Journal of Computing and Security

Research Article, 2025, Volume 12, Number 2 (pp. 25–47)

http://www.jcomsec.org

Aspect-Oriented Taxonomies of Requirements Development: A Systematic Review

Hassan Rashidi [a,*] Soheil Afraz [b] Zeynab Rashidi [c]

[a] *Faculty of Statistics, Mathematics and Computer Sciences, Allameh Tabataba'i University, Tehran, Iran.*
[b] *Faculty of Computer Engineering, Islamic Azad University, Ardabil Branch, Ardabil, Iran.*
[c] *Department of Instructional Technology, Allameh Tabataba'i University, Tehran, Iran.*

ARTICLE INFO.

Article history:
Received: 18 April 2025
Revised: 05 May 2025
Accepted: 01 July 2025
Published Online: 23 September 2025

Keywords:
Analysis, Elicitation, Business Requirements, Modelling, Negotiation, Requirements Development.

ABSTRACT

This study focuses on requirements development, a vital phase in the success of software projects, with particular attention to the classification of both functional and non-functional requirements. Despite its crucial role, there has been relatively limited research focused on developing comprehensive taxonomies for requirements development processes. To bridge this gap, this paper conducts a systematic literature review, examining studies published between 2011 and 2024 across six electronic databases. From an initial pool of 1025 studies, 250 were selected for further review based on their relevance to eight research questions. After careful manual scrutiny and application of specific quality metrics, a final set of 90 studies was identified for detailed analysis. The review revealed that most existing taxonomies primarily focus on activities such as requirements elicitation, analysis, and modelling, often overlooking other important phases. Building upon previous work, this paper proposes four new taxonomies that encompass various stages of requirements development, including elicitation, analysis, modelling, specification, verification, and validation. These proposed taxonomies aim to provide a more holistic view of requirements engineering and serve as practical tools for engineers and stakeholders to improve the effectiveness and accuracy of requirements development. By integrating and expanding upon previous taxonomies, these new taxonomies are designed to address neglected aspects and facilitate better decision-making throughout the requirements process, ultimately leading to higher-quality software solutions. The proposed taxonomies support practitioners in identifying better approaches to requirements management and development, contributing to the overall success of software projects through more systematic and structured requirements practices.

1 Introduction

Nowadays, the main reason for software project failures is less attention to the requirements and their incomplete elicitation. The main criterion for the pro-

* Corresponding author.
Email address: hrashi@atu.ac.ir (H. Rashidi)
https://dx.doi.org/10.22108/jcs.2025.144966.1165
ISSN: 2322-4460

duction of high-quality software is the thorough inception of requirements. Due to the importance of requirements engineering and the efforts of developers in the production of high-quality software, extensive academic and industrial research has been carried out. Therefore, the systematic and organized description of research subjects [1] in various sciences and engineering areas significantly contributes to the advancement of knowledge and research in that field [2].

Requirements Engineering (RE) is one of the main challenging stages of obtaining knowledge in software engineering, which is known as requirements in many kinds of literature [3]. Requirements engineering is an umbrella activity that covers the entire software development cycle. The main activities in requirements engineering are focused on the development phases of the requirements, which include elicitation, analysis, modelling, negotiation, specification, verification, and validation V&V (see [4, 5] and [6]).

As an important point, addressing the role of stakeholders and classifying them as a key element in requirements development is essential for the correct and accurate elicitation of requirements. Because they directly influence requirements engineering activities in the development and management phases. The study in [7] showed that the significant distribution of defects (56%) and efforts to fix them (82%) are related to the requirements, indicating the necessity for software requirements management.

Furthermore, there are several standards for the specification of software requirements [8] that help experts to document comprehensive specifications of requirements based on specific features. For example, the IEEE Std.830-1998 provides a structure (template) for documenting the software requirements. They do not commit common mistakes and find the correct inception of requirements. However, due to the different approaches in the analysis of the requirements as well as variations in the modelling and notation of the requirements, extensive research has been conducted to provide standard structures for documenting them.

According to the definition of the Oxford Dictionary, taxonomy is a schematic of the classification [9]. Hence, the proposed structures for classifying the taxonomies include hierarchical, paradigm, tree-based, and aspect-based structures. The aim of taxonomies is the exact description and terminology of the main subject, as well as the definition of qualitative and quantitative classification procedures [10, 11]. Importantly, there is no significant comprehensive study on a multi-aspect general taxonomy of requirements development. Therefore, the existing gaps and defects in prior research in the taxonomies of requirements development have provided the necessary impetus for further research in this area. Consequently, in this review, the hierarchical and applicable taxonomies are proposed for use by the requirements development engineers and provide more details of the domain knowledge in requirements engineering research.

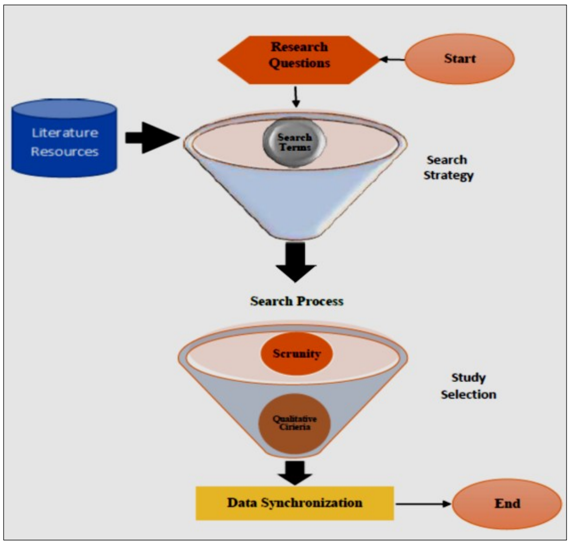

Figure 1. Review Phases.

The remaining sections of this paper are organized as follows. Section 2 presents the research method in this study. Section 3 gives an overview of the recent taxonomies on requirement development. Section 4 proposes a set of new taxonomies in requirements development. Additionally, this section introduces a new taxonomy of the requirements development process in terms of elicitation, modelling, specifications, as well as verification and validation. Finally, Section 5 considers the summary and future works.

2 Research Method

We adopted the approach shown in Figure 1 from [12]. The six phases of the review protocol are: research questions, search strategy, results, scrutiny, qualitative criteria, and data synthesis. The first phase involves formulating several research questions according to the aim of the research. In the second phase, we design search strategies according to the research questions, search keywords, and selection of literature resources. The third phase is about the result of extracted data, while the fourth phase focuses on the data refinement based on securing the title of the collected research. In the fifth phase, we evaluate securitizing by qualitative criteria. Finally, the selected studies are analysed for the subsequent actions.

2.1 Research Questions

This study aims to summarize the state-of-the-art research about software requirements development. We identify the research gaps and recommend points to promote the generality and performance of previous taxonomies in selected aspects. The following eight research questions (RQs) are designed to achieve the aim of the study. The following questions are intertwined and investigated simultaneously.

- RQ1: What are the taxonomies of requirements development?
- RQ2: Which drawbacks and gaps are in the existing requirements development taxonomies?
- RQ3: How many techniques exist in requirements elicitation, verification, and validation?
- RQ4: What are the current approaches or methodologies in requirements development?
- RQ5: Which tools are applied in requirements development?
- RQ6: Which models are proposed in the requirements development phases?
- RQ7: Which notations and types are used in requirements modelling?
- RQ8: How many standards and templates are used in requirements specification?

2.2 Search Strategy

A detailed explanation of the search strategies used in this study includes the search string, literature sources, and search process, which are described below:

2.2.1 Search Strings

To conduct this research, we used the following search string: Requirements AND (development/OR elicitation/ OR analysis /OR modelling / OR specification / OR verification / OR validation) AND ("categories" OR "taxonomies" OR "classifications" OR techniques OR "methods" OR "approaches /" OR "methodology/" OR "tools/" OR "types/" OR "notations" OR "templates" OR "standards/" OR "models/" OR "issues/" OR "challenges/"). The following steps were used to construct the search terms [11]:

- Extracting key terms from the research questions.
- Identifying alternative spellings and synonyms recognition for key terms.
- Identifying keywords in relevant papers or books.
- Using Boolean OR to include alternative spellings and synonyms.
- Using Boolean AND to link the major terms.

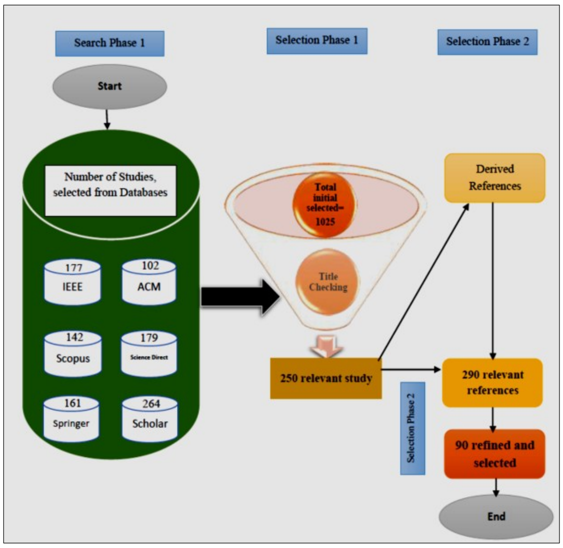

Figure 2. Search and Selection Procedure.

2.2.2 Literature Resources

In this study, six electronic database sources were used to extract data for synchronization. These include: IEEE Xplore, Springer, ACM Digital Library, Science Direct, Scopus, and Google Scholar. The title, abstract, keywords, and conclusion/summary were used to search for published journal papers, IEEE bulletins, conference proceedings, workshops, symposiums, and book chapters.

2.2.3 Search Procedure

A systematic literature review refers to a comprehensive search for all related sources on the requirements development. In this study, the method used is Systematic Literature Review (SLR) according to the guidelines proposed by Kitchenham [12]. However, the search procedure used in this study includes the following steps, as shown in Figure 2 .

- Search Phase 1: A complete search started on six electronic database sources, and the returned results (papers) were collected as papers for the future. (The overlapping and duplicative studies have been discarded at this phase)
- Search Phase 2: Reference lists of all related papers were investigated to identify further related papers, and then, if any, they were combined with items in Phase 1.

2.3 Studies Selection

In the first stage of the search, prospective studies were conducted to eliminate duplicate, similar, and irrelevant research documents. Then, the titles and keywords of these papers were used to examine and scrutinize the relevant studies. As a result, 250 relevant studies were selected. Given the possibility of missing some articles in the initial search phase, the references of each selected article were studied to identify important relevant literature. This effort led to the identification of 41 other studies, bringing the number of selected studies to 290. Finally, a quality assessment criterion was applied to these 290 studies. Therefore, at the end of this process, 90 studies were selected and were able to answer and satisfy the formulated research questions.

Table 1 and Table 2 show the number of selected studies in six databases based on sources, respectively. In these tables, the notation 'n (m)' indicates 'n' is the number of selected studies, and 'm' is the total number of studies on RD phases. Moreover, 'Other Sources' indicates Other Sources that include IEEE standards, bulletins, and magazines.

Table 3 shows the number of studies in the databases of 'Science Direct', 'IEEE', 'Springer', as well as 'ACM, Scopus, and Google Scholar'. In this table, the subject area focused on 'computer sciences', 'decision sciences and engineering' (in Science Direct). Publication Titles are limited to 'Information and Software Technology', 'Journal of Systems and Software', 'Procedia Computer Science', 'Computer Standards & Interfaces', 'Computers in Industry', 'Information Systems', 'Science of Computer Programs', and 'Information & Management Journals' (in Science Direct). The notation 'n (m)' indicates 'n' is the number of selected studies, and 'm' is the total number of studies on RD phases.

