

Requirements Engineering for Internet of Things (IoT) Software Systems Development: A Systematic Mapping Study

José-Alfonso Aguilar-Calderón ^{1,*} , Carolina Tripp-Barba ¹ , Aníbal Zaldívar-Colado ¹ 
and Pedro-Alfonso Aguilar-Calderón ² 

¹ Facultad de Informática Mazatlán, Universidad Autónoma de Sinaloa, Mazatlán 82140, Mexico; ctrippp@uas.edu.mx (C.T.-B.); azaldivar@uas.edu.mx (A.Z.-C.)

² Escuela de Ingeniería Mazatlán, Universidad Autónoma de Sinaloa, Mazatlán 82140, Mexico; pedro_a4@uas.edu.mx

* Correspondence: ja.aguilar@uas.edu.mx; Tel.: +52-669-110-4470

Abstract: The Internet of Things (IoT) paradigm is growing, affecting human life and aiming to solve problems in the real world, i.e., in education, healthcare, smart homes, intelligent transportation, and other areas. However, it is a fact that the development of IoT systems is complicated compared to that of traditional software systems, especially in relation to requirements engineering (RE). The RE of IoT systems is not implemented frequently due to their broad aspects, such as the variety of user needs, making these systems difficult to construct. In this sense, the use of IoT-based systems has not been well explored by the research community in order to provide well-planned proposals to improve the quality of their performance. In this work, we present a comprehensive and inclusive review of the RE of IoT-based systems. To accomplish this, a systematic mapping study (SMS) is presented to evaluate the use of parameters based on the existing literature. SMS is a methodology used for research in the medical field and has recently been implemented in software engineering (SE) to sort and organize research publications to gain knowledge on progress and identify research gaps. In this article, we aim to classify the existing research publications in the current scientific literature regarding RE proposals for IoT software systems and review their implications for future research. This will make it possible to establish lines of research in order to improve the quality of the development of future IoT systems.

Keywords: systematic mapping study; requirements engineering; Internet of Things; IoT software systems development



Citation: Aguilar-Calderón, J.-A.; Tripp-Barba, C.; Zaldívar-Colado, A.; Aguilar-Calderón, P.-A. Requirements Engineering for Internet of Things (IoT) Software Systems Development: A Systematic Mapping Study. *Appl. Sci.* **2022**, *12*, 7582. <https://doi.org/10.3390/app12157582>

Academic Editors: Alberto Rodrigues Da Silva and Luis Olsina

Received: 2 July 2022
Accepted: 25 July 2022
Published: 28 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

The constant changes in the technological field have forced the emergence of new automation systems, such as IoT systems [1]. The Internet of Things (IoT) is a network of physical things that employ the Internet. The information is transmitted by the Internet to be accessed by the users at any time and in any place [2]. These connected things will permit apparatuses to become smarter, stronger, more effective, and more readily available in the future [3]. The things connected in the IoT are hardware and software systems such as sensors, micro-motors, wearable devices, the cloud, etc. This network is currently used in a variety of fields, such as logistics [4], smart cities [5], healthcare systems [6] (particularly in COVID-19 pandemic management [7]), robotics [8], image recognition [9–11], and agriculture [12].

The IoT concept is emerging and evolving rapidly. Several technical solutions for multiple purposes have been proposed for its implementation. Regrettably, the rapid evolution and utilization of IoT technologies' have resulted in a lack of user support. Furthermore, unlike independent software, the audience for IoT systems is unbounded and massive. Therefore, users most likely have distinct necessities and desires, and IoT engineers should be able to deal with these distinct needs through a detailed requirements engineering

(RE) phase, considering functional (FR) and non-functional requirements (NFRs). RE is a sub-discipline of software engineering (SE) and is concerned with establishing the goals, functionality, and constraints of hardware and software systems [13]. According to [14], RE is the part of SE that is applied to identify stakeholders' needs; they have documented an exhaustive set of phases such as analysis, specification, and validation in a requirements document. For RE, it is necessary to have synergy with customers to understand their specific needs. This activity helps the software development team collect sufficient information about the problem domain to define models for the solution domain. However, poor implementation of RE means that the final product will not satisfy the customer's expectations or, instead, it will not be used. Now, more than ever, there is an opportunity to implement well-planned and efficient IoT systems, not only on the shop floor, but with user goals and satisfaction in mind, since there is more access to IoT systems available now than ever. Nevertheless, there are great challenges involved in implementing RE. There is an absence of systematic methodological approaches to building IoT applications, especially for IoT-based RE.

Despite the required complexity of such implementations in the real world, there is no reference guide in the context of RE to assist in implementing these systems. Consequently, it is essential to establish a baseline that can serve as a starting point when developing IoT systems, in order to develop reliable proposals that can be applied to the real world. Furthermore, this will satisfy user expectations since requirements and system elements are highly interconnected [15].

Hence, in this study we have the following objectives:

- To draw the attention of the software engineering (SE) research community to the absence of empirical information in the field of RE in IoT systems;
- To provide a conceptual map that organizes the data that have been published so far and a starting point for researchers to enrich the application of RE in IoT software systems;
- To show the shortcomings regarding the application of RE in IoT systems development;
- To highlight the main contributions in RE for IoT systems development that fully implement the RE and its phases;
- To indicate the most implemented RE phases in the existing proposals; and
- To list the most commonly used techniques for the application of RE in IoT systems development.

In this work we present a systematic mapping study (SMS) [16] of the existing empirical data concerning RE implemented to date in IoT systems development. This SMS analyzes the current state of research regarding RE in IoT systems. Our primary aim was to observe the research trends, the type of studies conducted (proposal, extension, meta-studies, etc.), the research trends over time, and the research motivations that academia should focus on more in order to improve IoT systems development. This helps to bifurcate the RE research area, first identifying how much research has been conducted to date by researchers regarding RE phases, and second, predicting how much support IoT systems can obtain to improve their projects from the scientific literature. Furthermore, these research outcomes may allow researchers to analyze the current research directions in this area to identify the gaps that must be attended to by providing some articles that can be applied outside of the research literature, i.e., in industrial cases. Our final goal in this research was to provide a classification for this area, providing ideas for researchers or professionals looking for information regarding RE in IoT systems. Moreover, those classifications can support researchers conducting primary or secondary studies.

This article is structured as follows. In Section 2 we present those RE concepts relevant to this article's context. In Section 3 we outline the research method applied in this article. In Section 4 we present the results of the mapping process. In Section 6, we discuss the current study's results and limitations. We present our conclusions and some ideas for future research in Section 7.

2. Fundamentals

The Internet of Things (IoT) is a broad-spectrum concept referring to the ability of network apparatus, gadgets, or devices collecting and sharing information from our ecosystem for distinct purposes, such as (i) notification systems for supermarket discounts triggered when the customer is passing by the supermarket, (ii) day-to-day food ordering applications, (iii) the monitoring of automobile fluid levels, and (iv) heart monitoring with a holder monitor, among other devices. It is essential to mention that IoT differs from traditional computing and monitoring, mainly due to the size of the files used, which can be small and frequently transmitted. However, the number of devices that connect to the system in the IoT is more significant.

The IoT broadcasts data to users and systems/software. For example, for users, the IoT can transmit information regarding health through software that collect data from sensors monitoring vital signs. In several cases, a business or user can also manipulate a device remotely to execute tasks such as turning on the garage light or programming an office air conditioning system to refresh at 8:00 a.m. for the arrival of employees.

The application of RE phases is crucial in affecting the achievement of development projects in most software factories. RE is a method of finding, studying, detailing, and validating the services that a software should offer, as well as its operational limitations. There are countless methods to describe the phases of RE, such as those suggested in [17,18]. Of course, the phases differ for different situations, such as the application domain and the team developing the requirements. Nevertheless, some universal phases generally emerge, such as elicitation, analysis, specification, validation, and management.

- Elicitation aims to find what issues need to be solved [18], identifying the goals that a software needs to achieve. To achieve this, it is necessary to apply techniques [19] such as questionnaires, brainstorming, prototyping design, and conceptual modeling approaches such as goal-oriented languages [20,21];
- The analysis activity refers to creating models or prototypes satisfying the requirements. Moreover, this activity allows for an understanding of the organizational goals, considering the analysis of the relevant information, rules, and stakeholder communication to represent the software's functionality more accurately [21];
- The specification activity involves a description of the system's behavior. It is applied using techniques such as templates, scenarios, use cases, modeling, and natural language [22];
- Validation is performed in order to establish whether or not the software requirements are a truthful exemplification of the stakeholder requirements. To validate this, reviews and requirement traceability techniques can be used [22,23];
- The management stage involves identifying the variations in requirements that may occur during the project's development at the stakeholders' request. Among the techniques used in this stage are configuration management and version control [23];
- Non-functional requirements are aspects that a system should ensure to guarantee quality. Some of these are related to usability, privacy, availability, interoperability, and accessibility.

This classification of requirements is used throughout this SMS to provide a better understanding and ensure completeness in this research.

3. Research Method

We conducted a systematic mapping study (SMS). SMSs are mostly implemented as a beginning step to analyze primary studies or systematic reviews. The guidelines proposed in [16,24] were adopted here. An SMS provides an overview of a research field and its classification [25,26]. The aim of performing an SMS is to point out research tendencies by separating a research area into several significant areas of evidence related to the study's target. Furthermore, one aims to categorize the quantity and type of investigation and the available results. Likewise, it is important to map the occurrences of publications over time to understand trends and to categorize the conferences and journals where

research has been published. The planning activity made it possible to recognize the necessity for this review, specifying the research questions, the search string, and the inclusion/exclusion criteria.

