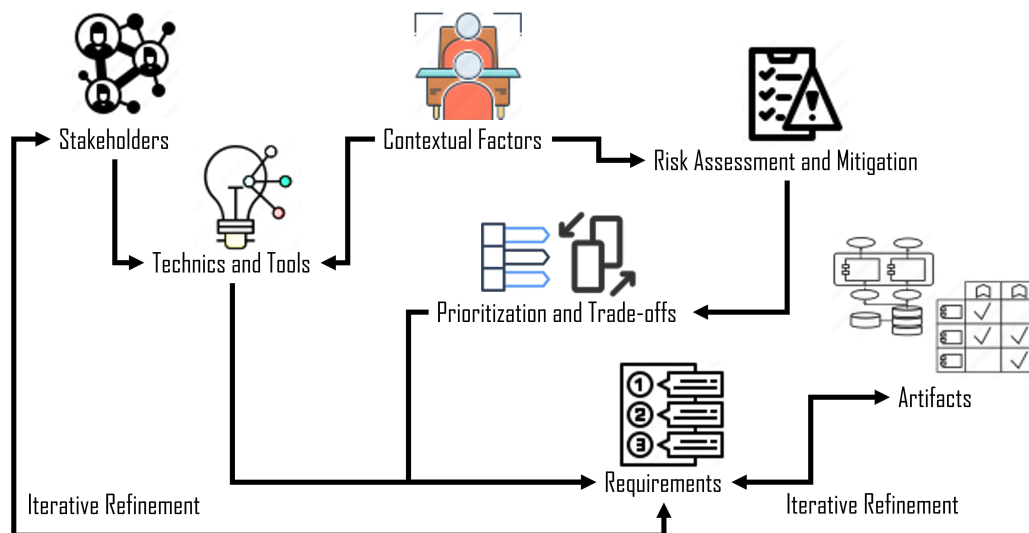


Figure 11. Software requirements elicitation process refined metamodel.

In summary, the improvements made were as follows:

- Addition of Key Nodes:
 - **Prioritization and Trade-offs:** Highlights decision making during elicitation.
 - **Risk Assessment and Mitigation:** Emphasizes risk identification and management.
- Improvements in Connectivity: Contextual factors now influence both techniques and risk assessment, ensuring external factors impact elicitation and risk management processes.
- Enhanced Visual Clarity: The iterative refinement process remains, linking back to artifacts for continuous refinement.

7. Threats to Validity and Limitations

As with any systematic research, this study faces threats to validity that may impact the interpretation and generalization of its findings. The primary threats identified and the strategies adopted to mitigate them are described below.

7.1. Internal Validity

Internal validity pertains to the accuracy with which the data and analysis address the research questions. The following risks were identified:

- **Selection bias:** The inclusion of articles may have been limited by the search, selection, and exclusion criteria. This risk was mitigated by conducting an exhaustive search across recognized databases using well-defined keywords and a transparent review protocol.
- **Data extraction errors:** Subjective interpretation during data extraction may introduce bias. We minimized this risk using standard templates for data extraction and analysis.

7.2. Construct Validity

Terms such as “metamodel” and “crosscutting activities” may be interpreted differently by other researchers or practitioners, affecting the replicability of the study. This risk was addressed by providing clear and consistent definitions throughout this study.

7.3. Limitations

The generalization of the findings to other domains presents certain challenges:

- **Limited generalization:** The study focuses on works published in English and within a specific time frame (2019–2024), which, while ensuring the currency of findings, may exclude relevant perspectives from other regions or periods. This limitation could affect the applicability of the results in different cultural or historical contexts and potentially overlook seminal works or prior relevant studies.
- **Domain-specific context:** Requirements engineering techniques and challenges may vary across industries or software types. Our metamodel might require adaptation for use in highly specialized sectors, such as embedded systems or critical software.

7.4. Mitigation Strategies

The following measures were employed to address these threats:

- **Systematic protocols:** A protocol based on recognized standards, such as PRISMA guidelines, was followed to ensure transparency and rigor.
- **Expert review:** The metamodel and study findings were validated by six experts with diverse backgrounds in SWE and requirements analysis.
- **Comprehensive documentation:** Detailed descriptions of the search, selection, and analysis processes were provided to ensure traceability and enable future replication.