2.3.1 Scrutiny

According to Figure 2, the prospective studies were obtained during the first search phase. Therefore, the detailed examination of these studies was necessary to make them effective. First, the titles of the papers were reviewed with a glance at the keywords. Then their contents, including results and conclusion or summary, were skimmed. Hence, all articles that do not reflect the topic of discussion or do not cover any of the research questions were excluded from the relevant studies list. In addition, only English language studies from reputable journals, arbitration review papers, workshops, symposia, book chapters, and IEEE bulletins are included in the list of related studies.

When several similar or improved versions of an article are found in different databases in the search results, the most complete and updated literature published in authentic journals or scientific events is included in the search phrases, and the rest are eliminated. Specifically, we conducted a systematic literature review on the development of requirements for articles published by 2024 between 1 January 2011 and July 31, 2024. These articles included taxonomies, classification, reviews, and surveys in this research area. A summary of the used criteria for review is shown in Table 4.

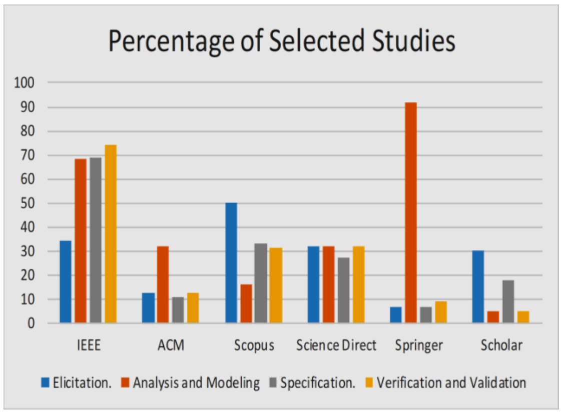

Figure 3. shows the percentages of studies in: (a) Science Direct, (b) IEEE, (c) Springer, (d) ACM, Scopus, and Google Scholar databases.

- a: The research aspects include Techniques, Approaches, Methodologies, Tools, Models, Types, Notation/Templates, Issues/Problems/Challenges, Standards/levels, and specific aspects.
- b: The nominal scales of fitting degree: Low, Medium, and High.

2.4 Data Synchronization

Due to the need to summarize the evidence of selected studies to investigate the coverage of research questions, data synchronization should be carried out. Here, among selected studies, 90 of the studies were investigated to assess the exact contents of each study according to the criteria defined in Table 6.

The purpose of this activity is to synchronize the selected studies to increase clarity. This also helps to identify the exact answers to the research questions and their coverage degree. The elicited data obtained from data synthesis in two levels of quantitative data (such as the number of cases obtained from aspects) and qualitative data (such as the level of coverage of research questions in the requirements development phases, the relationship between selected studies and requirements development, and also existing challenges and gaps in the literature) is classified. Whereby, in the tables presented in the third part of this study, the mapping status of the coverage domain and the degree of compliance of research questions with the proposed taxonomies are specified (see Tables 1 to 6).

Table 1. The Number of Studies Selected From Six Databases.

Database	Elicitation	Analysis and Modelling	Specification	Verification and Validation
1-IEEE	12 (35)	56 (82)	20 (29)	23 (31)
2-ACM	8 (63)	7 (22)	1 (9)	1 (8)
3-Scopus	5 (10)	18 (113)	1 (3)	5 (16)
4-Science Direct	17 (53)	26 (82)	6 (22)	7 (22)
5-Springer	4 (58)	11 (120)	4 (59)	3 (32)
6-Scholar	30 (100)	4 (76)	9 (50)	2 (38)
Total	76 (319)	122 (495)	42 (172)	41 (147)

Table 2. The Total Number of Studies Selected From Research Sources.

Sources	Elicitation	Analysis and Modelling	Specification	Verification and Validation
Journal	61 (267)	73 (415)	27 (126)	24 (117)
Chapters & Conferences	9 (46)	47 (78)	9 (38)	17 (31)
Other sources	6 (6)	2 (2)	6 (8)	-
Total	76 (319)	122 (495)	42 (172)	41 (147)

2.5 Threats to Validity

Because publication bias in the choice of studies is a serious threat in review studies. To perform the review, studies were selected without bias based on the search strategies discussed in the previous section, including different literature databases, selection criteria, and qualitative criteria.

On the other hand, it is practically impossible to fully extract studies based on RQs defined in 6 selected databases (considering the overlap of articles in some databases and indexing). Hence, there is a probability of missing some of the desired studies. To curb this threat, manual scrutiny of references of the extracted studies was carried out to identify the desired studies that were lost in the initial search. Meanwhile, due to the intertwining of research on aspects in the requirements development stages, more precision has been taken in defining exit criteria.

In general, to reduce the effects of the mentioned threats on the quality of this study, the definition of processes and strategies, as well as the design of criteria, is based on the consensus of the authors of the article, and the approaches of similar literature have been addressed.

3 An Overview of Existing Taxonomies

Many studies suggest useful taxonomies and classifications in requirements engineering activities [13, 14]. In this section, we conduct an overview of the important taxonomies on methodologies, techniques, and tools in requirements engineering. After that, the requirements development issues are thoroughly reviewed.

As an integrated view of requirements classification, the layered view of requirements and their types is shown in [5]. It indicates that Business Requirements (BR) at the business level of software requirements are one of the four fundamental perspectives for the requirements engineering solution [15]. Accordingly, Afreen et al. [16] provided a new classification of non-functional requirements (NFR), which are useful for requirements analysts. For example, the performance requirements play a significant role in software development, which can be considered in the category of NFRs [17]. Also, the techniques, approaches, and tools, as well as issues, trends, pitfalls, and current challenges of requirements elicitation were investigated by Zowghi and others. They emphasized that the proper identification and selection of technique, approach, or tool in requirements elicitation could play a significant role in reducing the complexity and costs of other requirements development stages [18–20].

Basri et al. did a systematic literature review on the current trend of software requirement engineering processes in Small and Medium Enterprises (SMEs)

Table 3. The number of studies in the Database of Science Direct, IEEE, Springer, as well as ACM, Scopus, and Google Scholar.

	Database							
	Science Direct				**IEEE**			
Type of Research	Elicitation	Analysis and Modelling	Specification	Verification and Validation	Elicitation	Analysis and Modelling	Specification	Verification and Validation
Review Article	2 (3)	6 (26)	1 (2)	1 (1)	1 (12)	5 (5)	8 (8)	7 (11)
Publication Title	6 (21)	9 (24)	1 (8)	3 (7)	7 (19)	34 (75)	8 (17)	16 (20)
Subject Area	9 (29)	11 (32)	4 (12)	3 (14)	4 (4)	2 (2)	4 (5)	-
Total	17 (53)	26 (82)	6 (22)	7 (22)	12 (35)	56 (82)	20 (29)	23 (31)

	Database							
	Springer				**ACM, Scopus, and Google Scholar**			
Type of Research	Elicitation	Analysis and Modelling	Specification	Verification and Validation	Elicitation	Analysis and Modelling	Specification	Verification and Validation
Review Article	1 (18)	7 (79)	2 (22)	1 (11)	8 (63)	7 (27)	1 (9)	1 (8)
Publication Title	2 (27)	2 (13)	1 (21)	1 (11)	30 (100)	4 (76)	9 (50)	2 (38)
Subject Area	1 (13)	2 (28)	1 (16)	1 (10)	5 (10)	18 (113)	1 (3)	5 (16)
Total	4 (58)	11 (120)	4 (59)	3 (32)	43 (173)	29 (216)	11 (62)	8 (62)

[21]. This research emphasized the importance of considering the human elements of SRE, such as team dynamics, training, and motivation, and suggests that SMEs should establish specialized SRE processes that fit their requirements and resources.

There are some valuable works entitled Systematic Literature Review (SLR), Systematic Mapping Study (SMS), or Survey in the requirement engineering process involving development and management phases. An SLR about requirements elicitation was presented to identify the latest research in this field, based on a general framework for literature review. The research questions of the mentioned SLR concentrate on three concepts: covered aspects, activities, and influencers in requirements elicitation. Although there are some defects in the automation of requirements elicitation activities, there are also some missing points in this process. Also, Rodriguez and Wong introduced some frameworks, models, methods, techniques, and tools for elicitation activity [22]. Since stakeholders have the most significant role in the elicitation of requirements [23**?**], Burnay proposed a Taxonomy of Elicitation Information Sources called TELIS, which can be used during the elicitation phase to systematically investigate the sources of information about a system by requirements engineers [24]. Thereafter, a Stakeholder quantification and prioritization (SQP) SLR was proposed based on Kitchenham and Charters' guidelines

Table 4. Inclusion and Exclusion Criteria.

Type / Criteria	Inclusion	Exclusion
Published Language (C1)	English	Not English
Publishing Time (C2)	1 Jan. 2011 – 31 July 2024	Out of range
Concentration (C3)	Requirements Development Phases	Requirements management / out of software requirements
RQ coverage (C4)	At least covering one designed RQ	Eliminated papers and unable to respond to RQs
Relative content (C5)	Title / Abstract / Keywords / Conclusion (summary)	Introduction / Literature / Results (artifacts)

[11] that SQP is a crucial process in requirement prioritization [25].

AlJassasi et al. provided a front-end requirement engineering framework of social commerce enterprises with a focus on the design phase [26]. This study has sketched out the generic set of front-end requirements of an s-commerce platform and showed how to use the generic framework to generate or recommend an instantiation of the requirements for a specific platform. This study can assist the developers to focus on the social features that enhance the economic revenue of the enterprise and improve the architecture of their products/services and/or their business processes (BPs).

Yapa et al. explored the critical success aspects of requirement engineering that influence software development efficiency [27]. This study was conducted to gather data from business analysts within a leading telecommunication organization and its subsidiaries across three countries. The data was thoroughly analyzed through a structured analysis using Smarts PLS. The success aspects considered in this study are the business analyst's skills, the quality of the requirement, customer involvement, and team capability. The results highlighted that software development efficiency is strongly influenced by requirements engineering aspects, including the skills of the Business Analyst, customer involvement, and team capability, each with a significance value of less than 0.05. By addressing requirement engineering challenges and implementing standardized procedures derived from critical success aspects, organizations can streamline their requirement engineering processes. This study provided valuable insights to enhance requirement engineering best practices, fostering software development efficiency.

The prioritization of requirements is known as one of the essential activities in the elicitation and analysis phases. They are reviewed based on their techniques and tools [28, 29], metrics, and sizes. Further, in the new SLR, Elicitation, and gathering of requirements in Mobile Application Development (MAD), compared to elicitation techniques in traditional development, are addressed in [30]. Besides, the impact of Metadata and document interpretation in Software Requirements Specification and elicitation of NFRs from requirements artifacts was explained by Asif et al., who classified NFRs into three categories: security, performance, and cost [31]. In addition, Dermeval et al. [32] conducted an SLR on applications of ontologies in RE. Because the ontology-driven approaches support RE activities, such as requirements elicitation, analysis, specification, validation, and management issues.

An effective alternative to requirements documents is to store the information in a database. Dozens of requirements management (RM) tools are available for this purpose [33]. They range from simple, open-source products available for free download to complex commercial packages that can handle huge systems development projects. These tools make it much easier to work with requirements than documents or other representations permit. An effective change control process serves as a structure to manage change effectively, not as a barrier [34].