An alternative to an SMS is a systematic literature review (SLR) [24], in which the work review phase is much more rigorous. This makes it possible to establish the state of evidence through the exhaustive extraction of quantitative data and meta-analysis studies and respond to much more specific research questions. These two types of studies are related, and their goal is to identify research opportunities. An SMS is widely considered to be a fundamental early step in determining in which explicit topics of a field it may be interesting to undertake a more detailed SLR. According to [25], an SMS is designed to structure a research area, collect and synthesize evidence, and present research findings. Furthermore, the application of an SMS allows for the detection of gaps in the current literature.

The present study was conducted based on the following steps:

- Construction of research questions;
- Collection of the data: structuring a search plan for searching for articles (identification process), then selecting the relevant articles (selection process);
- Extraction of the selected data;
- Mapping of the data by classifying and analyzing the information extracted (mapping process).

As far as we know, the literature lacks studies such as SMSs that point out the research trends of RE in the area of the IoT. Therefore, the results of this review provide research topics with regard to RE for IoT software systems to boost the level of investigation in this field and to bridge research gaps.

3.1. Definition of the Research Questions

IoT systems have become more popular during the past few years. Nonetheless, contemporary findings show elevated project failure rates [27,28]. There is a lack of knowledge regarding RE in IoT systems development. Motivated by this observation, in this article we aimed to obtain a more profound understanding of RE applied to IoT systems development. To extract detailed information, a set of five research questions (RQ) were introduced along the following lines.

- RQ-1. Are there any proposals for developing IoT software systems that cover the phases (elicitation, analysis, specification, validation, management) of requirements engineering? With this research question, we aimed to provide a quantifiable summary of the existing research trends concerning RE in IoT software system development.
- RQ-2. Which RE phases are currently addressed in the field of IoT software system development? We classified the studies according to the RE classification presented in Section 2 to answer this research question. This, in turn, will help to find probable areas that have been ignored. With this question, we also aimed to study which areas in RE in relation to IoT system development require deep research.
- RQ-3. What RE techniques have been implemented/developed within the phases of RE in the context of IoT systems development? Different techniques have been implemented and developed for each RE phase in the research associated with RE. With this question, we intended to understand which techniques have been used so far.
- RQ-4. What scientific publications have been published to address research in IoT software systems development, considering the RE phases? Studies have been published over the years on IoT in the context of software engineering. This question was proposed in order to gain knowledge on the types of studies (proposal, formalization, meta-study, implementation, or extension) that have been conducted according to the classification scheme presented in Section 3.2.

3.1.1. Search Strategy

To search for significant articles, we began by (i) identifying keywords, (ii) formulating the search strategy, and (iii) choosing publication sources. The careful choice of search

terms is essential since they limit the scope of results from each source, which should reflect the research question's scope. Therefore, the keywords were identified considering different spellings, tenses, word variants, synonyms, and related concepts, e.g., RE for requirements engineering, IoT for the Internet of Things, cyber-physical systems, and IoT software systems. Finally, the necessary search plan was structured using the content of the research questions. Following an initial search, the keywords were refined, and keywords were studied until synonyms were obtained. The initial key terms were merged using the AND logical operator, whereas for their synonyms we used the OR operator. Numerous search rounds were realized until the authors felt that the best equilibrium between precision and recall measures had been obtained. The search terms are detailed in Table 1, and these were also changed according to the synonyms used.

Table 1. Search terms used in the construction of search strings.

Terms and Synonyms Used in Search Strings
requirements, Internet of Things, software, cyber-physical systems, systems, IoT software systems, IoT software, RE, IoT systems development, requirements engineering, Internet of Things, methodology, approach, proposal, method, elicitation, validation, management, analysis, specification

Subsequently, the search process was performed to find articles in order to respond to the four research questions formulated in Section 3.1 and we excluded a manual search and gray literature. This process was chosen as it is fundamental to ensuring the accuracy and totality of the evidence. The search was carried out in the central online scientific databases, including the Web of Science, Scopus, MDPI, ACM Digital Library, IEEE Digital Library, and Google Scholar. As reported in [24], the selected databases are well-known and are commonly adopted in secondary studies in the SE. Furthermore, since the establishment of the IoT as an area of study at the beginning of the 21st century, available studies from 2011 to 2021 were included in the present research.

3.1.2. Screening of Studies for Inclusion/Exclusion Criteria

After searching for keywords in the online scientific databases, the initial search procedure returned a set of 132,327 studies, of which the title, abstract, and introduction/conclusion (when necessary) were analyzed, following the inclusion/exclusion criteria. Afterwards, the inclusion/exclusion criteria were applied to the articles by reading the full text. The criteria applied for inclusion were:

1. Peer-reviewed scientific articles written in English and published in conferences or journals up to December 2021;
2. Articles regarding IoT software systems development and RE;
3. Research proposals concerning the RE phases or activities in IoT software systems development;
4. Research in which a method is reasonably present;
5. If an article published in a journal followed the same conference study, only the journal publication would be included;
6. If a similar article was published in several sources by the same authors, only the recent/extended publication would be included.

The criteria applied for exclusion were:

1. Articles that were not related to IoT and RE;
2. Research that did not imply the evolution of any RE proposal for IoT software systems development;
3. Non-peer-reviewed articles;
4. Scientific articles of which the written language was not English;
5. Research in which a method was manifestly missing;

6. Scientific articles of which the content was a tutorial, conference poster presentation, or discussion panel.

A total of 24 research articles conformed to the criteria for the collection of primary studies (see Section 4). Thirty-four studies were rejected for being duplicated. Table 2 shows the articles selected in each phase of the research procedure.

Table 2. Primary studies obtained according to the inclusion/exclusion criteria for each source.

Source	First Search	First Exclusion	Second Exclusion	Primary Studies
Scopus	11,276	62	5	0
MDPI	62	2	2	2
ACM Digital Library	1000	14	7	3
IEEE Digital Library	989	248	46	5
Google Scholar	119,000	96	78	14
TOTAL	132,327	422	138	24

3.2. Quality Assessment

In order to filter the studies found in the search stage, in this SMS, a quality assessment step was implemented following [29]. The key idea behind this was to provide a method for the selection of studies that would contribute to the value of this research work. In this evaluation, we classified studies with the minimum quality according to two criteria. Criterion number one was to consider the minimum quality standard of the primary study if the study was a research article, the goal of the research was clear, and if there was an adequate description of the context in which the research was carried out, as well as the reported results. The second criterion, objectivity, stated that the research design was applied to the research goal. Then, the article provided detailed outcomes with reliable results and reached an acceptable conclusion. Finally, the article made significant contributions in regard to relevant criteria that could be used for practical applications in the industry. Both criteria were assessed on a checklist, verified with yes/no questions. Each article with at least one negative response on the checklist was eliminated as a minimum quality threshold was essential for this analysis.

After the quality assessment process was applied, a total of 24 primary studies were retained to analyze and extract the results in order to answer the research questions from Section 3.1.

3.3. Data Classification (Mapping Process)

To continue with the SMS, a form (table) was designed to reduce the bias involved in the process; the data collected were the title, author, publication year, journal, and DOI (document object identifier). Individually, the primary studies were classified according to the type of study scheme, as detailed in Table 3.

Once the classification scheme was defined, the significant primary studies were then organized to execute the data extraction process. Finally, the classification scheme was created using a spreadsheet. There it was possible to observe the classification of the primary studies according to the article type, i.e., a proposal, formalization, meta-study, implementation, or an extension, as well as the studies' activities per RE and techniques applied in RE.

Table 3. Type of study classification scheme.

Type of Study	Description
Proposal	An article that presents some new research, e.g., a methodology, method, and technique for RE in the IoT. The degree of novelty is not judged.
Formalization	Articles that contain formal or logical language associated with the proposal. Logical operators. Pseudo-code is not considered a formalization.
Meta-study	Articles that provide a significant overview of an existing research study. They are considered surveys and reviews.
Implementation	Articles presenting tools, plugins, or suites to improve the contribution of the work presented in the primary study.
Extension	Articles focused on concepts which had not been considered first in previous proposals.

4. Mapping Results

In this section we introduce the results of the extraction of information from the primary studies. Many publications obtained from the literature were focused on singular aspects of RE in IoT. The publications selected offered significant knowledge regarding the RQs. It is essential to highlight that the 24 primary studies offered answers to more than one RQ. Their answers were sorted according to the RE phases introduced in Section 2. The research questions were answered as follows.

RQ-1. Are there any proposals for developing IoT software systems that cover the phases (elicitation, analysis, specification, validation, management) of requirements engineering?

In order to answer this question, the classification introduced in Section 2 of this study was considered. According to the primary studies analyzed in this SMS, only one proposal was found that covered the phases of RE; this was the work presented in [30]. The authors presented the *REIoT (Requirements Engineering Technology for IoT software systems)* approach in that article. The aim of REIoT was to offer methodological, technical, and tool support for constructing requirements documents for IoT systems. The technology provides constructive processes and models that support the process phases involved in constructing a requirements document. Their proposal comprises an engineering process divided into eight phases: IoT ideation and conception, IoT procurement, IoT analysis, IoT specification, IoT verification, negotiation, IoT evaluation, and management. Although REIoT is not a complete IoT software system development methodology, it is dedicated to creating a requirements specification document covering RE activities through UML (unified modeling language) use cases.