Despite the aforementioned limitations, the measures adopted strengthen the validity of the findings and their relevance to practice in software engineering.

8. Discussion

This study highlights the multifaceted nature of RE processes within SWE, emphasizing both their complexities and potential for improvement. The following discussion addresses the implications of the metamodel, the identified challenges, and avenues for future research and practice.

8.1. Implications of the Proposed Metamodel

The proposed refined metamodel consolidates iterative improvement, stakeholder engagement, and the integration of contextual factors into a cohesive framework. This structure provides practical guidance for managing the inherent complexities of RE. By accommodating diverse project contexts and stakeholder needs, the metamodel offers a flexible approach that can be tailored to both small-scale projects and large, distributed teams. Its adaptability is particularly valuable for agile environments, where iterative development and frequent stakeholder feedback are essential.

8.2. Sociocultural and Human-Centered Challenges

Differences in cultural norms, communication styles, and stakeholder priorities can significantly impact the RE process [45]. For example, in culturally diverse teams, variations in expectations or decision-making processes may lead to misunderstandings or conflicting requirements. Also, human-centered challenges, such as resistance to change or varying levels of technological literacy among stakeholders, can further complicate the RE process.

8.3. Challenges in Current Practices

Despite its strengths, this study reveals persistent challenges that hinder effective RE:

- **Ambiguity in stakeholder communication:** Natural language remains the dominant medium for capturing requirements [46], which introduces risks of both misinterpretation and inconsistency.
- **Limited stakeholder involvement:** The active engagement of stakeholders throughout the elicitation process is often constrained by their availability, technical expertise, or understanding of the scope of the system [47].
- **Scalability of methods:** Techniques suitable for small projects may fail to address the complexities of large-scale or highly specialized systems, necessitating further refinement or hybrid approaches [48].

These challenges underscore the need for enhanced techniques, such as leveraging advanced tools like natural language processing (NLP) and domain-specific modeling, to streamline the RE process and mitigate risks. Additionally, modern technologies such as artificial intelligence (AI) offer promising opportunities to address them. Some examples are as follows:

- **AI-driven requirements extraction:** AI models trained on large datasets can automatically identify and classify requirements from unstructured text, reducing ambiguity and improving accuracy in requirements documentation.
- **NLP-enhanced communication:** NLP-based tools can facilitate stakeholder communication by translating technical jargon into accessible language or summarizing lengthy discussions into actionable requirements.
- **Domain-specific modeling tools:** These tools can provide tailored frameworks for capturing requirements in highly specialized fields, such as healthcare or finance, where standard methods may be insufficient.

The integration of these technologies into the requirements elicitation process could significantly improve its efficiency, scalability, and adaptability, particularly in complex or distributed project environments.

8.4. Limitations of this Study

While this study provides a comprehensive framework, its findings are bounded by certain limitations:

- The focus on empirical studies from 2019 to 2024 may exclude relevant but older or unpublished insights.
- The metamodel was validated through expert reviews, which, while valuable, may benefit from broader industry testing and feedback.

Addressing these limitations in future research will help refine the metamodel and expand its applicability across diverse software development contexts.

9. Conclusions

This study addresses SWRE process challenges by introducing a comprehensive metamodel integrating iterative refinement, stakeholder engagement, and contextual adaptability. The findings provide a structured approach to navigating the complexities inherent in capturing and managing software requirements, offering valuable insights for both researchers and practitioners.

The proposed metamodel highlights the importance of balancing technical precision with effective communication among diverse stakeholders. Its iterative nature ensures adaptability to evolving project needs, while its emphasis on contextual factors enhances its applicability across varied software development environments, including agile and

large-scale distributed projects. The validation by domain experts underscores its practical relevance and potential for widespread adoption.

This work enhances the phases of the SWRE process by integrating best practices and identifying key opportunities for innovation and enhancement. By fostering a deeper understanding of the elicitation process, it provides a pathway for improving the alignment between software systems and stakeholder expectations, ultimately advancing the quality and success of SWE projects.