Developing machine learning (ML) approaches for requirements classification has attracted great interest in the RE community in recent years. The research [35] addressed two related problems that have been challenging real-world applications of ML approaches: the problems of class imbalance and high dimensionality with low sample size data (HDLSS). This research proposed an ML approach for multiclass classification of requirements that solves the aforementioned problems through semantic-role-based feature selection, dataset decomposition, and hierarchical classification. This research compared the effectiveness of the proposed approach with two traditional statistical classification models and an advanced deep learning model. The experiment showed that the proposed approach is simple to use and can effectively address the class imbalance and HDLSS problems compared to similar approaches.

Agarwal et al. focused on Deep Learning in requirement engineering [36]. This research examined the bibliographies of 98 research papers that employ deep learning techniques to respond to these inquiries. Deep learning integrated solutions have facilitated 41 tasks across all phases in software engineering. This study

Table 5. Quality Assessment Questions/Metrics.

QA Metric	Questions
M1	Are the objectives of the research clearly explained?
M2	How much is the contribution or achievement of the article to industry or academia?
M3a	Are the aspects to be clearly described in papers on Requirements Development?
M4	Is the structure or categorization of reviews suitable?
M5b	How much does the data set used in the experiments sufficiently fit the objectives of the articles?
M6	To what extent is the overlap of aspects taken in the articles?

Table 6. Assessment Criteria for Study Contents.

Selected study	Criteria definition
Recognition of the bibliography in the study	Unique ID for the study in terms of publication year, title, subject area, and publisher
Study type	Journal and conference papers, IEEE bulletins, magazines, and book chapters
Study concentration	Subject area, problems, challenges, motivation, and goals
Research methodology	Case study, survey, classification, taxonomy, comparison, experiment, interview, observation, and questionnaire
Data analysis	Quantitative/qualitative analysis
Application domain	Applicability of the study in academia or industry
Constraints	Identification of the study's deficiencies, research gaps, and future research directions
Study finding	Lessons and practical/theoretical solutions learned from the study

indicated that 84.7% of papers only use standard profound learning models and variations to address issues in software engineering. Utilizing significant learning systems brings worries up for their practicality. The viability, proficiency, understandability, and testability of profound learning-based arrangements might provoke the curiosity of additional specialists in software engineering.

Another research [37] addressed the requirement elicitation and evaluation for product-service systems (PSSs). This research proposed a requirement elicitation and evaluation framework for PSSs. It introduced a new requirement elicitation method based on contextual matching to capture the complete PSS requirements. Moreover, it proposed a hybrid decision-making method that considers the preferences of enterprise experts and users, as well as the uncertainty during the decision-making process, to determine the prioritization of PSS requirements. In the experiment, an illustrative case study of an automatic guided vehicle PSS was examined to justify the proposed framework and methods.

Saleem et al. did a study on generative language models for requirement engineering applications [38]. Traditional language models have been extensively evaluated for the software engineering domain; however, the potential of ChatGPT and Gemini has not been fully explored. To fulfil this gap, the paper in hand presents a comprehensive case study to investigate the potential of both language models for the development of diverse types of requirement engineering applications. It deeply explored the impact of varying levels of expert knowledge prompts on the prediction accuracy of both language models. Across 4 different public benchmark datasets of requirement engineering tasks, it compared the performance of both language models with existing task-specific machine/deep learning predictors and traditional language models. Specifically, the paper utilized 4 benchmark datasets: Pure (7445 samples, requirements extraction), PROMISE (622 samples, requirements classification), REQuestA (300 question-answer (QA) pairs), and Aerospace datasets (6347 words, requirements NER tagging). Our experiments reveal that, in comparison to ChatGPT, Gemini requires more careful prompt engineering to provide accurate predictions. Moreover, across the requirement extraction benchmark dataset, the state-of-the-art F1-score is 0.86, while ChatGPT and Gemini achieved 0.76 and 0.77, respectively. The State-of-the-art F1-score on the requirements classification dataset is 0.96, and both language models are 0.78. In the name entity recognition (NER) task, the state-of-the-art F1-score is 0.92, and ChatGPT managed to produce 0.36, and Gemini 0.25. Similarly, across the question answering dataset, the state-of-the-art F1-score is 0.90, and ChatGPT and Gemini managed to produce 0.91 and 0.88, respectively. Our experiments show that Gemini requires more precise prompt engineering than ChatGPT. Except for question-answering, both models underperform compared to current state-of-the-art predictors across other tasks.

4 The Proposed Taxonomies of Requirements Development

The requirements engineering activities influence the software development and life cycle, and consequently the software architecture design. These activities involve gathering, documenting, and managing requirements. Many RE process models depend on the domain of application, the system stakeholders, and the Organizations that utilize the requirements. Nevertheless, there are several generic RE activities common to all processes: elicitation, analysis and modelling, specification, verification and validation, and management and tracing of requirements.

This section pursues two important goals. First, it investigates the previous taxonomies and classifications based on various practical aspects until the involved deficiencies and gaps are discovered and revealed. Second, by considering the neglected aspects as well as the integration and aggregation of different aspects in prior works, new taxonomies in the requirements development are presented. Hence, the requirement engineering based on terminology, their descriptive principles, and qualitative and quantitative classification procedures with a graphical representation approach, and new taxonomies for the requirements development process are presented in this section.

4.1 Requirements Elicitation

Requirements elicitation is a process through which the system providers discover, review, integrate, understand, and document the requirements of that system and its life cycle process. On the other hand, the abundance of recent research and validated papers in this area of requirements engineering indicates the importance of this phase in the requirements engineering process (see [18] and [39]).

Based on the recent development in requirements elicitation and prioritization, we address their taxonomies here. In Figure 4 and Figure 5, a new aspect-based and integrated taxonomy is presented for requirements elicitation. This taxonomy provides five different key aspects: (a) 'Techniques', 'Approaches/Methodologies' in Figure 4, and (b) 'Models', 'Issues and Pitfalls' as well as 'Tools' in Figure 5. The different parts of these figures are shown with different colours and the important parts described briefly below. Along with these aspects, one can also consider issues, pitfalls, and related trends and challenges during the elicitation stage.

In the left part of Figure 4 (see the box 'Techniques'), the aspect techniques include 'Traditional', 'Contextual', 'Collaborative', and 'Cognitive' that form a significant part of our taxonomy. The traditional techniques comprise the 'Interview', 'Questionnaires' (Surveys), 'Document Analysis', 'Introspection', 'Meeting', and 'Data Gathering from existing system'.

The 'Interview' is the most popular and traditional technique used in RE elicitation [40]. With the help of this technique, a massive volume of data can be gathered quickly and efficiently. Thus, the feedback on using this technique enormously depends on the expertise of the interviewer.

A 'Questionnaire' is a requirements elicitation technique that extracts the requirements from a large number of people at a lesser cost and time. A well-designed questionnaire can be useful to elicit the actual requirements from the stakeholders. These surveys are used to collect data from a sample of persons [40].

In the 'Document analysis' technique, information from existing documents is gathered and analyzed [41]. The technique is successfully used to initiate the process of requirement elicitation. The information collected in this technique may vary due to the availability of documents and interactions with humans, so it is often used when there is a need to have domain information studied by a domain expert.

In the 'Introspection' technique, the requirement analyst attempts to develop all the requirements based on the needs and wishes of stakeholders for a particular system [42]. The technique is more effective when the analyst is fully aware of the field. Since this technique is used with a combination of other techniques to initiate the requirement elicitation process [39], the

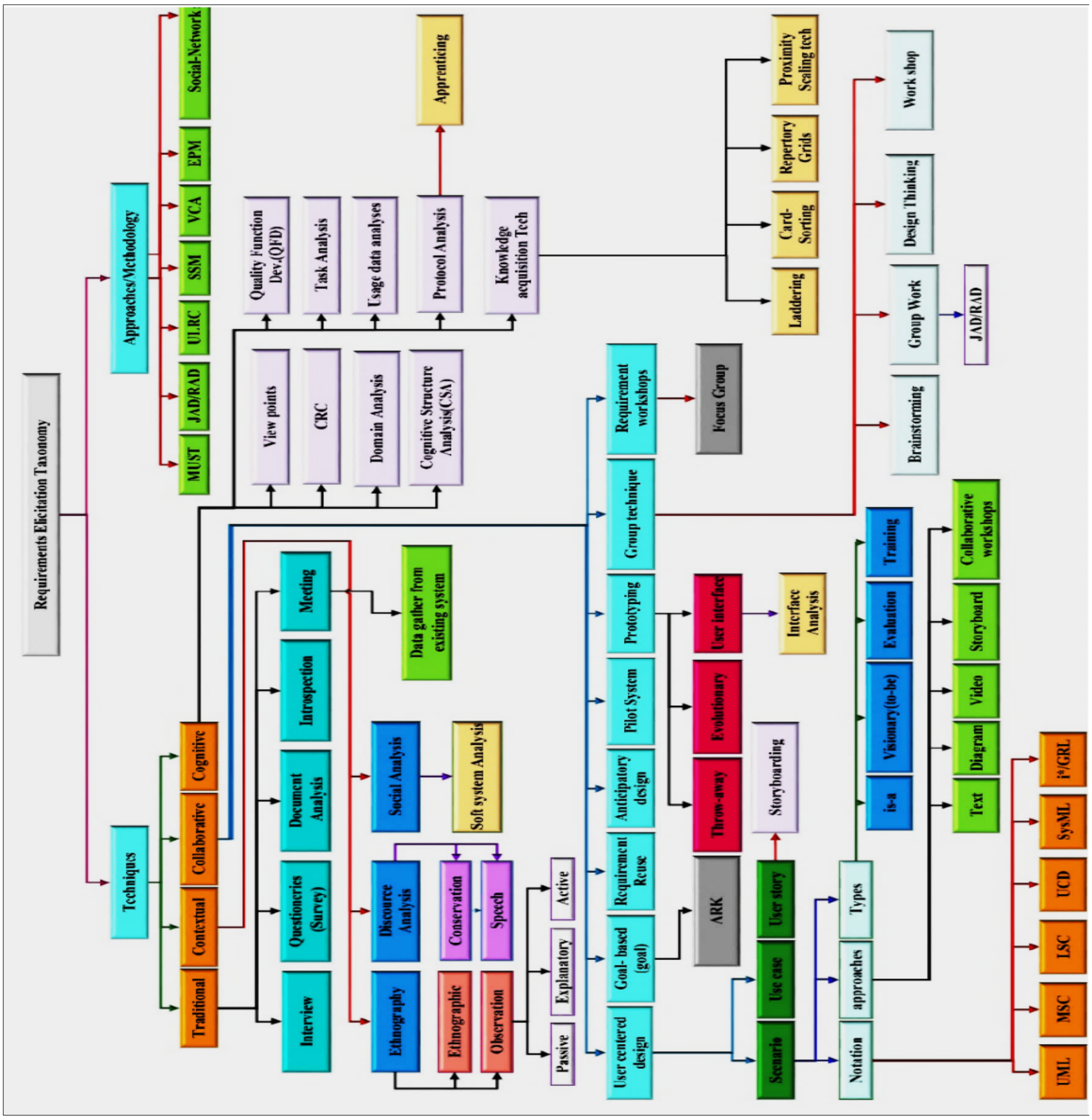

Figure 4. A New Requirements Elicitation Taxonomy.

system analyst must have experience and expertise in the domain.