Other proposals were found in primary studies in relation to this RQ that did not cover all the RE phases. In [31], the authors presented an integrated approach, considering RE techniques for IoT-based smart applications. This was formed over five phases of RE, applying RE techniques for a well-organized requirement management approach. This work supported four of the five phases of the RE process discussed in Section 2. The supported phases were elicitation, analysis specification, and validation. The authors in [32] presented the definition of an RE process for IoT systems. This process is an adapted version of ISO IEC/IEEE 12207:2017 processes. It supports elicitation, analysis, and specification staged of requirements engineering. The authors in [33] presented a methodology that extended the agent-oriented ACOSO-Meth methodology for the engineering of IoT systems. These authors considered the elicitation, analysis, and specification phases.

The rest of the proposals obtained from the primary studies, along with the RE phases they covered (less than three), can be observed in Table 4.

RQ-2. Which RE phases are currently addressed in the field of IoT software systems development?

The current research trends in RE in relation to IoT software systems development have addressed the elicitation, analysis, and specification phases. Moreover, we observed a trend towards the covering of non-functional requirements. As shown in Figure 1, of the primary studies, 32% of them were focused on the requirements elicitation phase, 22% on analysis, 20% on specification, and 19% on research on non-functional requirements. On the other hand, requirements validation (5%) and management (2%) were the least investigated phases in the scientific literature.

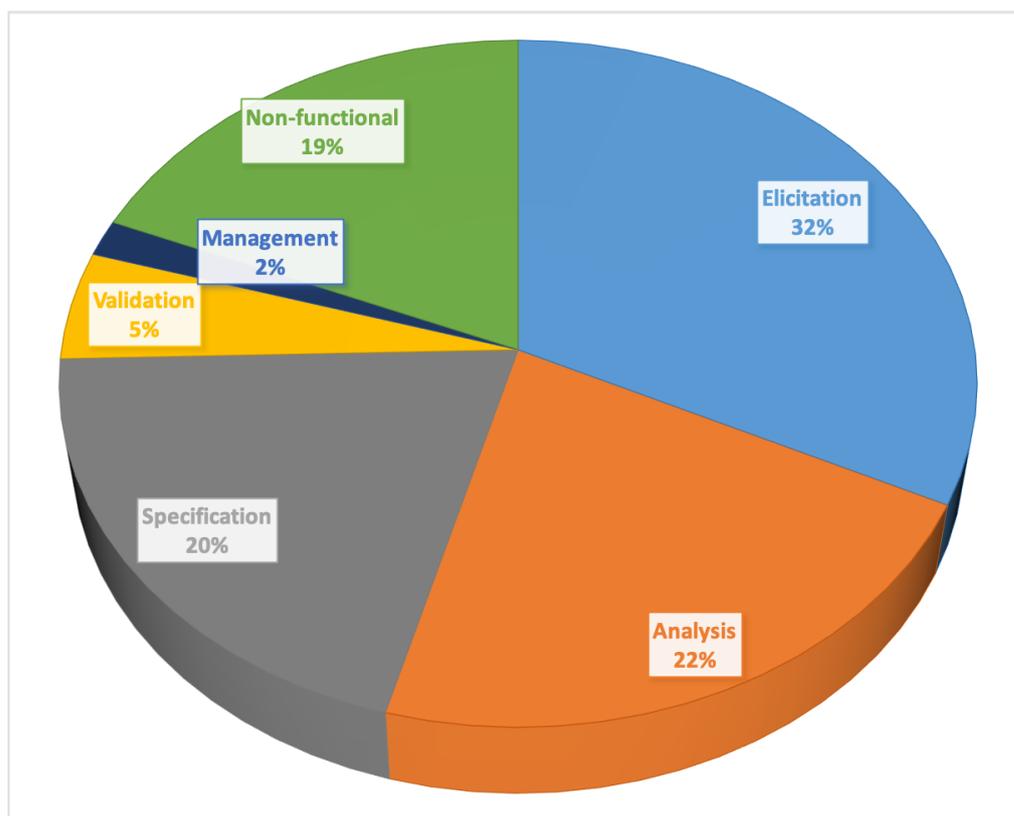


Figure 1. Classification scheme of primary studies according to the RE activities introduced in Section 2.

Table 4 presents the primary studies, classified according to the phases introduced in Section 2. The RE phase covered by each primary studied is highlighted with the symbol ✓. The last column of Table 4 shows the total score obtained by each primary study. To calculate this score, we assigned one point for each of the RE phases implemented in each primary study. The article with the highest score is highlighted in bold. Only one proposal covered the five phases for RE and supported NFRs; this was the work presented in [30], named *A Technology to Support the Building of Requirements Documents for IoT Software Systems*. The closest one to this was *An Improved RE Framework for IoT-Oriented Smart Applications using Integrated Approach* [31], which covered four phases.

RQ-3. What RE techniques have been implemented/developed within the phases of RE in the context of IoT systems development?

Considering the classification presented in Section 2, from the 24 primary studies obtained, the most applied techniques for RE were extracted. These were metamodels, UML use cases, misuse cases, interviews, surveys, workshop sessions, questionnaires, goal-oriented language, UML profiles, prototypes, UML activity diagrams, test cases, requirements specification documents, and interaction matrices.

Table 4. Classification scheme per requirements engineering phase.

Primary Study	Requirements Engineering Phase						Score
	E	A	S	V	M	N	
An Effective Security Requirements Engineering Framework for Cyber-Physical Systems	✓					✓	2
Mitigating the Impact on Users' Privacy Caused by over Specifications in the Design of IoT Applications	✓	✓				✓	3
Software Engineering for IoT-Driven Data Analytic Applications	✓		✓				2
An Application Development Framework for Internet-of-Things Service Orchestration	✓		✓				2
IoT-HarPSecA: A Framework and Road-map for Secure Design and Development of Devices and Applications in the IoT Space	✓					✓	2
Agent-Oriented Cooperative Smart Objects: From IoT System Design to Implementation	✓	✓	✓				3
A Model-Driven Methodology for the Design of Autonomic and Cognitive IoT-Based Systems: Application to Healthcare		✓	✓			✓	3
A Technology to Support the Building of Requirements Documents for IoT Software Systems	✓	✓	✓	✓	✓	✓	6
Functional Requirements Elicitation in IoT Systems: a follow-up study	✓						1
A Requirements Engineering Process for IoT Systems	✓	✓	✓				3
Elicitation Techniques for Internet of Things Applications Requirements: A Systematic Review	✓						1
COMFIT: A Development Environment for the Internet of Things	✓		✓				2
Stakeholder Identification and Use Case Representation for Internet-of-Things Applications in Healthcare	✓						1
Towards a General Software Engineering Methodology for the Internet of Things	✓						1
REUBI: A Requirements Engineering method for ubiquitous systems	✓	✓				✓	3
Modeling IoT Applications with SysML4IoT	✓						1
Specifying Functional Requirements and QoS Parameters for IoT Systems	✓	✓				✓	3
A Review of IoT Systems Engineering: Application to the Smart traffic lights system	✓	✓				✓	3
Requirements engineering methods for an Internet of Things application: fall-detection for ambient assisted living		✓	✓			✓	3
An Improved RE Framework for IoT-Oriented Smart Applications using Integrated Approach	✓	✓	✓	✓			4
TrUStAPIS: a trust requirements elicitation method for IoT	✓					✓	2
A UML-based Proposal for IoT System Requirements Specification	✓						1
Methodology for the Model-Driven Development of Service Oriented IoT Applications		✓	✓			✓	3
Papyrus for IoT—A Modeling Solution for IoT		✓	✓				2

E: elicitation, A: analysis, S: specification, V: validation, M: management, N: non-functional.

Figure 2 shows the techniques found for the RE phases in each primary study. First, there seemed to be a trend towards applying UML use cases, as 8 of 24 primary studies (33.33%) implemented this technique for some phase in RE in relation to IoT systems

development. The second most common technique applied was goal-oriented languages, with six (25%) proposals using it. Next, metamodel and UML profile techniques appeared in the third position, with five (20%) articles each. These were followed by interviews and UML activity diagrams. Finally, workshops and questionnaires appeared in two (8%) articles from the primary studies, taking the fourth position. In the end, surveys, prototypes, test cases, requirements specification documents, misuse cases, and interaction matrices were techniques that appeared only once in the results.

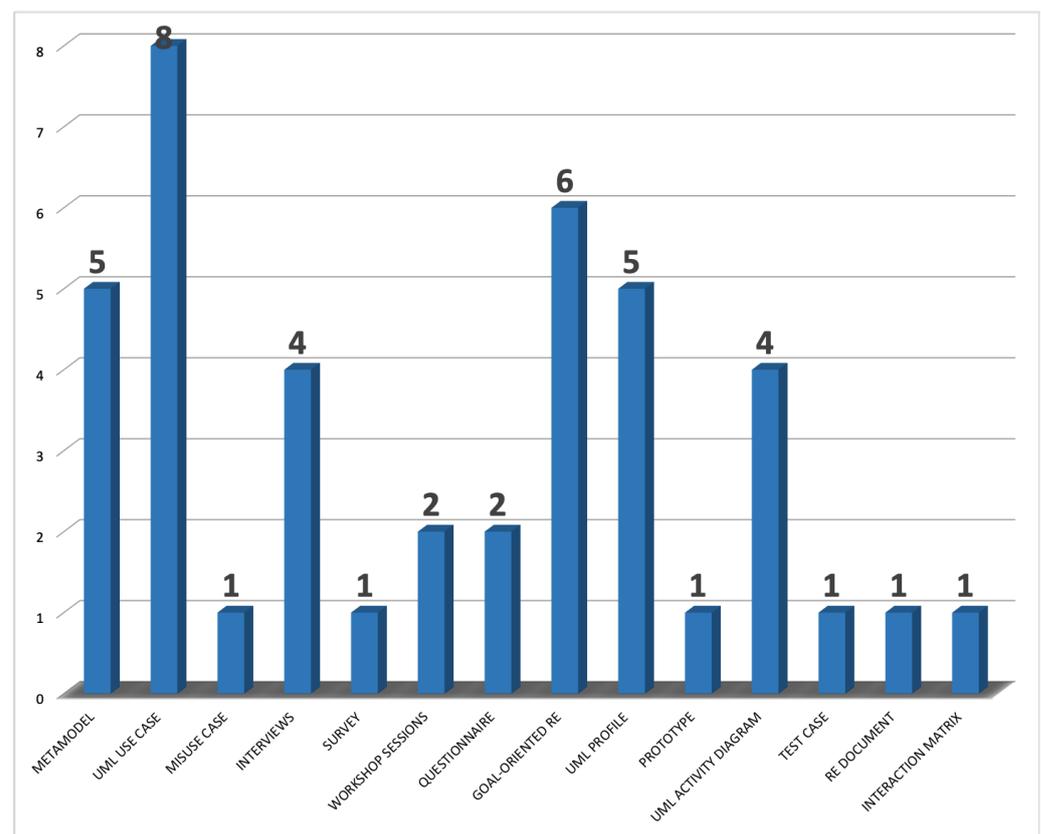


Figure 2. RE techniques applied in the proposals from the primary studies.