Future Work

Future research could benefit from the incorporation of statistical analyses to further validate and examine the adoption and effectiveness of elicitation techniques over time. Specifically, the use of Chi-square tests and regression analysis could reveal significant trends in the adoption of techniques across different periods and their relationship with improved project outcomes. In particular, the following approaches are recommended for further analysis and validation of the techniques:

- **Chi-Square Test for Trend Analysis:** This test can be used to assess whether the adoption rates of specific elicitation techniques have changed significantly over time. By creating contingency tables that categorize techniques based on periods, the Chi-square test can identify patterns in the adoption of techniques and determine if certain methods are gaining or losing popularity.
- **Regression Analysis for Effectiveness Evaluation:** Regression analysis can measure how specific techniques correlate with improved requirements clarity, stakeholder satisfaction, or project success. This will allow researchers to pinpoint the most effective techniques and determine their contribution to project success.
- **Industry-Wide Validation and Real-World Applications:** Expanding the validation of the proposed metamodel to include practical implementations in real-world settings would enhance its generalizability and practical relevance. In fact, pilot projects across diverse industrial sectors, such as healthcare, finance, or education, could test the metamodel's adaptability to domain-specific requirements challenges. Case studies or longitudinal analyses in these contexts could provide actionable insights into how the metamodel performs in terms of scalability, effectiveness, and stakeholder satisfaction.

Incorporating these statistical methods, alongside broader industry validation, will provide a more comprehensive understanding of the trends, factors, and practical applications influencing the adoption and effectiveness of elicitation techniques. This integrated approach will help refine the metamodel and ensure its applicability across diverse software development contexts.

Author Contributions: All authors contributed to the current work. Conceptualization, M.H. and R.P.; methodology, M.H. and F.Y.; validation, M.H. and J.F.; formal analysis, R.P. and F.Y.; investigation, M.H.; data curation, M.H. and R.P.; writing—original draft preparation, M.H.; writing—review and editing, R.P.; visualization; supervision, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All the research data are provided within the article.

Acknowledgments: The authors would like to express their gratitude to the experts who generously shared their time and insights during the validation of the proposed metamodel.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations have been used in this manuscript:

SW	Software
SWE	Software Engineering
SWRE	Software Requirements Elicitation
SRL	Systematic Literature Review

Appendix A. Descriptive Synthesis

Table A1 summarizes the 27 selected articles, which span academic and industrial contexts.

Table A1. Descriptive synthesis.

Citation	Synthesis
[49]	Explores requirements elicitation (RE) methods for Virtual Reality (VR). Categorizes them into widely used, focused, customized, and unique approaches, emphasizing their adoption in academia and industry.
[50]	Describes a tool for creative RE based on combinatorial creativity. Validated through case studies, it supports innovative requirements generation using multimodal stimuli.
[51]	Investigates functional RE techniques in the IoT. Highlights the prevalence of scenario-based models and UML, addressing IoT-specific challenges and limitations of traditional RE methods.
[37]	Proposes a conceptual model to mitigate communication and coordination challenges in global software development (GSD) during requirements change management (RCM). Validated through expert surveys.
[30]	Identifies 62 communication and 14 coordination challenges in RCM for GSD. Proposes and validates mitigation strategies to address these challenges.
[27]	Categorizes design thinking techniques for RE. Offers comparative tables to assist software engineers in selecting the most suitable technique for specific project goals.
[52]	Surveys practitioners' experiences with RE techniques. Highlights common challenges and evaluates the utility of various techniques, offering insights into their practical application.
[31]	Mapping study on requirements engineering methods for VR software development. Discusses specific challenges and proposes RE methods tailored to VR's unique requirements.
[41]	Explores the intersection of blockchain technology with software requirements (SWRE) engineering, presenting a blockchain-based SWRE model to improve transparency, security, and traceability.
[36]	Proposes a tacit knowledge-based RE model, addressing challenges in eliciting implicit requirements, particularly in the COVID-19 context, through a systematic review.
[53]	Introduces a UML-based performance evaluation method for real-time systems using Timed Petri Nets, leveraging model-driven engineering to enhance early-stage system performance evaluation.

Table A1. Cont.