Another traditional technique called 'Data gathered from the existing system' is used when we collect data for a system to replace the current system. Hence, this is a useful way to gather profound system knowledge [40]. In brief, the 'Contextual' (Observational) techniques embrace 'Social Analysis', 'Discourse Analysis', and 'Ethnography', and their related method [39]. Also, the 'Collaborative' techniques are classified into several categories: pilot system, requirement workshop, requirement reuse, anticipatory design, goal-based (goals), prototyping, group techniques, and user-entered design (e.g., storyboarding [40]). The instances of collaborative techniques are the use-case, user story, and scenario, which have their types, approaches, and notation. The important notations of scenarios that can be mentioned are UML (in RUP methodology), 'MSC', 'LSC', 'UCM' , 'SysML ', and 'i*/GRL' (goal-based). Further, the group techniques consist of brainstorming (as a creativity technique), group work (e.g.,' JAD/RAD' and workshop), and the design thinking method [40]. Moreover, the 'User interface', 'Throw-away', and 'Evolutionary' are examples of group techniques. In particular, the 'ARK' (Act-based Representation of Knowledge), focus groups in clouds ([43], [44]), and Interface Analysis [45] are instances of Goals, requirement workshops, and user interface techniques, respectively.

The 'Cognitive' techniques branch in prescribed

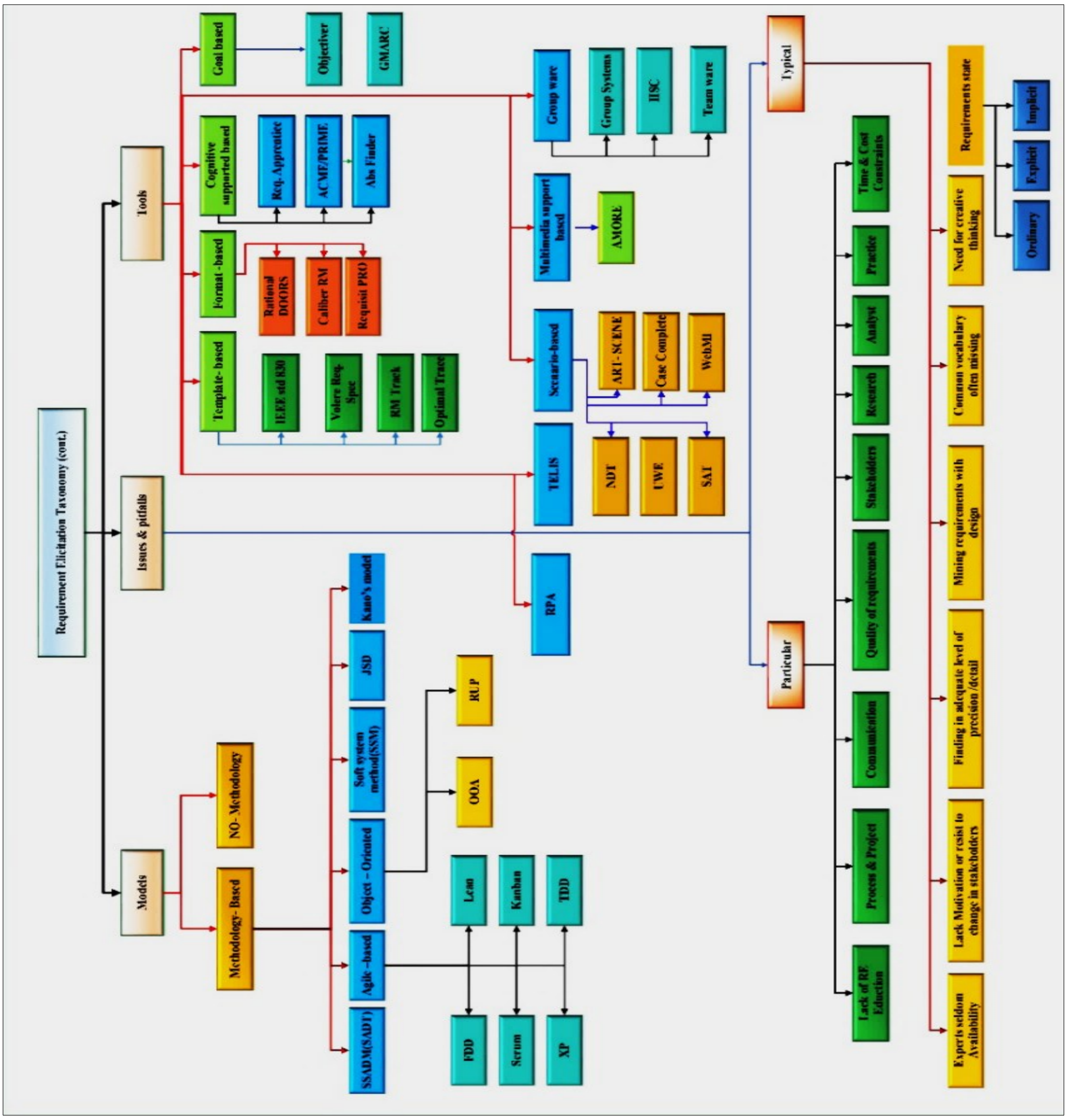

Figure 5. A New Requirements Elicitation Taxonomy (continued from Fig. 4).

taxonomy has various methods, as follows: 'Protocol Analysis' (e.g., Apprenticing) [40], 'Task Analysis' [43], 'Domain Analysis', 'View Points', 'Usage data analysis', 'Quality Function Development (QFD)', 'Cognitive Structure Analysis (CSA)', and 'Knowledge Acquisition' techniques. The last technique includes 'Laddering', 'Card-sorting', 'Repertory Grids', 'Classic Responsibility Collaboration (CRC)', and 'Proximity Scaling' techniques. Moreover, the 'Analogical' techniques include metaphors and personas that help stakeholders to consider more deeply and be more accurate about their requirements [1].

There are different methods to achieve the required information in requirement elicitation, including direct and indirect approaches [46]. However, in the direct approach, we interact with an expert in the domain, and the indirect approach is classified according to the type of information obtained. In particular, interviews, case studies, and prototyping are of the direct type, but surveys and document analysis are indirect.

The second category of the requirement elicitation approaches (see the right part of Figure 4, the box 'Approaches/Methodology') include 'MUST', 'Joint Application Development (JAD/RAD)', 'User-Led Requirements Construction (ULRC)', 'Soft System Methodology (SSM)', 'Value Chain Analysis (VCA)' and 'Elementary Pragmatic Model (EPM)' and 'Social Network' approaches [18]. 'MUST' utilizes ethnographic techniques as a specific step in the methodology, while 'ULRC' and 'SSM' employ ethnographic techniques. Also, 'IBM's JAD' is set up to improve the relationship between user representatives and speed up decision-making [47] with all attendees in five defined phases. Also, VCA can identify, classify, and sort business activities of the enterprise.

The VCA uses the systematic approach from the strategy angle and analyses the composition of the value chain, the status of the value activity, and the relationship among them ([44], [48]). Moreover, the EPM concentrates on a pragmatic" approach by examining the communication of behaviours between two individuals and an "elementary" approach for categorizing communicative interaction [49]. Furthermore, the approaches of automated requirements data-driven elicitation used human source data and techniques for the extraction of process-mediated and machine-generated data. The elicitation process using human source data is performed in three steps. First, the data type analysis is performed by NLP, then the classification is done based on Machine learning, rule-based, or model-oriented schemes, and finally using topic modelling or traditional clustering [50].

We also allocate a category to the issues and pitfalls, that researchers involved in the requirements elicitation phase (see the box 'Issues & pitfalls' in Figure 5). They arise for several reasons, for example, the people involved in the elicitation (e.g., Participation of stakeholders, educational issues (see [51] and [52]), different constraints such as time and cost constraints, requirements sources, status and quality, and some of the problems identified in RE research areas [53]. Therefore, there are many general and particular issues and research pitfalls. Thus, the majority of trends and challenges in requirements elicitation focus on elicitation practices and research [19].

According to the proposed elicitation taxonomy, the elicitation tools branch is classified into nine tools (see the right part of Figure 5-the box 'Tools'). The first type is 'Template-Based' tools that include 'Optimal Trace', 'Volere Shells', 'RMTrack' [54], and 'IEEE-std830' requirements specification (detailed in SRS taxonomy). Also, the 'Format-Based' tools covering 'Rational DOORS', 'CaliberRM', and 'Requisite-pro'. So, the Requirements Apprentice [55], 'ACME/PRIME' [56], and 'AbsFinder' [57] tools are in the cognitive support-based category, and the Objective [58] is considered in 'Goal-Based' tools.

In the rest of the tool branch, the 'ART-SCENE' [59], 'CaseComplete' [54], 'NDT' [51], 'UWE' [60], 'WebML' [61], and 'SAT' [62] tools can be seen in the 'Scenario-Based' category, and the 'AMORE' tool in the multimedia support-based category. Besides, the 'Groupware' tools such as 'Teamware' [58], 'Group Systems' [63], and 'HSC' are all categorized in this taxonomy. Likewise, there are the 'RPA' [64], 'GMARC' [65], and 'TELIS' tools [24], which are the latest and have recently been given more attention by requirement engineers. Moreover, there are other RE tools evaluated in a phase-wise manner by Shah et al. [62].

As a final aspect, the requirement elicitation 'Models' is divided into two approaches: 'Methodology-Based' and 'No-Methodology' or Ad-hoc models (see the left part of Figure 5, the box 'Models'). The 'Methodology-Based' models take into account eight methodologies: 'SSADM', 'Object-Oriented' (e.g., 'OOA' and 'RUP'), Agile-based such as 'XP', 'Scrum', 'Lean', and 'FDD' ([66], [67], the Kanban methodology [68], 'TDD' [69], and 'KAONO's model [70]. In this regard, Kasab et al. The study [71] showed that the use of agile methodologies in the software development life cycle is the most interesting, making it possible to make agile methodologies more popular than waterfall methods. Moreover, Alsharari et al. [72] reviewed the available agile methodologies used to assess requirement change management in web engineering [72]. Srivastava et al. incorporated requirement engineering into agile methodologies by discussing challenges and then proposed solutions [73]. This paper addressed all the issues raised and suggested possible solutions from the standpoint of requirement engineering. The results can be extremely beneficial to the software business in terms of improving development processes, as well as individuals who would like to continue working in this area.

In the end, Table 7 and Table 8 (refer to Figure 4 and Figure 5) indicate the Comparison of the proposed Requirements Elicitation Taxonomy with other taxonomies previously presented.

In summary, in Figure 6, the taxonomy of requirements modelling with an aspect-based structure is presented in two forms: the 'Types' and 'Notations'. We use different colours to clarify the aspects.

There are several challenges in requirements modelling methods, such as diversity, dynamicity, and inconsistency, that make their classification difficult [75]. Nevertheless, the different types of requirements modelling are classified into two formal and informal general categories in the proposed taxonomy. Hence, the 'Natural Languages' or 'Natural Language Processing (NLP)' fall into the informal category [76], and so the 'Goal-Oriented', 'Functional', and 'Diagram Modelling' are suggested formal modelling types. Thus, the 'Goal-Oriented Modelling Languages (GRL)' include 'CSRML' (an i* extension for specifying CSCW Systems requirements), 'KAOS' [77], Troops, Techne [78], and 'Fuzzy (AND/OR)' goal models [79]. Moreover, 'Functional Modelling' comprises 'BPM' [80], 'SDL' [81], and 'UCM' [74].