RQ-4. What scientific publications have been published to address research in IoT software systems development considering the RE phases?

It was of interest to gain knowledge on the type of scientific publications that have been published in recent years regarding RE for IoT software systems development. For this purpose, we present a classification by the type of study (proposal, formalization, meta-study, implementation, or extension). This classification scheme can be consulted in Table 3 from Section 3.3 of this article.

A proposal in this context is a primary study that presents a new methodology, method, technique, etc. According to Table 5, for RE of IoT, there were a total of 12 proposals for IoT software systems development, and these were [30,32,34–43].

The formalization category refers to an article that contains logical language or operators. Of these primary studies, only one fit in this category, and this was [39].

The research on IoT software systems development considering the RE phases includes meta-studies (surveys, reviews). Of the 24 primary studies, only three were found to provide a significant overview of an existing research study. These were [44–46].

Another category is implementation, referring to primary studies presenting tools, plugins, or suites. In this regard, seven articles were cataloged as implementations. These were [37,47–52].

Extension studies is the last category from the classification presented in Table 3. These primary studies present new functionalities or proposals derived from previous implementations. In this work, there were two extension studies: [33,52].

In Table 5, all 24 primary studies are listed.

Table 5. Classification scheme per primary study type.

Primary Study	Publication Type				
	P	F	M	I	E
An Effective Security Requirements Engineering Framework for Cyber-Physical Systems [47]				✓	
Mitigating the Impact on Users' Privacy Caused by over Specifications in the Design of IoT Applications [34]	✓				
Software Engineering for IoT-Driven Data Analytic Applications [35]	✓				
An Application Development Framework for Internet-of-Things Service Orchestration [48]				✓	
IoT-HarPSecA: A Framework and Road-map for Secure Design and Development of Devices and Applications in the IoT Space [49]				✓	
Agent-Oriented Cooperative Smart Objects: From IoT System Design to Implementation [33]					✓
A Model-Driven Methodology for the Design of Autonomic and Cognitive IoT-Based Systems: Application to Healthcare [36]	✓				
A Technology to Support the Building of Requirements Documents for IoT Software Systems [30]	✓				
Functional Requirements Elicitation in IoT Systems: a follow-up study [44]			✓		
A Requirements Engineering Process for IoT Systems [32]	✓				
Elicitation Techniques for Internet of Things Applications Requirements: A Systematic Review [45]			✓		
COMFIT: A Development Environment for the Internet of Things [50]				✓	
Stakeholder Identification and Use Case Representation for Internet-of-Things Applications in Healthcare [51]				✓	
Towards a General Software Engineering Methodology for the Internet of Things [53]	✓				
REUBI: A Requirements Engineering method for ubiquitous systems [37]	✓			✓	
Modeling IoT Applications with SysML4IoT [38]	✓				
Specifying Functional Requirements and QoS Parameters for IoT Systems [39]	✓	✓			
A Review of IoT Systems Engineering: Application to the Smart traffic lights system [46]			✓		
Requirements engineering methods for an Internet of Things application: fall-detection for ambient assisted living [40]	✓				
An Improved RE Framework for IoT-Oriented Smart Applications using Inetgrated Approach [31]					
TrUStAPIS: a trust requirements elicitation method for IoT [41]	✓				
A UML-based Proposal for IoT System Requirements Specification [42]	✓				
Methodology for the Model-Driven Development of Service Oriented IoT Applications [43]	✓				
Papyrus for IoT – A Modeling Solution for IoT [52]				✓	✓
Total	13	1	3	7	2

P: Proposal, F: Formalization, M: Meta-study, I: Implementation, E: Extension.

In summary, the scientific literature regarding research on RE in relation to IoT software systems development is focused on proposals (new methods, methodologies, ideas) and

implementations (frameworks, libraries, and suites). Few meta-studies, work extensions, and formalization research articles have been published.

5. Analysis

In this section we present an analysis of the primary studies found. The aim of this section is to discuss in detail the strengths and weaknesses that were analyzed in each primary study. The entire approach was related to RE in IoT software systems development. This involves more than simply answering the research questions.

In [30], the authors present RETIoT (Requirements Engineering Technology for IoT software systems), which is a software technology designed for the creation of the requirement specification documents for IoT software systems to be built. RETIoT aims to offer systematic, technical, and tool support for constructing requirements documents for IoT systems. RETIoT involves the different RE phases detailed in Section 2, adapting their activities to the reality of IoT systems. It consists of eight phases: IoT ideation and conception, IoT procurement, IoT analysis, IoT specification, IoT verification, negotiation, IoT evaluation, and management. It is essential to mention that this is not a complete methodology; it is dedicated to creating a requirements specification document covering RE activities through UML use cases, considering NFRs.

The authors in [44] present a systematic mapping study (SMS) to investigate techniques for the elicitation of functional requirements (FR) in IoT software systems. The authors aimed to classify the RE techniques, tools, and models applied for supporting the elicitation of FRs in the IoT field. The objective of this SMS was to update opinions regarding RE in IoT by including the views of experts. The authors state that this can support software quality as it addresses the gaps in traditional requirements elicitation techniques in the IoT field. Regrettably, that work only focuses on elicitation, whereas the present work studies all RE activities with the taxonomy that is presented in Section 2. Moreover, it presents an analysis of the primary studies found and suggestions for future research.

In [32] the authors present the definition of an RE process for IoT systems. This process is an adapted and corresponding version of ISO IEC/IEEE 12207:2017. The RE process begins with identifying the context of the use of the IoT system, obtaining the stakeholders' requirements, prioritizing the requirements obtained, and elaborating the final list of the stakeholders' requirements. The RE guide detailed in this work is a preliminary one. In this regard, it requires application for its assessment in real-world projects in order to correct its deficiencies. The RE techniques are based on scenarios and UML use-cases.

In the work presented in [47], the authors focus on the frameworks for security requirements in IoT cyber-physical systems (CPSs). They state that there is currently no framework for this task. They start with the assumption that the developer and the requirements analyzed in this type of system must broaden their analysis vision and consider the hardware perspective. This includes network security and the sensors that will be used. The author argues that most proposals for software engineering processes in this system focus only on the software part. The authors thus propose and evaluate an incremental security requirements evolution approach, configuring several components of different types to generate a secure system. To achieve this, they build a CPS RE framework for security, solving the problem of eliciting security requirements for different CPS sensors, gadgets, and components. Ultimately, the proposal only focuses on eliciting security requirements, rather than considering the RE phases detailed in Section 2. It is a framework for security requirements in CPS. It is not a requirements engineering methodology or an IoT software systems development proposal.

The primary study introduced in [34] is focused on privacy. The authors point out that there are several approaches to helping professionals to analyze system privacy in the design phase. Nevertheless, it is a difficult task, particularly in IoT scenarios. In this sense, the introduction of over-specification issues throughout the system development process can convert privacy into a serious problem. In this work, the authors performed a controlled experiment with students who analyzed privacy implications using goal-oriented analysis. Some points can be highlighted from this study, for example, this work

mentioned RE, but from an NFR perspective only. Thus, it is not a RE methodology or proposal for IoT software systems development. Moreover, the authors did not consider the requirements engineering phases from Section 2 in their entirety. Likewise, the study presents a controlled experiment with students. One drawback of controlled experiments is that they lack external validity (which means that their results may not be generalizable to real-world settings).

In [35], the authors propose a framework to evaluate software processes, methods, and other artifacts of software engineering in the design of applications (software) for data analytics driven by the Internet of Things (IoT-DA). Likewise, they carried out an SMS of 16 evaluations (from inside and outside of academia) of software engineering for IoT-DA to compare them and apply their proposal in a case study to demonstrate the development of an IoT-DA healthcare application. This proposal is not entirely focused on requirements engineering. However, in its framework, they consider a phase called software/systems requirements, in which they consider the elicitation of requirements to support practices such as methods, tools, and technologies for identifying the FRs and NFRs of the software. To accomplish this, they suggest the use of ThingsML for (i) the abstraction of hardware and software mapping, (ii) the interaction of things, and (iii) the specification of non-functional properties for IoTs.