Citation	Synthesis
[54]	Presents an RE process for developing an assistance system to manage complexity in small and medium-sized enterprises (SMEs) during COVID-19. Discusses challenges and proposes remote elicitation techniques.
[28]	Introduces a metamodel for evaluating Industry 4.0 readiness in enterprises. Integrates existing maturity models and readiness indexes to identify gaps and potential research areas.
[55]	Explores the use of ArchiMate for visualizing business strategies in SMEs, addressing alignment challenges between business and IT strategies. Offers insights into strategic planning and enterprise architecture modeling.
[33]	Proposes a method merging design science research with innovation processes for methodic creation and reuse of information systems artifacts. Validated through case studies, including automated conversational interfaces.
[56]	Analyzes deep learning techniques for anomaly detection and failure prediction in industrial use cases, emphasizing the importance of explainable AI (XAI) and model deployment via the RAI4.0 metamodel.
[57]	Proposes the Wide Intelligent Management Architecture model to integrate process, knowledge, and transactional systems. Demonstrates its applicability through a case study in acrylic fiber production.
[58]	Reviews AI-enabled methods for chemical process optimization in Industry 4.0. Highlights machine learning (ML) and deep learning applications, particularly in molecular design and synthetic route planning.
[29]	Identifies 31 challenges in RCM within GSD, emphasizing the need for better communication, coordination, and structured RCM models.
[32]	Reviews challenges in requirements engineering for ML projects. Suggests adjustments to traditional RE practices to align with the iterative and data-driven nature of ML systems.
[40]	Explores the relationship between similar requirements and software similarity using natural language processing models like BERT. Demonstrates the potential of recommender systems for code reuse in the railway domain.
[38]	Proposes artifacts like conceptual and reference models to incorporate explainability as a non-functional requirement. Highlights its impacts on transparency, usability, and trustworthiness in systems.
[39]	Implements a lightweight adaptation of ISO/IEC/IEEE 15939:2017 to measure and improve requirements elicitation in small-sized software organizations. Focuses on quality attributes and iterative refinement.
[42]	Proposes a hybrid technique combining minimal spanning tree and AHP for prioritizing functional requirements in parallel development projects. Demonstrates scalability and reduced delays in ODOO ERP.
[59]	Develops a cost-effective RE model tailored for limited customer engagement scenarios. Emphasizes systematic elicitation, domain knowledge, and validation through expert reviews.
[34]	Proposes a model-based engineering approach for developing a hospital situational awareness system. Relies on domain-specific modeling to support rapid prototyping and decision making during the COVID-19 pandemic.
[35]	Presents survey results from 84 practitioners to understand challenges and practices in requirements engineering. Emphasizes practitioners' reliance on natural languages and issues with cost, errors, and customer involvement.