Table 7. Comparison of the Proposed Requirements Elicitation Taxonomy with Others (Part A).

Author(s)/ [References] (RQ1)	**Aspects**											
	Techniques (RQ3)					**Approaches/Methodologies (RQ4)**						
	Analogical	Traditional	Contextual	Collaborative	Cognitive	MUST	JAD/RAD	ULRC	SSM	VCA	EPM	Social-network
Zowghi & Coulin [19]		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓
Yousuf & Asger [39]		✓	✓	✓	✓		✓		✓			
Kassab et al. [74]		✓		✓	✓							
Cheng et al. [1], Lim et al. [50]	✓			✓		✓						
Proposed Taxonomy		✓	✓	✓	✓							✓

Table 8. Comparison of the Proposed Requirements Elicitation Taxonomy with Others (Part B).

Author(s)/ [References] (RQ1)	**Aspects**										
	Models (RQ6)		**Issues & Pitfalls (RQ2)**		**Tools (RQ5)**						
	Methodology based	No-Methodology	Particular	Typical	TELIS	Template based	Format based	Cognitive supported	Goal based	Multimedia supported	Group ware
Zowghi & Coulin [19]	✓		✓			✓	✓		✓	✓	
Shah et al. [62], Burnay [24]					✓		✓	✓			
Kassab et al. [74]	✓			✓			✓				
Proposed Taxonomy		✓	✓	✓	✓		✓				

Besides this model, the 'Diagram Modelling' consists of 'DSLs/DSML' and 'UML profiles' that are divided into the 'SysML' [80], 'URN' ([82], [83]), and 'MARTE' Stereotypes. Subsequently, the 'SysML' profile can give 'OMEGA2' (an executable UML/SysML profile) and 'SysML/KAOS' Metamodel in combination with UML and KAOS [77] goal-oriented requirements methodology, respectively.

As indicated in another branch of the proposed taxonomy, most research in the notation of requirements modelling languages is focused on the 'Formal', 'Semi-formal', 'Ad-hoc', or Informal notations through 'Natural Languages'. Hence, concerning Formal notation of requirements modelling languages, the 'Logic/Temporal Logic' as a Logic system notation (e.g., Tree logic, CTL , and CTL*) [84], 'SDL' as a complex language, 'Petri-Nets', 'FSM /STD' , and 'BPMN' have been introduced as important instances. Ad-hoc Modelling notations related to important NFRs such as security requirements for secure software design, which include 'Alam-SECTET', 'Buyens-LP', 'Hafner-SOA', 'Hoisl-SOA', 'Kim-AC', 'Kong-Threat', 'Memon-SECTET', 'Nakamura-SOA', 'PbSD', 'Vela-DB-XML', 'Xu-Petri', and 'Yu-AC', etc. ([78], [85]).

Moreover, the notations 'URN', 'SysML', and 'UML' for semi-formal notations are considered. To that end, the 'Use-Case Diagram', 'Class Diagram', 'Activity Diagram', 'Sequence Diagram', 'Collaboration Diagram', as well as 'State-Chart Diagram', are intended for UML in the form of a conceptual model [86]. More specifically, the domain-specific and case-based notations fall into the Ad-hoc category that includes particular SRS models. In particular, for modelling and analysing user requirements, a semi-formal language called 'URN' is presented in the form of goals, scenarios, and relations between them. The 'URN' combines two elicitation notations, analysis, specifica-

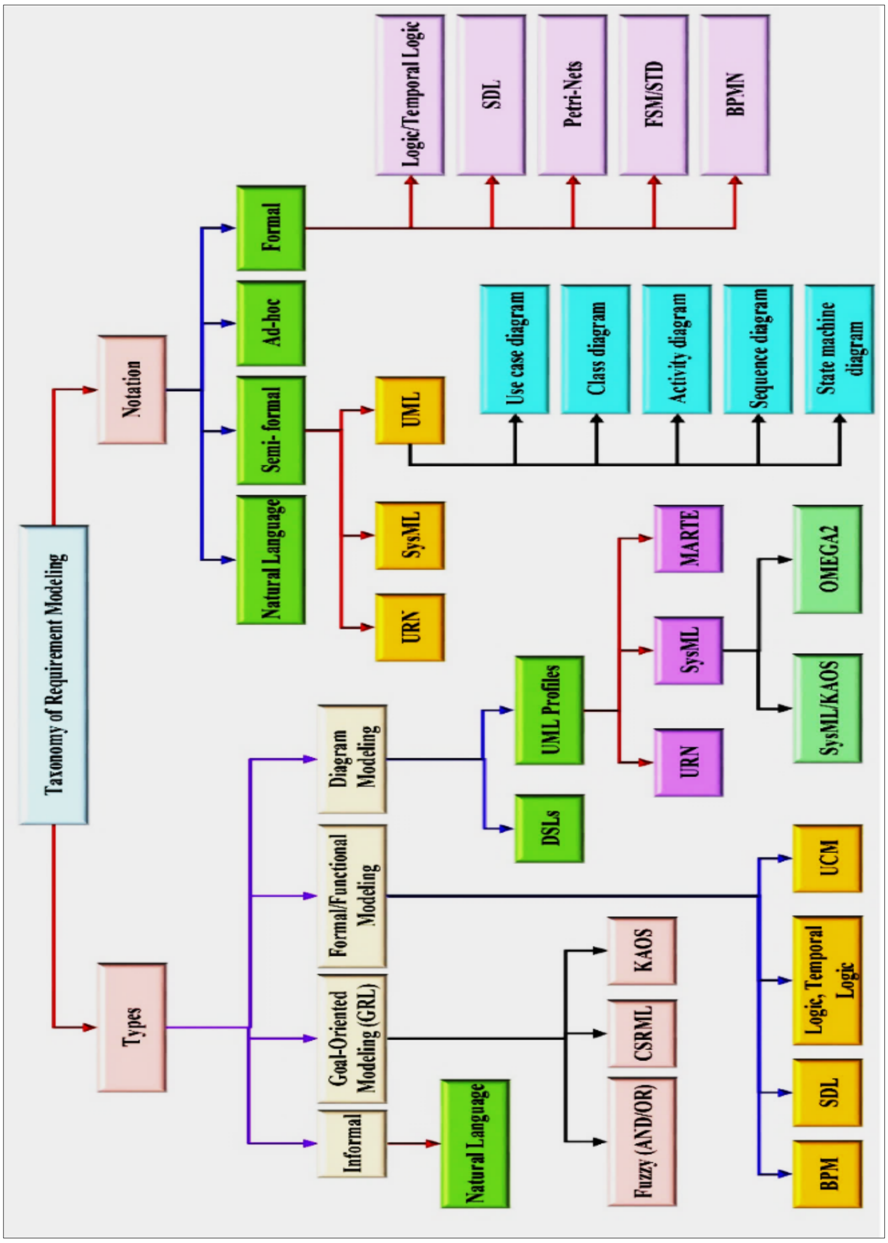

Figure 6. Requirements Modelling Taxonomy.

tions, and validation of requirements. This language allows requirements engineers to find and recognize the requirements of a suggested or evolving system. Furthermore, the 'URN' analyses some requirements for correctness and integrity. As a result, 'URN' models can be used to diagnose and analyse various types of reaction systems, business processes, and telecommunications standards. For example, one of the tools presented based on the 'URN' model is 'jUCMNav', in which there is a relation between 'GRL' and 'UCM' [83].

Overall, it is also important to highlight that the objective of the type branch in this taxonomy is a model-based classification of modelling approaches and languages. However, the purpose of the notation branch is to emphasize its role in the development of requirements models as well as the formation of new modelling languages and profiles with a formalized approach. Hence, the repetition of some concepts in the suggested taxonomy is rooted in our perspective on the categorization of this taxonomy. Table 9 shows the comparison of the proposed Requirements Modelling Taxonomy with other taxonomies in terms of types and notations.

Table 9. Comparison of the Proposed Requirements Modelling Taxonomy with Others.

Taxonomy/ [References] (RQ1)	Types (RQ7)			Notation (RQ7)				
	Informal	GRL	Formal/Functional	Diagram Based	Natural language	Semi-formal	Ad-hoc	Formal
Pnueli [84]	✓		✓		✓	✓		✓
Arif et al. [78], Bozigit et al. [86]	✓		✓		✓	✓		✓
Manzoor et al. [77], Chatzikonstantinou et al. [79]		✓		✓		✓		✓
Ćwikła et al. [80], Belina et al. [81], Janssens [76]			✓	✓	✓			
Amyot et al. [82], Abdi et al. [83]	✓		✓	✓		✓		
Proposed Taxonomy	✓	✓		✓			✓	

4.2 Software Requirements Specification

The main purpose of the requirements specification phase is to obtain documentation of software requirements with standards and templates. They are structured sets of requirements (including functions, efficiency, and design constraints with features) in major software and their external interfaces [87–89]. Ergo, the output of this stage is a requirements specification document. The document must clearly and precisely specify the details of each of the outputs and primary requirements, scope and boundaries, their feasibility, and reality because they are the basis for contractual agreements between contractors or suppliers, and customers.

To provide a convenient SRS , we need to apply some standards or design and use RS 'Templates' (e.g., 'ID', 'Business process', 'Activity', 'Goal', 'Primary Actor', 'Trigger', 'Positive Scenario', and 'Failure Condition and Management' templates). Therefore, these standards and templates are the basis for the requirements Verification and Validation (V&V) phase. In particular, the 'ISO/IEC/IEEE29148-2018' standard is an international standard that accommodates 'IEEE830', 'SWEBOK', and seven other standards ([3], [8], [90]). It provides a detailed template for well-formed textual software requirements and guidance for their implementation. Similarly, the 'CELENEC', 'CEN', and 'ETSI' are European standards that are used in addition to 'DND', 'MIL-STD-498', and some IEEE standards such as 'IEEE1233-1998' and 'IEEE830-1998' in the requirements specification [85]. As a result, the RS standards have an integrated behaviour of processes and products involved in RE processes and operational areas.

On the other hand, the SRS 'Types' should be categorized into 'Executable' and 'Non-Executable' classes with a model-oriented approach. For this reason, the executable models are auto-models, which are classified in the 'XUML' and 'Declarative' categories. The declarative category contains some executable types, for example, the 'Logic Language', 'Functional Language', 'B-Method', 'Franketal', 'jML', 'NP-SPEC', 'SPILL', 'TUG', and 'Z-Notation'. Likewise, the non-executable types of RS fall into two classes: Manual and Automated. The first class includes 'XML', 'Natural Language', 'ARM', and 'TIGER PRO'. Also, the manual class covers a variety of types such as 'UML', 'Reading', 'Checklists', 'Interviews', 'Models' (e.g., 'SDL'), 'Cross-referencing', 'Simple Scenarios', and 'Mathematical Proofs' [91].

In all, to write the right requirement specification, we need to consider several attributes that are valid, complete, consistent, beneficial, necessary, unambiguous, uniquely identifiable, verifiable, understandable (clear), and modifiable. That is why we do not get these typical mistakes like noise, silence, over-specification-contradiction, ambiguity, format reference, wishful thinking, and so on in the requirement specification.

Finally, a new taxonomy of software requirement specification is illustrated in Figure 7, which is based on the above description in the three categories 'Standards', 'Templates', and 'Types'. Table 10 shows the comparison of the proposed Requirements Specification Taxonomy with the other taxonomies in those three categories.