In [48], the authors propose an IoT application development framework called IADev. It is based on the RestFUL architecture and the model-driven development (MDD) approach. The article is not focused entirely on RE for IoT. IADev uses an attribute-based design to transform requirements into a solution architecture by considering the concerns of all the stakeholders involved. Then, the authors define a set of MDD metamodels for transforming design models into software artifacts. Regarding RE, they define a metamodel called ADDReq, used for obtaining, modeling, and transforming the requirements to identify the components of the system. In this framework, the requirements are prioritized according to the goals proposed by the stakeholders. The requirements in this phase are re-prioritized according to their impacts on the architecture, which can be categorized as high, low, and medium. Then, an impact analysis of the requirements is applied, and the requirements are prioritized in ascending order of relevance. A list of prioritized requirements is generated. Subsequently, with this list of requirements, the authors develop a metamodel, which refers to obtaining, modeling, and transforming the requirements in order to identify the components of the system. Some issues to highlight from this work are that it does not consider the activities of requirements engineering in its entirety, only covering elicitation. It is not a requirements engineering methodology or proposal for IoT software systems development; it is a framework for IoT software systems development.

The authors in [49] present a framework called IoT-HarPSecA, which provides assistance through a roadmap in designing and developing IoT devices and applications. Although security requirements are generally considered NFRs, they represent a significant characteristic that prescribes what a system or application requires to obtain a particular security goal. In this framework, the authors consider the elicitation of requirements. They define a functional module that consists of three components: a user response analyzer, a security requirement selector, and a requirement aggregator. The security requirement selector uses this information to determine an appropriate one from the database's set of security requirements. This proposal does not consider the RE phases in their entirety. It is not a requirements engineering methodology or proposal for IoT software systems development. It is a framework for IoT software systems development focused on security requirements.

This proposal presented in [33] extends the agent-oriented ACOSO-Meth methodology for the engineering of IoT systems. In the analysis phase, ACOSO-Meth provides a high-level OS metamodel that shares core features with IoT standards, such as the IEEE P2430, AIOTI, and IoT-A domain models. In the design phase, the agent-oriented harassment-based OS metamodel derived from the high-level OS metamodel is implemented. The OS metamodel is based on ACOSO assistance agent-based modeling of functional system components, relationships, and interactions. As a methodology, in the requirements phase, it considers the elicitation, analysis, and design activities. No mention is made of the

consideration of NFRs. The mentioned RE activities are performed through metamodels, where the functionality of the IoT software system to build is specified and analyzed.

The authors in [36] propose a model-based methodology (model-driven) to refine the FRs and NFRs of the system gradually. Their methodology is based on a set of autonomous cognitive design patterns, reducing the complexity involved in the design of intelligent IoT-based systems, considering extensive data management and scalability. The authors define a set of patterns to develop a flexible cognitive monitoring system to manage patient health using different wearable devices as a proof-of-concept to define their methodology. Regarding RE, the authors propose two main phases; the first is called requirements identification, and the second is requirements formalization. The first phase is based on discussions with domain experts to obtain the system's functionality and identify NFRs. This is an iterative process in which RE is specified in UML use-cases and refined; although this is performed without specifying any implementation details. The second phase is focused on formalizing and structuring the identified requirements into concrete models that describe the interactions of the system processes.

The goal of another SLR [45] was to identify the elicitation techniques trends that have been used in implementing the IoT. The findings from this SLR revealed that interview and prototype techniques were the most commonly applied elicitation techniques for the development of IoT applications, according to the authors. Unfortunately, this study was conducted in 2018, so its results need to be updated. In addition, an electronic search was included in this SLR in two ACM and IEEE databases, without obtaining results in the latter. Therefore, the scope of this study is limited compared to the work presented in the present SMS. Furthermore, the authors focused only on requirements elicitation, leaving aside the other RE phases described in Section 2 of this article.

The authors in [50] introduce COMFIT, a model-based, cloud-based integrated development environment for IoT. COMFIT is made up of two modules, the first of which is designed for application development and infrastructure based on a model-driven architecture (MDA). The second module is for application management and execution, and it contains a cloud-based web interface connected to a server with compilers and simulators for the development of IoT applications. The MDD development process of this proposal is integrated into nine activities. As far as RE is concerned, requirements analysis is, in fact, considered as the traditional phase of requirements elicitation (collecting requirements from stakeholders; see Section 2). The artifacts produced in this activity are the first step (obligation) required to continue to the following activities, as they document all the application requirements. The authors use metamodels, UML Profiles, and UML activity diagrams to model the application's functionality. However, this work exhibits a contradiction because the authors use an MDA process, which begins with the computational-independent model (CIM), in which the requirements of the software to be developed are specified, but do not use this first level of the MDA. Instead, they begin with the platform-independent model (PIM) level, and they use an Eclipse MARS plugin called Papyrus (<https://www.eclipse.org/papyrus/> accessed on 15 May 2022), which is used to generate UML diagrams.

In the work of [51], the author describes a systematic approach to starting the requirements elicitation process for an IoT system. This approach is only designed to support the activities in an emergency room in a hospital. For requirements specifications, they use UML use case diagrams. The proposal is not a complete development methodology; it is only designed for the software system requirements elicitation phase and subsequent architectural design. It focuses only on an emergency room software system but also shows how this proposal can be used for the identification of system and stakeholder boundaries in order for its implementation to be extended to IoT solutions for the full management of a hospital.

In [53], the author presents some guidelines for a methodology for designing and developing IoT systems. Regrettably, the proposal is unfinished and only describes guidelines for two phases. These phases are analysis and design. The first includes the activities for actors, infrastructure, and functionality requirements analysis. RE is considered in this

study, including identifying goals, policies, and functions, and defining whether the goals and policies are local or global. This requirements elicitation activity may primarily involve global managers, who have the right to decide what services to provide to end-users, and local managers can set local goals and policies. The proposal does not specify tools or techniques for this phase in the development process. It simply offers a conceptualization.

In another primary study [37], a RE method used for the analysis of ubiquitous systems, called REUBI, is presented. Ubiquitous systems are also known as pervasive computing. For several years, this has also been referred to as ambient intelligence. From the actor's elements perspective, it is also known as IoT elements. REUBI is a goal-based approach, used to represent the influence of context and adverse situations. It provides an evaluation procedure to help in decision-making regarding goal satisfaction. According to the authors, this hybrid method (goal-centered and scenario-assisted) provides mechanisms for the decomposition of goals into subgoals, employing different scenarios. A set of UML stereotypes are defined to enable the representation of requirements, eventually leading to complex architectural and design decisions. The RE phases covered are elicitation and analysis. The techniques used are metamodels, UML use-cases, UML profiles, and scenarios, and the study considers NFRs such as usability.

In another study, the authors present an MDD methodology for developing IoT applications called the IDeA-IoT DevProcess & AppFramework [38]. This is focused on the design phase for the development of IoT applications. IDeA-IoT consists of a method and a tool. The authors define a UML profile based on the SysML profile named SysML4IoT. The requirements phase is focused on satisfying the goals of stakeholders. This task is performed as part of the "analyze system requirements" activity. In this stage, the IoT requirements engineer specifies the requirements for the IoT application as a "black box", as well the device requirements. This is carried out in order to define which devices are truly needed, considering required characteristics such as data formats and protocols. They are modeled using SysML requirement diagrams and requirement tables. The technique these authors use for RE is UML use-case diagrams. They focus on eliciting requirements. IDeA does not provide methods to deal with the dynamic evolution of IoT systems or for the security concerns of systems because it does not allow the user to represent different threats to systems.

IoT-RML [39] is a modeling language that provides a means to devise requirement models for IoT systems. As far as RE is concerned, requirements are obtained from stakeholders and expressed as propositions. Requirements can be FRs or NFRs. FRs are considered with reference to one of the following parameters: detected variable, location, sending rate, and detection rate. Nonetheless, NFRs are represented as quality of service (QoS) parameters, which are measured based on specific metrics. Each metric has specific parameters. The authors define an IoT requirements domain model. IoT-RML is designed to be used for requirements specification and analysis. Conflicts among requirements are also considered.

In the work presented in [46], the authors had the goal of analyzing the different characteristics of IoT systems and then studying the IDeA (IoT DevProcess & AppFramework), AgileIoT & Duttile, and design methodology (DM) methodologies used to design them. The IDeA methodology was adopted more than the other two approaches for the design and development of IoT systems. Furthermore, the authors contribute to the extension of the IDeA domain model so that it can support security aspects and mechanisms to make IoT systems secure. This proposal will be considered for systematic mapping because they performed a comparative analysis of three proposals for the development of IoT systems, of which IDeA considers RE activities.

In another primary study [40], the authors propose a hybrid requirement modeling approach for an IoT application for ambient assisted living. The proposal covers the identification of requirements, modeling, design, and implementation phases. They propose a combination of different techniques to achieve this, such as templates for the documentation of requirements, UML use-case diagrams, SysML diagrams for specification, and SysML block definition diagrams for system design. Regrettably, this proposal is only

for the development of IoT systems for ambient assisting living, and it is not a general methodology. The approach was recently created. It does not cover all phases of RE, but the authors remark that NFRs are covered by SysML models.

In [31] the authors propose an approach associated with emerging RE techniques in the area of IoT-based smart applications. Regarding RE, different techniques are used for the phases involved in the RE process as techniques for stakeholder analysis, elicitation, analysis, specification, and validation. RE is carried out through five steps. First, the stakeholder analysis phase is carried out through the goal-question-metric method, taking into account distinct questions. Second, requirements elicitation is performed by means of a socio-technical method. Third, the analysis is performed using an interaction matrix (an interaction matrix finds interactions among several requirements and reveals conflicts). Fourth, interactive workshops are integrated with this RE framework for the specification of requirements and documentation. Finally, the validation is carried out by means of test-case design techniques.