References

1. Davis, A.; Dieste, O.; Hickey, A.; Juristo, N.; Moreno, A.M. Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review. In Proceedings of the 14th IEEE International Requirements Engineering Conference (RE'06), Minneapolis/St. Paul, MN, USA, 11–15 September 2006; pp. 179–188. [\[CrossRef\]](#)
2. Yeazitis, T.; Weger, K.; Mesmer, B.; Clerkin, J.; Van Bossuyt, D. Biases in Stakeholder Elicitation as a Precursor to the Systems Architecting Process. *Systems* **2023**, *11*, 499. [\[CrossRef\]](#)
3. Wiegers, K.E.; Beatty, J. *Software Requirements 3*; Microsoft Press: Redmond, WA, USA, 2013.
4. Alavi, M.; Leidner, D. Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Q.* **2001**, *1*, 107–136. [\[CrossRef\]](#)
5. Hoblos, N.; Sandeep, M.; Pan, S.L. Achieving stakeholder alignment in digital transformation: A frame transformation perspective. *J. Inf. Technol.* **2023**, *39*, 630–649. [\[CrossRef\]](#)
6. Kober, C.; Medina, F.G.; Benfer, M.; Wulfsberg, J.P.; Martinez, V.; Lanza, G. Digital Twin Stakeholder Communication: Characteristics, Challenges, and Best Practices. *Comput. Ind.* **2024**, *161*, 104135. [\[CrossRef\]](#)
7. Pacheco, C.; García, I.; Reyes, M. Requirements elicitation techniques: A systematic literature review based on the maturity of the techniques. *IET Softw.* **2018**, *12*, 365–378. [\[CrossRef\]](#)
8. Cheng, B.H.; Atlee, J.M. Research Directions in Requirements Engineering. In Proceedings of the Future of Software Engineering (FOSE '07), Minneapolis, MN, USA, 23–25 May 2007; pp. 285–303. [\[CrossRef\]](#)
9. Lim, S.; Henriksson, A.; Zdravkovic, J. Data-Driven Requirements Elicitation: A Systematic Literature Review. *SN Comput. Sci.* **2021**, *2*, 16. [\[CrossRef\]](#)
10. Guelfi, N. The MESSIR Flexible Scientific Approach to Requirements Engineering. *Software* **2022**, *1*, 80–106. [\[CrossRef\]](#)
11. Kitchenham, B.; Brereton, P. A systematic review of systematic review process research in software engineering. *Inf. Softw. Technol.* **2013**, *55*, 2049–2075. [\[CrossRef\]](#)
12. Zowghi, D.; Coulin, C. Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In *Engineering and Managing Software Requirements*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 19–46. [\[CrossRef\]](#)
13. Canché, M.; Pino, J. Requirements Elicitation for Collaborative Systems: A Systematic Review. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; pp. 297–304. [\[CrossRef\]](#)
14. Meth, H.; Brhel, M.; Maedche, A. The state of the art in automated requirements elicitation. *Inf. Softw. Technol.* **2013**, *55*, 1695–1709. [\[CrossRef\]](#)
15. Cheliger, C.; Huang, J.; Wu, G.; Bhuiyan, N.; Xu, Y.; Zeng, Y. Machine learning in requirements elicitation: A literature review. *Artif. Intell. Eng. Des. Anal. Manuf.* **2022**, *36*, e32. [\[CrossRef\]](#)
16. Sommerville, I. *Software Engineering*, 10th ed.; Pearson Education Limited: Boston, MA, USA, 2016.
17. Hickey, A.; Davis, A. A Unified Model of Requirements Elicitation. *J. Manag. Inf. Syst.* **2004**, *20*, 65–84. [\[CrossRef\]](#)
18. Carrizo, D.; Ortiz, C. Models of requirements elicitation process: A systematic mapping. *Ing. Desarro.* **2016**, *34*, 184–203. [\[CrossRef\]](#)
19. Menezes, T. A Review to Find Elicitation Methods for Business Process Automation Software. *Software* **2023**, *2*, 177–196. [\[CrossRef\]](#)
20. Lim, S.L.; Finkelstein, A. StakeRare: Using Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation. *IEEE Trans. Softw. Eng.* **2012**, *38*, 707–735. [\[CrossRef\]](#)
21. Kitchenham, B. Procedures for performing systematic reviews. *Keele UK Keele Univ.* **2004**, *33*, 1–26.
22. Siddaway, A.P.; Wood, A.; Hedges, L. How to Do a Systematic Review: A Best Practice Guide for Conducting and Reporting Narrative Reviews, Meta-Analyses, and Meta-Syntheses. *Annu. Rev. Psychol.* **2019**, *70*, 747–770. [\[CrossRef\]](#)
23. Linnenluecke, M.; Marrone, M.; Singh, A.K. Conducting systematic literature reviews and bibliometric analyses. *Aust. J. Manag.* **2019**, *45*, 175–194. [\[CrossRef\]](#)
24. Gupta, S.; Rajiah, P.; Middlebrooks, E.; Baruah, D.; Carter, B.; Burton, K.; Chatterjee, A.; Miller, M.M. Systematic Review of the Literature: Best Practices. *Acad. Radiol.* **2018**, *25*, 1481–1490. [\[CrossRef\]](#)
25. Odeh, Y.; Al-Saiyd, N. Prioritizing Use Cases: A Systematic Literature Review. *Computers* **2023**, *12*, 136. [\[CrossRef\]](#)
26. Hoy, Z.; Xu, M. Agile Software Requirements Engineering Challenges-Solutions—A Conceptual Framework from Systematic Literature Review. *Information* **2023**, *14*, 322. [\[CrossRef\]](#)
27. Meireles, M.; Souza, A.; Conte, T.; Maldonado, J. Organizing the Design Thinking Toolbox: Supporting the Requirements Elicitation Decision Making. In Proceedings of the Brazilian Symposium on Software Engineering, Joinville, Brazil, 27 September–1 October 2021; pp. 285–290. [\[CrossRef\]](#)
28. Basl, J.; Doucek, P. A Metamodel for Evaluating Enterprise Readiness in the Context of Industry 4.0. *Information* **2019**, *10*, 89. [\[CrossRef\]](#)