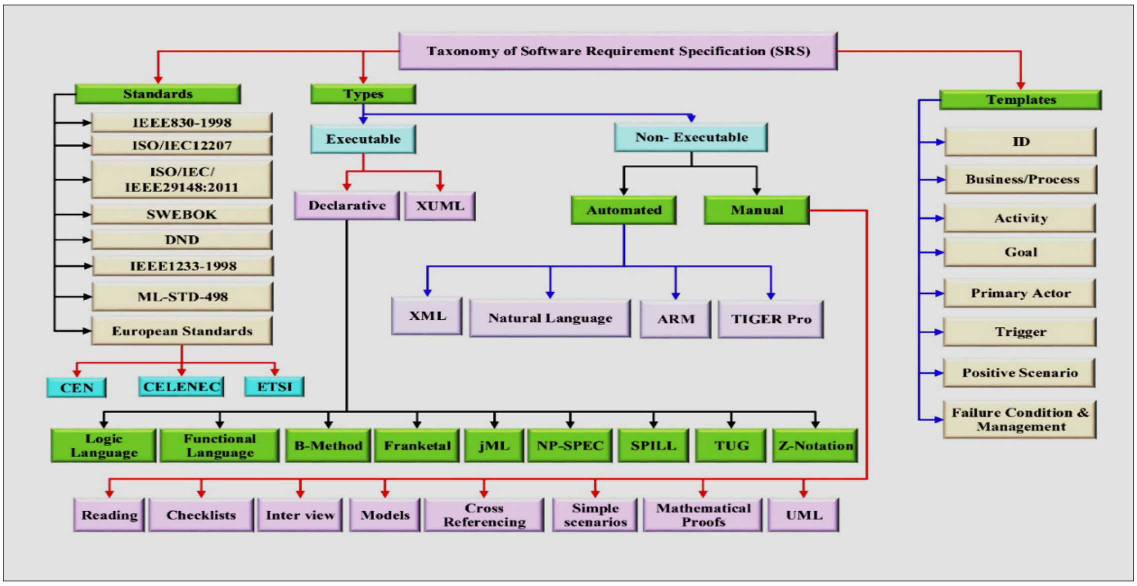

Figure 7. A New Taxonomy for Software Requirement Specification.

Table 10. Comparison of the Proposed Requirements Specification (SRS) Taxonomy with Others.

Taxonomy/ [References] (RQ1)	**Aspects**														
	Standards (RQ8)						**Types (RQ7)**		**Templates (RQ8)**						
	IEEE	SWEBOK	DND	ML-STD-498	European	ISO/IEC	Executable	Non-Executable	ID	Business/ process	Activity	Primary Actor	Goal	Positive Scenario	Trigger
Pnueli [84], Brouque et al. [3], Goldman et al. [90]	✓	✓	✓	✓	✓	✓		✓	✓		✓		✓		✓
Sequeda [91]							✓	✓							
Proposed Taxonomy							✓	✓		✓		✓		✓	✓

4.3 Requirements Verification and Validation

Another aspect that is investigated in this research is the verification and validation approaches. The main approaches in requirements validation include mathematical and logical, natural languages, UML, empirical, trigger, expert systems, and model checking [92].

According to the RE process, the requirement validation and validation (V&V) phase contains some activities. These activities comprised verifying the product's accuracy and ensuring the satisfaction of the actual needs to be expected by the stakeholders of the developed or modified software. Also, investigating SRS towards the goals and stakeholders' requirements [93] and testing approaches (i.e., UML models, ModelicalML, UCM, and time usage models [94]) are tasks considered in V&V. Also, in the stage of verifying the requirements, the product accuracy verification activities like product correctness of each step of the software development process must be performed. Additionally, traceability and compatibility of the SRS products and other software development products (design, implementation, etc.) are carried out against those specifications.

To this end, we suggest an aspect-based taxonomy for V&V in two aspects, 'Techniques' and 'Levels', shown in Figure 8. The most significant of these aspects are used 'Techniques' for requirements V&V that include 'Tracing Approaches', 'Model-Based', 'Prototyping', 'Testing Approaches', 'Formal Techniques', and 'Review and Inspection' techniques. Therefore, the 'Simple Check' and the 'Traceability Matrix' techniques are considered in the tracing approach class and FTD in the testing approach class. Besides, one of the main branches in the techniques branch is model-based techniques, which we categorize into 'Logic-

Based' and 'Behavioural Models'.

The 'Logic-Based' techniques are comprised of two classes: 'First-Order' and 'Descriptive Logic (DL)'. The former class covers the Z, VDM, and 'UML-OCL' logic methods, and the second contains the 'F-SHIN' derived from 'Fuzzy DL(F-D)' models. However, the 'Behavioral Models' have covered two diagrams: 'State Machine' and 'Interaction Diagram'. Also, the types of formal methods include the 'Z', 'FSM', profiled 'UML/OCL' (e.g., UML/MARTE/SysML fusion profile ([64], [65]), 'Petri-Nets', 'SCR', and 'RSML'.

Furthermore, the formal methods must have various functions such as 'Simulation', 'Testing', 'Checking' (including 'Completeness', 'Consistency', and 'Model Checking'), and theorem-proving methods through requirements tools -for example, IFX toolset [77]. Lastly, the review and inspection techniques encompass the 'Walkthrough', 'Checklist', 'Reading the format', 'Reading and approach(sign-off)', 'Active requires', and 'Formal inspection' techniques (e.g., 'Fagan' and 'Lightweight').

It is important to mention that at each stage during the requirements engineering process, both verification and validation (V&V) are required, which are called the "system-level". Besides, the difference between the analysis and modelling phases and the V&V is that the input of the analysis phase of the requirements is raw and incomplete. However, the V&V phase works on an SRS which is negotiated and agreed upon the domain requirements and called "specification-level" [92]. Accordingly, the second branch of our V&V taxonomy is dedicated to the V&V levels mentioned above. In conclusion, it is worth mentioning that although verifying is associated with modelling, we have put them in two separate taxonomies (see Figure 6 to Figure 7) to ease the classification of concepts.

Table 11 illustrates the comparison of the proposed Requirements V&V Taxonomy with others in the considered aspects. Our proposed taxonomy considers more aspects than others.

Table 12 summarizes the proposed taxonomies in requirements engineering development, in which the aspects considered in the various taxonomies are specified. The most important of these aspects include techniques, approaches, methods, tools, models, types, notation and templates, issues and challenges, standards and levels, as well as specific aspects in each of the phases in the requirements development. In fact, it shows that some aspects, such as techniques and methods or approaches, have received less attention in the analysis and modelling as well as the requirements specifications phases. Nevertheless, the majority of aspects are investigated in the requirements elicitation and prioritization phases, where prior research has not integrated and aggregated them.

5 Summary and Future Research

In this paper, we carried out a systematic literature review that provides an expanded view for practitioners and researchers in requirement development. Based on the review approach in this study and analysis of research aspects in requirement development, we proposed four hierarchical taxonomies in an aspect-oriented approach. The results and findings of suggested taxonomies and classification in the requirements development are summarized in four parts. These parts are Requirements Elicitation, Requirements Analysis and Modelling, Requirements Specifications, and Requirements Verification and Validation. The proposed taxonomies can be used by requirement engineers and system stakeholders to find a suitable and practical solution to requirements development problems.

For future research, the following directions can be considered:

- Requirements Elicitation: In the requirements development, one of the primary steps is to determine five different facets with an aspect-oriented structure, and the popularity of using agile methodologies. Hence, agile methods such as modern Requirement Engineering and scenario-based modelling can be considered in future works.
- Requirements Analysis and Modelling: The taxonomy of kinds of requirements analysis and modelling methods, with notation in previous models, shows that less research has been done in the field of ad-hoc and natural language notation.
- Requirements Specifications: For the integration of Requirement Engineering products and the ability to transform based on standards, feature specifications of well-defined requirements, proper utilization of templates, and avoiding common mistakes in their definition, the software requirement specification taxonomy is proposed.
- Requirements Verification and Validation (V&V): To check out the correctness of Requirement Engineering products as well as ensure the satisfaction of stakeholder expectations, techniques, challenges, and potential levels, taxonomies are created by aspect analysis structure in the V&V phase. Considering the importance of requirements verification and the frequency of research in this field, useful techniques and V&V levels are presented in the V&V taxonomy.

Furthermore, based on intended aspects for tax-

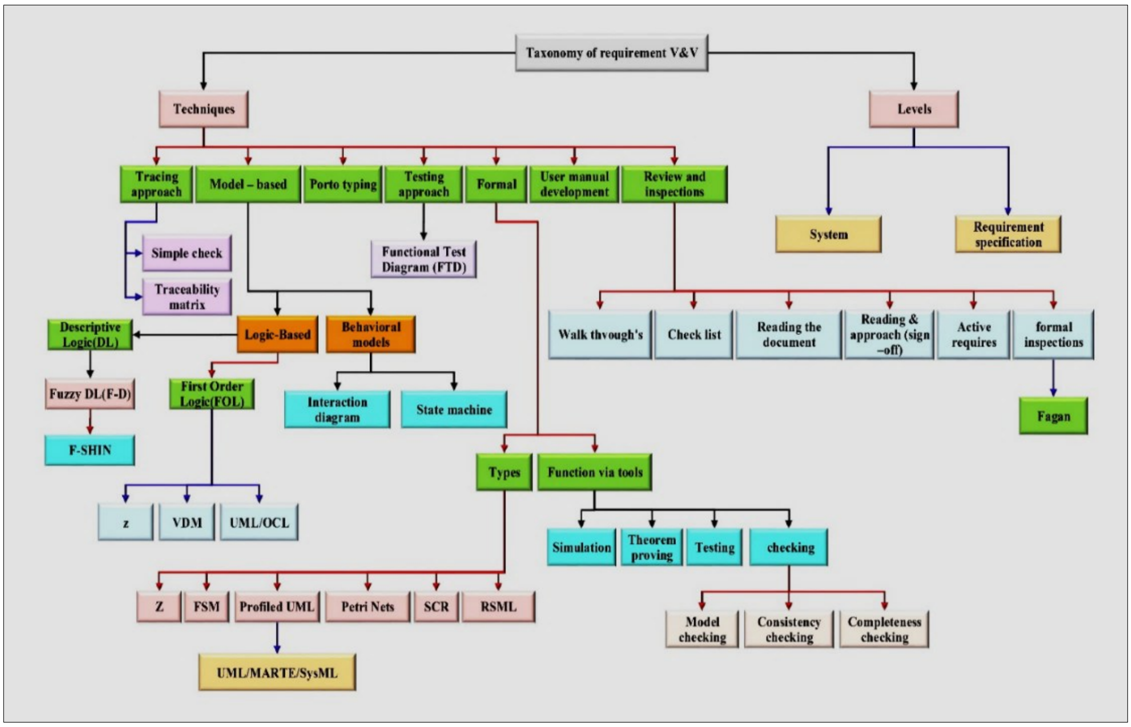

Figure 8. A New Taxonomy of Verification and Validation Aspects in RE.

Table 11. Comparison of the Proposed Requirements Verification and Validation Taxonomy with Others.