The primary study in [41] presents a method for the elicitation of requirements focused mainly on usability, security, and privacy, called TrustAPIS. Its goal is to increase the level of confidence in developing IoT projects. TrUStAPIS is an acronym that comes from using the first letters of each of the seven domains considered: trust, usability, security, availability, privacy, identity, and safety. RE uses goal-oriented modeling languages and JavaScript-based templates to elicit the requirements. It only provides support for elicitation but also considers NFRs and traceability support.

The authors in [42] present an initial version of a proposal called IoTReq. This method is proposed for the elicitation and specification of requirements for IoT systems. To achieve this, they model the domain using the service-oriented paradigm, combined with UML. The model is later extended to add the final goals, called strategic goals, which are later divided into operative goals. IoTReq also provides support for specifying NFRs in a domain model.

Another approach [43] is based on the use of conceptual models with different levels of abstraction. This allows one to generate artifacts that operate directly on the architecture. To achieve this, the system uses the model-driven development approach. Regarding RE, the authors propose a model called the business requirements model, in which the problem domain is analyzed. The authors consider FRs and NFRs. As techniques to achieve this, their method uses UML through use-cases and activity diagrams. Transformations are performed from the second layer of model-driven architecture, the platform-independent model (PIM).

Another proposal [52] extends the Papyrus modeling environment, which is widely used in the model-driven development. They combine the Smart, Safe, and Security Software Development and Execution Platform (S3P) to create Papyrus for IoT. This proposal is used to design an IoT system that will be developed. As it is used for the initial stages of development, the system involves the implementation of UML use-case diagrams and SysML diagrams as techniques for RE. It does not mention support or consideration for NFRs.

In summary, in this section the 24 primary studies collected in this SMS have been described one by one. The proposals considered some of the phases of RE, but none considered RE in its entirety. In the following section (Section 6) we present a discussion of the analysis presented in this section.

6. Discussion

This SMS has provided some insights regarding research trends in RE for IoT software systems. These insights are reviewed in this section.

Recently, particularly in the period of time investigated in this SMS, existing RE techniques have been combined, providing a few implementation frameworks for application development.

A recently published roadmap [54] regarding IoT in SE research and development pointed out that research has mostly concentrated on tackling issues in the development of IoT regarding smart cities. Nevertheless, SE approaches to IoT have focused principally on

modeling and implementation efforts, although they have overlooked the process-centric proposals and engineering life-cycles of IoT systems.

Since the IoT is a one-off class of systems that combines hardware components with software services, the main point regarding the effective use of SE approaches to IoT systems is the implementation of abstractions to develop this kind of system [55]. The use of abstractions must be considered since they can encapsulate RE phases such as the analysis, design, and development phases of IoTs, considering aspects of their architecture, frameworks, and tools. This will result in an engineering development process for IoT systems and software. In this regard, RE proposals must be evaluated in detail to help researchers and professionals understand the challenges that RE applications face in developing IoT systems. The aspects that must be analyzed in depth include support tools, frameworks, automation processes, the needs of this type of system user, and the most common algorithms used in machine learning. This will allow for the development of new-generation methodologies, considering the phases of RE in the development of IoT systems.

IoT industrial solutions indicate that IoT systems have particular requirements that necessitate personalized processes with assistance provided for the diversity of devices, such as fault tolerance and the algorithmic manipulation of sensors. However, an insufficient number of research works is now dedicated to defining methodological approaches and middle-ware for developing IoT software systems, with a shortage of methods and tools for the construction of IoT software systems. This proposals allows stakeholders to adequately describe the IoT environment. Nowadays, methods, methodologies, and tools are a basic element and are mandatory in guiding the development, design, and simulation environments for IoT systems. Thus, methodologies and tools to guide the development of IoT software systems are exciting research areas that should be deeply studied in the literature.

The application of a methodology is extensively documented as an essential practice in every system development process. This is especially true because it has been extensively reported that the unmethodical application of complicated techniques, methods, and frameworks will most likely decrease effectiveness, increase development time, and tend to make systems error-prone. Therefore, the necessity of a complete development methodology, considering the phases of RE, is crucial in IoT system development, which entails specific requirements.

The RE phases can differ according to the application domain, the stakeholders concerned, the processes, and the organizational culture. Nonetheless, we can observe some recurring phases and activities of RE, including elicitation, specification, analysis, validation, and management. In the analysis phase, it is essential to consider the existence of autonomously intelligent devices that can be used to accomplish some tasks. According to its environment, these activities could involve, for instance, monitoring and turning machinery on or off.

In the context of NFRs, according to the primary studies, limited proposals have been made to solve the problems that permanently exist in this kind of development—these are related to security in industrial IoT platforms. This may be because personalized security requirements in these projects increase the project's budget and the level of difficulty for different platforms. Nevertheless, of the 24 primary studies analyzed in this SMS, 11 provided support for at least one non-functional requirement. The most frequently investigated non-functional requirements in the primary studies were security, privacy, maintainability, performance, scalability, interoperability, and safety.

A bibliometric analysis was carried out, considering the terms detailed in Table 1. These terms were used in the search strings applied in each data source in this SMS. The analysis was performed using Vosviewer (<https://www.vosviewer.com> accessed on 13 January 2022), which is a software designed to analyze scientific productivity during a period of time. Moreover, in order to identify trends in the literature regarding RE and IoT, an initial analysis of co-occurrences of keywords was performed, considering publications with a minimum of five incidences. This resulted in six clusters (see Figure 3), involving 68

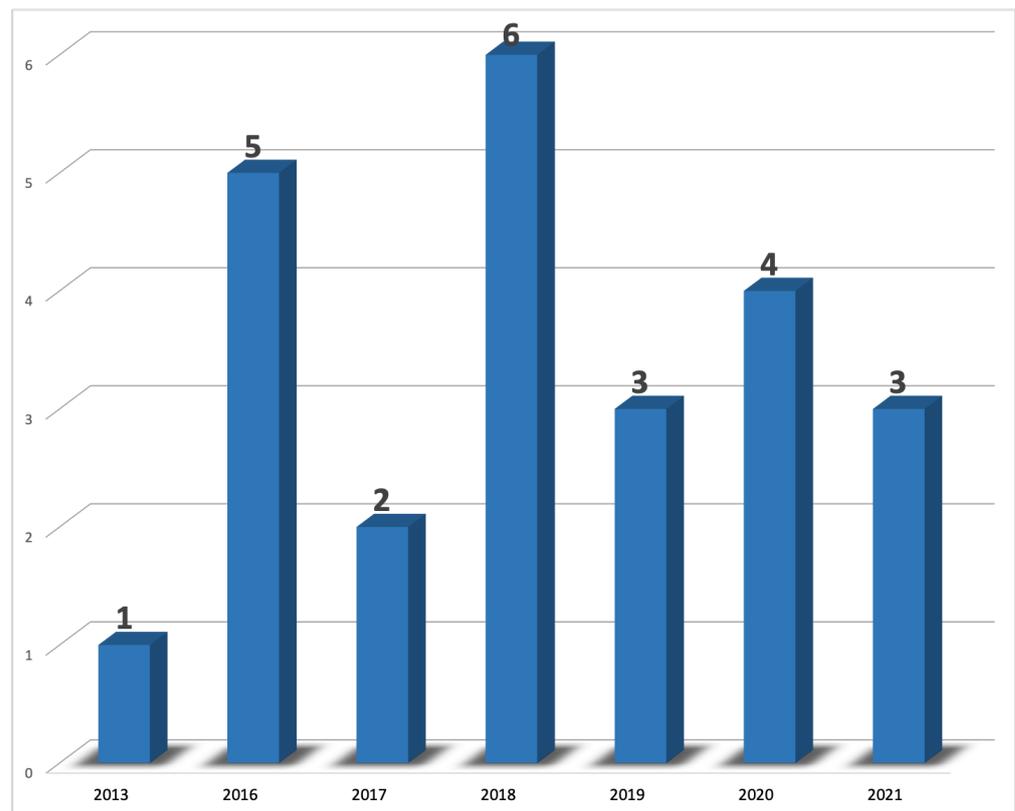


Figure 4. Primary studies published by year.

The effect of not providing support for RE in developing IoT software systems lies in the lack of knowledge regarding the challenges involved in its implementation. The proliferation of IoT in daily life is growing, and nowadays aspects of machine learning have also been incorporated into IoT systems. This implies the need for support in the form of tools and frameworks that can assist in the development process, considering the fundamentally heterogeneous audience of this type of system, that is, users with diverse profiles. The effect of not having new generations of methodologies that consider RE with sufficient weight at the beginning of the development process will result in industrial IoT solutions (with different types of devices) that do not fully meet user expectations. Therefore, the quality of these solutions in the industry will be affected. In this regard, NFRs have been barely considered, with research mostly focused on security, whereas quality aspects such as interoperability and responsiveness have been left aside. These quality attributes must be considered more closely since, due to the development of machine learning, IoT systems include autonomously intelligent devices.

7. Conclusions

This article presents the outcomes achieved after performing an SMS. Our goal was to comprehensively review the current state of the art in the field and provide a synthesis of our findings with regard to RE activities in IoT software systems development. As a result, a total of 132,327 articles in the literature were obtained from essential scientific data sources. A total of 24 articles (primary studies) were finally studied in depth. The findings of this SMS clearly show that proposals have not been well-defined and created to satisfactorily conduct the development of IoT software systems through the application of RE phases. Moreover, the literature has not placed much relevance on this topic, and the techniques implemented have been inadequately applied.