29. Akbar, M.A.; Naveed, W.; Alsanad, A.A.; Alsuwaidan, L.; Alsanad, A.; Gumaedi, A.; Shafiq, M.; Riaz, M.T. Requirements Change Management Challenges of Global Software Development: An Empirical Investigation. *IEEE Access* **2020**, *8*, 203070–203085. [[CrossRef](#)]
30. Qureshi, S.; Khan, S.U.R.; Iqbal, J.; Inayat-Ur-Rehman. A Study on Mitigating the Communication and Coordination Challenges During Requirements Change Management in Global Software Development. *IEEE Access* **2021**, *9*, 88217–88232. [[CrossRef](#)]
31. Karre, S.A.; Reddy, Y.R.; Mittal, R. RE Methods for Virtual Reality Software Product Development: A Mapping Study. *ACM Trans. Softw. Eng. Methodol.* **2024**, *33*, 88. [[CrossRef](#)]
32. Gjorgjevikj, A.; Mishev, K.; Antovski, L.; Trajanov, D. Requirements Engineering in Machine Learning Projects. *IEEE Access* **2023**, *11*, 72186–72208. [[CrossRef](#)]
33. Huseynli, M.; Bub, U.; Ogbuachi, M.C. Development of a Method for the Engineering of Digital Innovation Using Design Science Research. *Information* **2022**, *13*, 573. [[CrossRef](#)]
34. Shaked, A. Modeling for Rapid Systems Prototyping: Hospital Situational Awareness System Design. *Systems* **2021**, *9*, 12. [[CrossRef](#)]
35. Ozkaya, M.; Akdur, D.; Toptani, E.C.; Kocak, B.; Kardas, G. Practitioners' Perspectives towards Requirements Engineering: A Survey. *Systems* **2023**, *11*, 65. [[CrossRef](#)]
36. Anwar, H.; Khan, S.U.R.; Iqbal, J.; Akhunzada, A. A Tacit-Knowledge-Based Requirements Elicitation Model Supporting COVID-19 Context. *IEEE Access* **2022**, *10*, 24481–24508. [[CrossRef](#)]
37. Qureshi, S.; Khan, S.U.R.; Inayat-Ur-Rehman; Javed, Y.; Saleem, S.; Iqbal, A. A Conceptual Model to Address the Communication and Coordination Challenges During Requirements Change Management in Global Software Development. *IEEE Access* **2021**, *9*, 102290–102305. [[CrossRef](#)]
38. Chazette, L.; Brunotte, W.; Speith, T. Explainable software systems: From requirements analysis to system evaluation. *Requir. Eng.* **2022**, *27*, 457–487. [[CrossRef](#)]
39. Pacheco, C.; Garcia, I.; Calvo-Manzano, J.A.; Reyes, M. Measuring and improving software requirements elicitation in a small-sized software organization: A lightweight implementation of ISO/IEC/IEEE 15939:2017—Systems and software engineering—Measurement process. *Requir. Eng.* **2023**, *28*, 257–281. [[CrossRef](#)]
40. Abbas, M.; Ferrari, A.; Shatnawi, A.; Enoiu, E.; Saadatmand, M.; Sundmark, D. On the Relationship Between Similar Requirements and Similar Software: A Case Study in the Railway Domain. *Requir. Eng.* **2023**, *28*, 23–47. [[CrossRef](#)]
41. Farooq, M.S.; Ahmed, M.; Emran, M. A Survey on Blockchain Acquainted Software Requirements Engineering: Model, Opportunities, Challenges, and Future Directions. *IEEE Access* **2022**, *10*, 48193–48228. [[CrossRef](#)]
42. Yaseen, M.; Mustapha, A.; Shah, M.A.; Ibrahim, N. A hybrid technique using minimal spanning tree and analytic hierarchical process to prioritize functional requirements for parallel software development. *Requir. Eng.* **2023**, *28*, 347–376. [[CrossRef](#)]
43. Mihany, F.A.; Moussa, H.; Kamel, A.; Ezzat, E.; Ilyas, M. An Automated System for Measuring Similarity between Software Requirements. In Proceedings of the 2nd Africa and Middle East Conference on Software Engineering, AMECSE '16, Cairo, Egypt, 28–29 May 2016; pp. 46–51. [[CrossRef](#)]
44. Mumtaz, M.; Ahmad, N.; Usman Ashraf, M.; Alshafut, A.; Alourani, A.; Anjum, H.J. Modeling Iteration's Perspectives in Software Engineering. *IEEE Access* **2022**, *10*, 19333–19347. [[CrossRef](#)]
45. Siakas, E.; Rahanu, H.; Georgiadou, E.; Siakas, K. Towards Reducing Communication Gaps in Multicultural and Global Requirements Elicitation. In Proceedings of the Systems, Software and Services Process Improvement (EuroSPI 2021), Krems, Austria, 1–3 September 2021; Springer: Cham, Switzerland, 2021; CCIS; Volume 1442, pp. 222–232. [[CrossRef](#)]
46. Khurana, D.; Koli, A.; Khatter, K.; Singh, S. Natural language processing: State of the art, current trends and challenges. *Multimed. Tools Appl.* **2023**, *82*, 3713–3744. [[CrossRef](#)] [[PubMed](#)]
47. Yip, M.H.; Juhola, T. Stakeholder involvement in software system development—Insights into the influence of product-service ratio. *Technol. Soc.* **2015**, *43*, 105–114. [[CrossRef](#)]
48. Ajjiga, D.; Okeleke, P.A.; Folorunsho, S.O.; Ezeigweneme, C. Methodologies for developing scalable software frameworks that support growing business needs. *Int. J. Manag. Entrep. Res.* **2024**, *6*, 2661–2683. [[CrossRef](#)]
49. Karre, S.A.; Mittal, R.; Reddy, R. Requirements Elicitation for Virtual Reality Products—A Mapping Study. In Proceedings of the 16th Innovations in Software Engineering Conference, Allahabad, India, 23–25 February 2023; pp. 1–11. [[CrossRef](#)]
50. Pinto, R.; Silva, L.; Valentim, R. Managing sessions of creative requirements elicitation and assessment. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 30 March–3 April 2020; pp. 1355–1362. [[CrossRef](#)]
51. Paldês, R.A.; Canedo, E.D.; de Albuquerque Guimarães, F.; Calazans, A.T.S. Functional Requirements Elicitation in IoT Systems: A follow-up study. In Proceedings of the 19th Brazilian Symposium on Software Quality (SBQS), São Luís, Brazil, ACM, 1–4 December 2020; pp. 1–10. [[CrossRef](#)]
52. Mesquita, R.; Silva, G.; Canedo, E. On the Experiences of Practitioners with Requirements Elicitation Techniques. In Proceedings of the XXXVII Brazilian Symposium on Software Engineering, Campo Grande, Brazil, 25–29 September 2023; pp. 442–451. [[CrossRef](#)]