Taxonomy/ [References] (RQ1)	**Aspects**														
	Techniques (RQ3)						**Levels (RQ8)**		**Approaches (RQ4)**						
	Tracing	Model-based	Prototyping	Testing	Formal	Review and Inspection	System	Req. Specification	Math. & Logic	NLP process	UML	Expert System	Formal	Model checking	Trigger
Marques et al. [63], Gérard et al. [64]	✓	✓												✓	
Sequeda [91]								✓				✓		✓	
Manzoor et al. [77], Dos Santos [94]				✓	✓							✓			
Maalem et al. [92]			✓	✓		✓			✓	✓	✓	✓	✓	✓	✓
Proposed Taxonomy	✓	✓			✓	✓	✓	✓			✓	✓	✓		

onomies as well as defined request questions, some comparisons were made between the proposed and previous studies, and their results were presented in the tables. In future works, challenges, gaps, and research pitfalls such as scalability, reusability, security, requirements patterns, and globalization in requirement engineering can be considered to suggest appropriate and optimal solutions for these issues by new research trends in software requirements engineering.

References

[1] B. H. C. Cheng, J. M. Atlee, and M. Joanne. Research directions in requirements engineering. In *Future of Software Engineering*, pages 3–5, 2007. doi:10.1109/FOSE.2007.17.

[2] S. Vegas, N. Juristo, and V. R. Basili. Maturing software engineering knowledge through classifications: A case study on unit testing techniques. *IEEE Transactions on Software Engineering*, 35 (4):551–565, 2009. doi:10.1109/TSE.2009.13.

[3] P. Brouque and R. E. Farley. *Guide to the Software Engineering Body of Knowledge SWEBOK*

Table 12. Aspects in the Proposed Taxonomies of Requirements Development.

RE Taxonomy (RQ1)	Techniques (RQ3)	Approaches (RQ4)	Methodologies (RQ4)	Tools (RQ5)	Models (RQ6)	Types (RQ7)	Notation/Templates (RQ7 & RQ8)	Issues/Problems/Challenges (RQ2)	Standards/Levels (RQ8)	Specific (RQ1)
Elicitation & Prioritization	✓	✓	✓	✓	✓			✓		✓
Analysis & Modelling						✓	✓			✓
Requirements Specification				✓	✓	✓				✓
Verification & Validation	✓	✓		✓			✓	✓	✓	

v.3. IEEE Computer Society, 2014.

[4] L. Westfall. Software requirements engineering: What, why, who, when, and how. *ASQs Software Quality Professional Journal*, 4:9–15, 2006. doi:10.4018/978-1-4666-6026-7.ch001.

[5] K. Wiegers and J. Beatty. *Software Requirements*. Microsoft Press, 2013.

[6] C. Hokanson and K. Wiegers. *Software Requirements Essentials: Core Practices for Successful Business Analysis*. Addison-Wesley Professional, 2023.

[7] P. Skokovic and M. R. Skoković. Requirements-based testing process in practice. *International Journal of Industrial Engineering and Management*, 1(4):155–161, 2010. doi:10.24867/IJIEM-2010-4-018.

[8] ISO/IEC/IEEE. Systems and software engineering - life cycle processes - requirements. Technical report, IEEE, 2019.

[9] A. Stevenson. *Oxford Dictionary of English: Third Edition*. Oxford University Press, 2010.

[10] R. L. Glass and I. Vessey. Contemporary application-domain taxonomies. *IEEE Software*, 12(4):63–76, 1995. doi:10.1109/52.391837.

[11] G. R. Wheaton. *Development of a taxonomy of human performance: A review of classificatory systems relating to tasks and performance*. American Institute for Research, Washington DC, 1968.

[12] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering version 2.3. Technical report, Technical Report, 2007.

[13] M. Aoyama. Requirements engineering based on rebok (requirements engineering body of knowledge) and its practical guide. In *Asia-Pacific Software Engineering Conference (APSEC)*, 2012. doi:10.1109/APSEC.2012.157.

[14] C. Ayala, P. Botella, and X. Franch. Construction of a taxonomy for requirements engineering commercial-off-the-shelf components. *Requirements Engineering*, 5(2):94–99, 2005.

[15] D. Myers and T. Hathawy. On business requirements and technical specifications: A requirements taxonomy. Technical Report 813, Requirements Solutions Group, 2008.

[16] N. Afreen, A. Khatoon, and M. Sadiq. A taxonomy of software's non-functional requirements. In *Proceedings of the Second International Conference on Computing and Communication Technologies*, pages 47–53. Springer, 2016. doi:10.1007/978-81-322-2517-1_6.

[17] L. Eckhardt, A. Vogelsang, H. Femmer, and P. Mager. Challenging incompleteness of performance requirements by sentence patterns. In *24th International Requirements Engineering Conference*, pages 46–55. IEEE, 2016. doi:10.1109/RE.2016.24.

[18] J. Coughlan and R. D. Macredie. Effective communication in requirements elicitation: A comparison of methodologies. 2002. doi:10.1007/s007660200004.

[19] D. Zowghi and C. Coulin. Requirements elicitation: A survey of techniques, approaches, and tools. In *Engineering and Managing Software*

Requirements. Springer, 2005. doi:10.1007/3-540-28244-0_2.

[20] P. Rajagopal, R. Lee, T. Ahlswede, C. C. Chiang, and D. Karolak. A new approach for software requirements elicitation. In *Sixth International Conference on Software Engineering*, pages 32–42, 2005. doi:10.1109/SNPD-SAWN.2005.5.

[21] S. B. Basri, G. Kumar, F. F. Fahrurazi, E. P. B. Azmi, A. O. Balogun, and H. Mamman. Current trend of software requirement engineering process in it small and medium enterprises (smes)-a systematic literature review. In *13th International Conference on Information Technology in Asia (CITA)*, pages 82–87, Kota Samarahan, Malaysia, 2023. doi:10.1109/CITA58204.2023.10262498.

[22] L. R. Wong, D. S. Mauricio, and G. D. Rodriguez. A systematic literature review about software requirements elicitation. *Journal of Engineering Science and Technology*, 12(2):296–317, 2017.

[23] R. Ellis-Braithwaite, R. Lock, R. Dawson, and T. King. Repetition between stakeholder (user) and system requirements. *Requirements Engineering*, 2015. doi:10.1007/s00766-015-0239-x.

[24] C. Burnay. Are stakeholders the only source of information for requirements engineers? toward a taxonomy of elicitation information sources. *ACM Transactions on Management Information Systems*, 7(3):3, 2016. doi:10.1145/2965085.

[25] F. Hujainah, R. B. A. Bakar, B. Al-haimi, and M. A. Abdulgabber. Stakeholder quantification and prioritization research: A systematic literature review. *Information and Software Technology*, 102:85–99, 2018. doi:10.1016/j.infsof.2018.05.008.

[26] I. AlJassasi and Y. Baghdadi. A front-end requirement engineering framework of social commerce enterprises (design-phase). In *IEEE International Conference on e-Business Engineering (ICEBE)*, pages 231–239, Sydney, Australia, 2023. doi:10.1109/ICEBE59045.2023.00034.

[27] T. Yapa and D. Nawinna. Success factors of requirement engineering that influence software development efficiency. In *6th International Conference on Advancements in Computing (ICAC)*, pages 49–54, Colombo, Sri Lanka, 2024. doi:10.1109/ICAC64487.2024.10851100.

[28] R. Thakurta. Understanding requirement prioritization artifacts: A systematic mapping study. *Requirements Engineering*, 2016. doi:10.1007/s00766-016-0253-7.

[29] M. Sufian, Z. Khan, S. Rehman, and W. H. Butt. A systematic literature review: Software requirements prioritization techniques. In *International Conference on Frontiers of Information Technology*, pages 35–40, 2018.

[30] H. Dar, M. I. Lali, H. Ashraf, M. Ramzan, T. Amjad, and B. Shahzad. A systematic study on software requirements elicitation techniques and its challenges in mobile application development. *IEEE Access*, 6:63859–63867, 2018.

[31] M. Asif, I. Ali, M. S. A. Malik, M. H. Chaudary, S. Tayyaba, and M. T. Mahmood. Annotation of software requirements specification (srs), extractions of nonfunctional requirements, and measurement of their tradeoff. *IEEE Access*, 7:36164–36176, 2019. doi:10.1109/ACCESS.2019.2903133.

[32] D. Dermeval, J. Vilela, I. I. Bittencourt, J. Castro, S. Isotani, P. Brito, and A. Silva. Applications of ontologies in requirements engineering: A systematic review of the literature. *Requirements Engineering*, 21(4):405–437, 2016. doi:10.1007/s00766-015-0222-6.

[33] J. Smith. 13 best requirements management tools & software. urlhttps://www.guru99.com/requirement-management-tools.html, 2023. [cited 2024].

[34] B. Bals. A guide to effective requirements change management. `https://www.eqmc-consulting.de/requirements-change-management`, 2022. [cited 2024].

[35] M. Binkhonaina and L. Zhao. A machine learning approach for hierarchical classification of software requirements. *Machine Learning with Applications*, 12:1–12, 2023. doi:10.48550/arXiv.2302.12599.

[36] T. R. Agarwal and N. Thakur. Deep learning in requirement engineering: A statistical justification. In *International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS)*, pages 1–8, Bangalore, India, 2023. doi:10.1109/ICCAMS60113.2023.10525759.

[37] H. Xiang, W. Li, Y. Hong, and C. Li. A novel requirement elicitation and evaluation framework for product-service systems based on contextual matching and hybrid decision-making. *Computers & Industrial Engineering*, 194:110391, 2024. doi:10.1016/j.cie.2024.110391.

[38] S. Saleem, M. N. Asim, L. V. Elst, et al. Generative language models potential for requirement engineering applications: insights into current strengths and limitations. *Complex Intelligent Systems*, 11:278, 2025. doi:10.1007/s40747-024-01707-6.

[39] M. Yousuf and M. Asger. Comparison of various requirements elicitation techniques. *International Journal of Computer Applications*, 116(4):8–15, 2015. doi:10.5120/20322-2408.

[40] R. Agarwal and M. R. Tanniru. Knowledge acquisition using structured interviewing: An empirical investigation. *Journal of Management Information Systems*, 7(1):123–140, 1990.

doi:10.1080/07421222.1990.11517884.

[41] M. Galster and E. Bucherer. A taxonomy for identifying and specifying non-functional requirements in service-oriented development. In *Proceedings of the IEEE Congress on Services*, pages 345–352, 2008. doi:10.1109/SERVICES-1.2008.51.

[42] S. Tiwari and S. S. Rathore. A methodology for the selection of requirement elicitation techniques. *arXiv preprint*, 2017. doi:10.48550/arXiv.1709.08481.

[43] N. Garg, P. Agarwal, and S. Khan. Recent advancements in requirement elicitation and prioritization techniques. In *International Conference on Advances in Computing, Engineering and Applications*, pages 237–240, 2015. doi:10.1109/ICACEA.2015.7164702.

[44] M. I. Manzoor, M. Shaheen, H. Khalid, A. Anum, N. Hussain, and M. R. Faheem. Requirement elicitation methods for cloud providers in it industry. *International Journal of Modern Education and Computer Science*, 15:12, 2018. doi:10.5815/ijmecs.2018.10.05.

[45] M. A. Abbasi, J. Jabeen, Y. Hafeez, D. Batool, and N. Fareen. Assessment of requirement elicitation tools and techniques by various parameters. *Software Engineering*, 3(2):7–11, 2015. doi:10.11648/j.se.20150302.11.

[46] J. A. Goguen and C. Linde. Techniques for requirements elicitation. In *Proceedings of the IEEE International Symposium on Requirements Engineering*, pages 152–164, 1993. doi:10.1109/ISRE.1993.324822.

[47] G. Ruhe. Software engineering decision support: Methodology and applications. In *Innovations in Decision Support Systems*, pages 143–174. 2003. doi:10.1109/HICSS.2007.494.

[48] A. Pussep, M. Schief, T. Widjaja, P. Buxmann, and C. M. Wolf. The software value chain as an analytical framework for the software industry and its exemplary application for vertical integration measurement. In *Proceedings of AMCIS*, 2011.

[49] P. De Giacomo, R. A. Fiorini, and G. F. Santacroce. From elementary pragmatic model (epm) to evolutive elementary pragmatic model (e2pm). In *Towards a Post-Bertalanffy Systemics*, pages 135–145. Springer, 2016. doi:10.1007/978-3-319-24391-7_13.

[50] S. Lim, A. Henriksson, and J. Zdravkovic. Data-driven requirements elicitation: A systematic literature review. *SN Computer Science*, 2(1):1–35, 2021. doi:10.1007/s42979-020-00416-4.

[51] B. Tenbergen and M. Daun. Is requirements-engineering research delivering what it promised? a review of its accomplishments and opportunities after 10 years. *IEEE Software*, 36(4):6–11, 2019. doi:10.1109/MS.2019.2909127.

[52] J. Liu, J. Wu, H. Gan, T. Zhang, and C. Ma. Diversified teaching reform of software requirement engineering based on case. In *2023 5th International Conference on Computer Science and Technologies in Education (CSTE)*, pages 30–34, Xi'an, China, 2023. doi:10.1109/CSTE59648.2023.00013.

[53] H. Liao. Requirement elicitation based on value chain analysis. *Journal of Theoretical and Applied Information Technology*, 50(2), 2013.

[54] B. Davey and K. Parker. Requirements elicitation problems: A literature analysis. *Issues in Informing Science and Information Technology*, 12:71–82, 2015. doi:10.28945/2211.

[55] R. Evans, S. Park, and H. Alberts. Decisions not requirements: Decision-centered engineering of computer-based systems. In *Proceedings of the International Conference and Workshop on Engineering of Computer-Based Systems*, pages 435–442, 1997.

[56] R. D. Galliers and J. A. Swan. There's more to information systems development than structured approaches: Information requirements analysis as a socially mediated process. *Requirements Engineering*, 5(2):74–82, 2000. doi:10.1007/s007660070011.

[57] B. J. M. Abma. Evaluation of requirements management tools with support for traceability-based change impact analysis. Master's thesis, University of Twente, 2009.

[58] A. M. Hickey and A. M. Davis. Elicitation technique selection: How do experts do it? In *Proceedings of the 11th IEEE International Requirements Engineering Conference*, pages 169–178, 2003. doi:10.1109/ICRE.2003.1232748.

[59] N. A. M. Maiden. Systematic scenario walkthroughs with art-scene. In I. Alexander and N. A. M. Maiden, editors, *Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle*, pages 166–178. John Wiley, 2004.

[60] J. A. García-García, M. J. Escalona, E. Ravel, G. Rossi, and M. Urbieta. Ndt-merge: A future tool for conciliating software requirements in mde environments. In *Proceedings of the 14th International Conference on Information Integration and Web-based Applications and Services*, pages 177–186, 2012. doi:10.1145/2428736.2428765.

[61] S. Abdullahi, M. A. Zayyad, N. Yusuf, L. I. Bagiwa, A. Nura, A. Zakari, and B. Dansambo. Software requirements negotiation: A review on challenges. *International Journal of Innovative Computing*, 11(1):1–6, 2021. doi:10.11113/ijic.v11n1.264.

[62] A. Shah, M. A. Alasow, F. Sajjad, and J. J. A.

Baig. An evaluation of software requirements tools. In *Proceedings of the Eighth International Conference on Intelligent Computing and Information Systems*, pages 278–283, 2017. doi:10.1109/INTELCIS.2017.8260075.

[63] M. R. S. Marques, S. Siegert, and L. Brisolara. Integrating uml, marte and sysml to improve requirements specification and traceability in the embedded domain. In *12th IEEE International Conference on Industrial Informatics*, pages 176–181, 2014. doi:10.1109/INDIN.2014.6945504.

[64] S. Gérard and B. Selic. The uml–marte standardized profile. *IFAC Proceedings Volumes*, 41(2): 6909–6913, 2008. doi:10.3182/20080706-5-KR-1001.01171.

[65] M. U. Bokhari and S. T. Siddiqui. A comparative study of software requirements tools for secure software development. *Bharti International Journal of Information Technology (BIJIT)*, 2(2): 207–216, 2010.

[66] B. Bruegge and A. H. Dutoit. *Object-Oriented Software Engineering Using UML, Patterns, and Java*. Prentice-Hall, 3rd edition, 2012.

[67] 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*, 157:110396, 2019. doi:10.48550/arXiv.1908.07783.

[68] M. O. Ahmad, D. Dennehy, K. Conboy, and M. Oivo. Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, 137:96–113, 2018. doi:10.1016/j.jss.2017.11.045.

[69] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband. A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, 5: 915–929, 2015. doi:10.1016/j.chb.2014.10.046.

[70] N. Kano. Attractive quality and must-be quality. *Journal of the Japanese Society for Quality Control*, 14:39–48, 1984.

[71] B. Bals. A guide to effective requirements change management. 2022. URL `https://www.eqmc-consulting.de/requirements-change-management`. Accessed: 2024.

[72] A. S. Alsharari, W. M. N. Wan Zainon, S. Letchmunan, B. A. Mohammed, and M. S. Alsharari. A review of agile methods for requirement change management in web engineering. In *2023 International Conference on Smart Computing and Application (ICSCA)*, pages 1–9, Hail, Saudi Arabia, 2023. doi:10.1109/ICSCA57840.2023.10087734.

[73] P. Srivastava, N. Srivastava, R. Agarwal, and P. Singh. Incorporating requirement engineering into agile methodologies: Challenges and proposed solutions. In *2023 9th International Conference on Signal Processing and Communication (ICSC)*, pages 351–355, NOIDA, India, 2023. doi:10.1109/ICSC60394.2023.10441607.

[74] M. Kassab, C. Neill, and P. Laplante. State of practice in requirements engineering: Contemporary data. *Innovations in Systems and Software Engineering*, 2014. doi:10.1007/s11334-014-0232-4.

[75] Y. Li, R. Peng, and B. Wang. Survey on challenges and achievements in context-aware requirement modeling. *IEICE Transactions on Information and Systems*, 103(3):553–565, 2020. doi:10.1587/transinf.2019EDR0002.

[76] D. T. P. C. A. M. Janssens. Natural language processing in requirements elicitation and requirements analysis: A systematic literature review. Master's thesis, Unknown University, 2019.

[77] A. Manzoor, N. Belloir, and J. M. Bruel. Modeling and verification of functional and non-functional requirements of ambient self-adaptive systems. *Journal of Systems and Software*, 107: 50–70, 2015. doi:10.1016/j.jss.2015.05.028.

[78] M. Arif, C. W. Mohammad, and M. Sadiq. Software requirements modeling: A systematic literature review. In *IEEE International Conference on Computing, Power and Communication Technologies*, pages 194–200, 2020. doi:10.1109/GUCON48875.2020.9231058.

[79] G. Chatzikonstantinou and K. Kontogiannis. Efficient parallel reasoning on fuzzy goal models for run time requirements verification. *Software and Systems Modeling*, 17(4):1339–1364, 2018. doi:10.1007/s10270-016-0562-9.

[80] G. Ćwikła, A. Gwiazda, W. Banaś, Z. Monica, and K. Foit. Analysis of the possibility of sysml and bpmn application in formal data acquisition system description. In *IOP Conference Series: Materials Science and Engineering*, page 012034, 2017. doi:10.1088/1757-899X/227/1/012034.

[81] F. Belina and D. Hogrefe. The ccitt-specification and description language sdl. *Computer Networks and ISDN Systems*, 16(4):311–341, 1989.

[82] D. Amyot, N. Mansurov, and G. Mussbacher. Understanding existing software with use case map scenarios. In *International Workshop on System Analysis and Modeling*, pages 124–140, 2002. doi:10.1007/3-540-36573-7_9.

[83] M. Abid, D. Amyot, S. S. Somé, and G. Mussbacher. A uml profile for goal-oriented modeling. In *International SDL Forum*, pages 133–148, 2009.

[84] A. Pnueli. The temporal logics of programs. In *Proceedings of the Annual IEEE Symposium on Foundations of Computer Science*, pages 46–57.

IEEE, 1977.

[85] A. Van den Berghe, R. Scandariato, K. Yskout, and W. Joosen. Design notations for secure software: A systematic literature review. *Software and Systems Modeling*, 16(3):809–831, 2017. doi:10.1007/s10270-015-0486-9.

[86] F. Bozyiğit, Ö. Aktaş, and D. Kılınç. Linking software requirements and conceptual models: A systematic literature review. *Engineering Science and Technology, an International Journal*, 24(1): 71–82, 2021. doi:10.1016/j.jestch.2020.11.006.

[87] Z. Li, J. G. Hall, and L. Rapanotti. On the systematic transformation of requirements to specifications. *Requirements Engineering*, 19(4):397–419, 2014.

[88] N. Ali and R. Lai. A method of software requirements specification and validation for global software development. *Requirements Engineering*, 22 (2):191–214, 2015.

[89] P. Heck and A. Zaidman. A systematic literature review on quality criteria for agile requirements specifications. *Software Quality Journal*, pages 15–27, 2016.

[90] J. L. Goldman, G. Abraham, and I. Y. Song. Generating software requirements specification. In *IEEE*, pages 215–212, 2007.

[91] J. F. Sequeda. Taxonomy of verification and validation of software requirement and specifications. Technical report, 2007.

[92] S. Maalem and N. Zarour. Challenge of validation in requirements engineering. *Journal of Innovation in Digital Ecosystems*, 3(1):15–21, 2016.

[93] M. Kamalrudin and S. Sidek. A review on software requirements validation and consistency management. *International Journal of Software Engineering and Its Applications*, 9(10):39–58, 2015.

[94] J. Dos Santos, L. E. G. Martins, V. A. de Santiago Júnior, L. V. Povoa, and L. B. R. dos Santos. Software requirements testing approaches: A systematic literature review. *Requirements Engineering*, 25(3):317–337, 2020.

Hassan Rashidi is a Professor in Department of Mathematics and Computer Science of Allameh Tabataba'i University. He received the B.Sc. degree in Computer Engineering and M.Sc. degree in Systems Engineering and Planning, both from the Isfahan University of Technology, Iran. He obtained a Ph.D. from the Computer Science and Electronic System Engineering department of the University of Essex, UK. His research interests include software engineering, Decision support systems, and scheduling algorithms. He has published many research papers in International conferences and Journals.

Soheil Afraz obtained both his MSc and Ph.D. degrees from Islamic Azad University, Qazvin Branch, Iran. He currently works at the Faculty member in Computer Engineering, Islamic Azad University, Ardabil Branch. Soheil does research in Software engineering, Artificial Intelligence, Search-Based Software Engineering(SBSE), requirements engineering, multi-objective optimization, Bioinformatics and Cancer Research.

Zeynab Rashidi received her B.Sc. and M.Sc. degrees in Computer Engineering, respectively, from Alzahra University and Amirkabir University of Technology, Tehran, Iran. She obtained Ph.D. degree in educational technology, at Allameh Tabataba'i University. She has published several papers in Persian and English in journals and two books. Conducting educational workshops, teaching university courses, and collaborating on research projects are among her activities.