53. Shailesh, T.; Nayak, A.; Prasad, D. An UML Based Performance Evaluation of Real-Time Systems Using Timed Petri Net. *Computers* **2020**, *9*, 94. [[CrossRef](#)]
54. Herrmann, J.P.; Imort, S.; Trojanowski, C.; Deuter, A. Requirements Elicitation for an Assistance System for Complexity Management in Product Development of SMEs during COVID-19: A Case Study. *Computers* **2021**, *10*, 149. [[CrossRef](#)]
55. Kitsios, F.; Kyriakopoulou, M.; Kamariotou, M. Exploring Business Strategy Modelling with ArchiMate: A Case Study Approach. *Information* **2022**, *13*, 31. [[CrossRef](#)]
56. Dintén, R.; Zorrilla, M. Design, Building and Deployment of Smart Applications for Anomaly Detection and Failure Prediction in Industrial Use Cases. *Information* **2024**, *15*, 557. [[CrossRef](#)]
57. Muñoz, E.; Capon-Garcia, E.; Muñoz, E.M.; Puigjaner, L. A Systematic Model for Process Development Activities to Support Process Intelligence. *Processes* **2021**, *9*, 600. [[CrossRef](#)]
58. He, C.; Zhang, C.; Bian, T.; Jiao, K.; Su, W.; Wu, K.J.; Su, A. A Review on Artificial Intelligence Enabled Design, Synthesis, and Process Optimization of Chemical Products for Industry 4.0. *Processes* **2023**, *11*, 330. [[CrossRef](#)]
59. Amin, T.U.; Shahzad, B. Improving requirements elicitation in large-scale software projects with reduced customer engagement: A proposed cost-effective model. *Requir. Eng.* **2024**, *29*, 403–418. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.