The significance of the comprehensive and accurate specification of requirements has been widely recognized because it aids in achieving an understanding with clients concerning software functionality and the central concerns in the development process.

Nevertheless, according to the results of this SMS, the phases of RE have not implemented in their entirety in IoT projects, primarily due to the particular features of the IoT, the multidisciplinary development groups involved, and the short periods required to release products into the market. Furthermore, IoT software systems can no longer be considered standard because of the diverse users accessing these devices or systems. Hence, offering proposals within the RE in the IoT field is necessary in order to improve the construction of this kind of solution, bearing in mind the large and dissimilar user population, as well as their requirements, desires, and goals considering the RE phases and NFRs.

In this SMS, we also assessed trends over time. Our conclusions demonstrate a lack of proliferation of articles with new ideas in this field. We observed the continued application of traditional RE techniques, especially UML use-cases, metamodels, misuse cases, interviews, surveys, workshop sessions, questionnaires, goal-modeling oriented language, UML profiles, prototypes, UML activity diagrams, test cases, requirement specification documents, and interaction matrices. Nevertheless, we did not observe an increase in the number of articles building upon past work (implementations, integration, and extensions). Only two publications were extensions (improvements) of previous works. Overall, interest in RE for IoT systems has not increased.

The definition of a complete SE methodology, considering the phases of RE, is a very complex issue and must be based on many authentic pieces of evidence. Furthermore, it must be accompanied by a good set of models and tools to represent and produce results, according to the fundamental principles of the paradigm. These conceptual and software artifacts will eventually lead to the development of the final product.

To conclude, we found that 13 proposals considered some RE phases but only for a single type of IoT system, such as healthcare, disaster management, control, or building automation. Nevertheless, there were no generic proposals for use within the software development methodology for an IoT system that covered the generic RE phases of elicitation, analysis, specification, validation, and management. In this regard, as far as we know, only one proposal exists that covers all the RE phases described in Section 2; this was the primary study presented in [30], but it focused only on the development of a requirements specification document.

Finally, we recommend some guidelines for future research obtained from our results.

1. The conceptualization of an approach to be used within a software development methodology for IoT systems, covering the generic RE phases of elicitation, analysis, specification, validation and management, is required;
2. There is a lack of support in the form of a methodological approach to guide the conceptual design of IoT software systems considering RE;
3. There is a need for an approach, within the phases of RE, to identify stakeholders, providing descriptions and specific goals, including their business needs;
4. There is a need for a detailed analysis model with agreements with the client for use in the initial phases of the development process and in the RE validation phase;
5. Concerning NFRs, security and privacy requirements are often not handled properly because of their extensive diversity of aspects. Thus, this area is particularly difficult to express and analyze;
6. We propose considering more RE phases supporting NFRs, such as usability, accessibility, and scalability, in addition to those covered in the primary studies included in this SMS, such as security, privacy, maintainability, and interoperability;
7. It is necessary to provide assistance for further research considering architectures and full solutions for IoT systems to be used outside of some very particular domains;
8. Since IoT software systems must be continually maintained and modified to satisfy requirements that were not anticipated at the time of analysis and design, novel methods need to be proposed to support specific software adaptability, scalability, and maintainability.

Due to the gaps detected in this SMS, in our future research, we intend to undertake the conceptual design of a model-driven proposal in which the phases of RE can be specified

and analyzed through a goal-oriented requirements analysis model. This will serve as the first step in defining an approach to the software system development process for the IoT.

Author Contributions: Conceptualization, J.-A.A.-C. and C.T.-B.; methodology, J.-A.A.-C., C.T.-B., A.Z.-C. and P.-A.A.-C.; formal analysis, J.-A.A.-C. and C.T.-B.; investigation, J.-A.A.-C., C.T.-B., A.Z.-C. and P.-A.A.-C.; writing—original draft preparation, J.-A.A.-C. and C.T.-B.; writing—review and editing, J.-A.A.-C., C.T.-B., A.Z.-C. and P.-A.A.-C.; supervision, J.-A.A.-C. and C.T.-B. All authors have read and agreed to the published version of the manuscript.

Funding: This work has partially been supported by the Universidad Autónoma de Sinaloa (UAS, Mexico) through the PROFAPI projects: Aguilar-Calderón (PRO-A8-042), Tripp-Barba (PRO-A8-033), Zaldívar-Colado (PRO-A4-021) and Aguilar-Calderón (PRO-A8-046).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The classification scheme used can be consulted in the cloud (<https://doi.org/10.6084/m9.figshare.20137160> accessed on 13 June 2022), where it is possible to see the classification scheme for the primary studies, according to the article type—a proposal, formalization, meta-study, implementation, or an extension—as well as according to RE activities and the techniques applied in RE.

Acknowledgments: The authors would like to thank the anonymous reviewers for the constructive comments, which helped improve this article. Special thanks to Universidad Autónoma de Sinaloa (UAS, México) and the support of the Cuerpo Académico Tecnología Educativa I+D+I (UAS-CA-303).

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

References

1. Gangoiti, U.; López, A.; Armentia, A.; Estévez, E.; Marcos, M. Model-Driven Design and Development of Flexible Automated Production Control Configurations for Industry 4.0. *Appl. Sci.* **2021**, *11*, 2319. [[CrossRef](#)]
2. Majumdar, A.K. Chapter 5—All-Optical Broadband Global Communications for Internet Connectivity: Free-Space Optic Links and Optical Network Architectures. In *Optical Wireless Communications for Broadband Global Internet Connectivity*; Majumdar, A.K., Ed.; Elsevier: Amsterdam, The Netherlands, 2019; pp. 117–168. [[CrossRef](#)]
3. Cubo, J.; Nieto, A.; Pimentel, E. A Cloud-Based Internet of Things Platform for Ambient Assisted Living. *Sensors* **2014**, *14*, 14070–14105. [[CrossRef](#)]
4. Tang, Y.M.; Chau, K.Y.; Xu, D.; Liu, X. Consumer perceptions to support IoT based smart parcel locker logistics in China. *J. Retail. Consum. Serv.* **2021**, *62*, 102659. [[CrossRef](#)]
5. Syed, A.S.; Sierra-Sosa, D.; Kumar, A.; Elmaghraby, A. IoT in Smart Cities: A Survey of Technologies, Practices and Challenges. *Smart Cities* **2021**, *4*, 429–475. [[CrossRef](#)]
6. Verdejo Espinosa, Á.; Lopez Ruiz, J.; Mata Mata, F.; Estevez, M.E. Application of IoT in Healthcare: Keys to Implementation of the Sustainable Development Goals. *Sensors* **2021**, *21*, 2330. [[CrossRef](#)]
7. Alam, F.; Almaghthawi, A.; Katib, I.; Albeshri, A.; Mehmood, R. iResponse: An AI and IoT-Enabled Framework for Autonomous COVID-19 Pandemic Management. *Sustainability* **2021**, *13*, 3797. [[CrossRef](#)]
8. Vaščák, J.; Pomšár, L.; Papcun, P.; Kajáti, E.; Zolotová, I. Means of IoT and Fuzzy Cognitive Maps in Reactive Navigation of Ubiquitous Robots. *Electronics* **2021**, *10*, 809. [[CrossRef](#)]
9. Li, Z.; Liu, H.; Zhang, Z.; Liu, T.; Xiong, N.N. Learning knowledge graph embedding with heterogeneous relation attention networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *5*, 1–12. [[CrossRef](#)]
10. Liu, H.; Zheng, C.; Li, D.; Shen, X.; Lin, K.; Wang, J.; Zhang, Z.; Zhang, Z.; Xiong, N.N. EDMF: Efficient deep matrix factorization with review feature learning for industrial recommender system. *IEEE Trans. Ind. Inform.* **2021**, *18*, 4361–4371. [[CrossRef](#)]
11. Liu, H.; Nie, H.; Zhang, Z.; Li, Y.F. Anisotropic angle distribution learning for head pose estimation and attention understanding in human-computer interaction. *Neurocomputing* **2021**, *433*, 310–322. [[CrossRef](#)]
12. Martini, B.G.; Helfer, G.A.; Barbosa, J.L.V.; Espinosa Modolo, R.C.; da Silva, M.R.; de Figueiredo, R.M.; Mendes, A.S.; Silva, L.A.; Leithardt, V.R.Q. IndoorPlant: A Model for Intelligent Services in Indoor Agriculture Based on Context Histories. *Sensors* **2021**, *21*, 1631. [[CrossRef](#)]
13. Zave, P. Classification of Research Efforts in Requirements Engineering. *ACM Comput. Surv.* **1997**, *29*, 315–321. [[CrossRef](#)]
14. Gupta, V.; Fernandez-Crehuet, J.M.; Hanne, T.; Telesko, R. Requirements Engineering in Software Startups: A Systematic Mapping Study. *Appl. Sci.* **2020**, *10*, 6125. [[CrossRef](#)]
15. Graessler, I.; Oleff, C.; Scholle, P. Method for Systematic Assessment of Requirement Change Risk in Industrial Practice. *Appl. Sci.* **2020**, *10*, 8697. [[CrossRef](#)]

16. Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. Systematic mapping studies in software engineering. In Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12, Bari, Italy, 26–27 June 2008; pp. 1–10.
17. Hull, E.; Jackson, K.; Dick, J. *Requirements Engineering in the Solution Domain*; Springer: Berlin/Heidelberg, Germany, 2005.
18. Nuseibeh, B.; Easterbrook, S. Requirements engineering: A roadmap. In Proceedings of the Conference on the Future of Software Engineering, Limerick, Ireland, 4–11 June 2000; pp. 35–46.
19. Maiden, N.A.; Rugg, G. ACRE: Selecting methods for requirements acquisition. *Softw. Eng. J.* **1996**, *11*, 183–192. [[CrossRef](#)]
20. Yu, E. Modelling Strategic Relationships for Process Reengineering. Ph.D. Thesis, Computer Science Department, University of Toronto, Toronto, ON, Canada, 1995.
21. Aguilar, J.A.; Garrigós, I.; Mazón, J.N.; Trujillo, J. An MDA Approach for Goal-oriented Requirement Analysis in Web Engineering. *J. Univers. Comput. Sci.* **2010**, *16*, 2475–2494.
22. Bass, L.; Bergey, J.; Clements, P.; Merson, P.; Ozkaya, I.; Sangwan, R. *A Comparison of Requirements Specification Methods from a Software Architecture Perspective*; Technical Report; Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst: Philadelphia, PA, USA, 2006.
23. Chrissis, M.B.; Konrad, M.; Shrum, S. *CMMI for Development: Guidelines for Process Integration and Product Improvement*; Pearson Education: Upper Saddle River, NJ, USA, 2011.
24. Kitchenham, B.; Pretorius, R.; Budgen, D.; Brereton, O.P.; Turner, M.; Niazi, M.; Linkman, S. Systematic literature reviews in software engineering—A tertiary study. *Inf. Softw. Technol.* **2010**, *52*, 792–805. [[CrossRef](#)]
25. Petersen, K.; Vakkalanka, S.; Kuzniarz, L. Guidelines for Conducting Systematic Mapping Studies in Software Engineering. *Inf. Softw. Technol.* **2015**, *64*, 1–18. [[CrossRef](#)]
26. Kitchenham, B.A.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; Technical Report EBSE 2007-001; Keele University and Durham University Joint Report: Keele, Newcastle, UK, 2007.
27. Lv, Z.; Han, Y.; Singh, A.K.; Manogaran, G.; Lv, H. Trustworthiness in industrial IoT systems based on artificial intelligence. *IEEE Trans. Ind. Inform.* **2020**, *17*, 1496–1504. [[CrossRef](#)]
28. Arakaki, R.; Hayashi, V.T.; Ruggiero, W.V. Available and Fault Tolerant IoT System: Applying Quality Engineering Method. In Proceedings of the 2020 IEEE International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Istanbul, Turkey, 12–13 June 2020; pp. 1–6.
29. Dybå, T.; Dingsøyr, T. Strength of evidence in systematic reviews in software engineering. In Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Helsinki, Finland, 19–22 September 2008; pp. 178–187.
30. Silva, D.V.d.; Gonçalves, T.G.; Travassos, G.H. A Technology to Support the Building of Requirements Documents for IoT Software Systems. In Proceedings of the 19th Brazilian Symposium on Software Quality, SBQS'20, New York, NY, USA, 1–4 December 2020; Association for Computing Machinery: New York, NY, USA, 2020. [[CrossRef](#)]
31. Kaleem, S.; Ahmed, S.; Ullah, F.; Babar, M.; Sheeraz, N.; Hadi, F. An Improved RE Framework for IoT-Oriented Smart Applications Using Inetgrated Approach. In Proceedings of the 2019 International Conference on Advances in the Emerging Computing Technologies (AECT), Medina, Saudi Arabia, 10 February 2020; pp. 1–6. [[CrossRef](#)]
32. Silva, D.; Gonçalves, T.G.; da Rocha, A.R.C. A Requirements Engineering Process for IoT Systems. In Proceedings of the XVIII Brazilian Symposium on Software Quality, SBQS'19, Fortaleza, Brazil, 28 October–1 November 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 204–209. [[CrossRef](#)]
33. Fortino, G.; Russo, W.; Savaglio, C.; Shen, W.; Zhou, M. Agent-Oriented Cooperative Smart Objects: From IoT System Design to Implementation. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 1939–1956. [[CrossRef](#)]
34. Pérez Fernández, A.; Sindre, G. Mitigating the Impact on Users' Privacy Caused by over Specifications in the Design of IoT Applications. *Sensors* **2019**, *19*, 4318. [[CrossRef](#)]
35. Ahmad, A.; Fahmideh, M.; Altamimi, A.B.; Katib, I.A.; Albeshr, A.; Alreshidi, A.; Alanazi, A.; Mehmood, R. Software Engineering for IoT-Driven Data Analytics Applications. *IEEE Access* **2021**, *9*, 48197–48217. [[CrossRef](#)]
36. Mezghani, E.; Exposito, E.; Drira, K. A Model-Driven Methodology for the Design of Autonomic and Cognitive IoT-Based Systems: Application to Healthcare. *IEEE Trans. Emerg. Top. Comput. Intell.* **2017**, *1*, 224–234. [[CrossRef](#)]
37. Ruiz-López, T.; Noguera, M.; Rodríguez, M.J.; Garrido, J.L.; Chung, L. REUBI: A Requirements Engineering method for ubiquitous systems. *Sci. Comput. Program.* **2013**, *78*, 1895–1911. [[CrossRef](#)]
38. Costa, B.; Pires, P.F.; Delicato, F.C. Modeling IoT Applications with SysML4IoT. In Proceedings of the 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Limassol, Cyprus, 31 August–2 September 2016; pp. 157–164. [[CrossRef](#)]
39. Costa, B.; Pires, P.F.; Delicato, F.C. Specifying Functional Requirements and QoS Parameters for IoT Systems. In Proceedings of the 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Orlando, FL, USA, 6–10 November 2017; pp. 407–414. [[CrossRef](#)]
40. Meacham, S.; Halp, K. Requirements Engineering Methods for an Internet of Things Application: Fall-Detection for Ambient Assisted Living; In Proceedings of the BCS Quality Specialist Group Annual International Software Quality Management SQM/INSPIRE Conference, Bournemouth University, Poole, UK, 21–22 March 2016.
41. Ferraris, D.; Fernandez-Gago, C. TrUStAPIS: A trust requirements elicitation method for IoT. *Int. J. Inf. Secur.* **2020**, *19*, 111–127. [[CrossRef](#)]

42. Reggio, G. A UML-Based Proposal for IoT System Requirements Specification. In Proceedings of the 2018 IEEE/ACM 10th International Workshop on Modelling in Software Engineering (MiSE), Gothenburg, Sweden, 27–28 May 2018; pp. 9–16.
43. Sosa-Reyna, C.M.; Tello-Leal, E.; Lara-Alabazares, D. Methodology for the model-driven development of service oriented IoT applications. *J. Syst. Archit.* **2018**, *90*, 15–22. [[CrossRef](#)]
44. Paldês, R.A.; Canedo, E.D.; Guimarães, F.d.A.; 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'20, São Luis, Brazil, 1–4 December 2020; Association for Computing Machinery: New York, NY, USA, 2020; [[CrossRef](#)]
45. Lim, T.Y.; Chua, F.F.; Tajuddin, B.B. Elicitation Techniques for Internet of Things Applications Requirements: A Systematic Review. In Proceedings of the 2018 VII International Conference on Network, Communication and Computing, ICNCC 2018, Taipei City, Taiwan, 14–16 December 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 182–188. [[CrossRef](#)]
46. Bouanaka, C.; Benlahrache, N.; Benhamaid, S.; Bouhamed, E. A Review of IoT Systems Engineering: Application to the Smart traffic lights system. In Proceedings of the 2020 International Conference on Advanced Aspects of Software Engineering (ICAASE), Constantine, Algeria, 28–30 November 2020; pp. 1–8. [[CrossRef](#)]
47. Rehman, S.U.; Gruhn, V. An Effective Security Requirements Engineering Framework for Cyber-Physical Systems. *Technologies* **2018**, *6*, 65. [[CrossRef](#)]
48. Rafique, W.; Zhao, X.; Yu, S.; Yaqoob, I.; Imran, M.; Dou, W. An Application Development Framework for Internet-of-Things Service Orchestration. *IEEE Internet Things J.* **2020**, *7*, 4543–4556. [[CrossRef](#)]
49. Samaila, M.G.; Sequeiros, J.B.F.; Simões, T.; Freire, M.M.; Inácio, P.R.M. IoT-HarPSecA: A Framework and Roadmap for Secure Design and Development of Devices and Applications in the IoT Space. *IEEE Access* **2020**, *8*, 16462–16494. [[CrossRef](#)]
50. de Farias, C.M.; Brito, I.C.; Pirmez, L.; Delicato, F.C.; Pires, P.F.; Rodrigues, T.C.; dos Santos, I.L.; Carmo, L.F.; Batista, T. COMFIT: A development environment for the Internet of Things. *Future Gener. Comput. Syst.* **2017**, *75*, 128–144. [[CrossRef](#)]
51. Laplante, N.L.; Laplante, P.A.; Voas, J.M. Stakeholder Identification and Use Case Representation for Internet-of-Things Applications in Healthcare. *IEEE Syst. J.* **2018**, *12*, 1589–1597. [[CrossRef](#)] [[PubMed](#)]
52. Dhoubib, S.; Cuccuru, A.; Fèvre, F.L.; Li, S.; Maggi, B.; Paez, I.; Rademacher, A.; Rapin, N.; Tatibouet, J.; Tessier, P.; et al. Papyrus for IoT—A Modeling Solution for IoT. In Proceedings of the l'Internet des Objets: Interaction Homme-Machine et Facteurs Humains, Paris, France, 29 November 2016.
53. Zambonelli, F. Towards a General Software Engineering Methodology for the Internet of Things. *arXiv* **2016**, arXiv:1601.05569.
54. Taivalaari, A.; Mikkonen, T. A roadmap to the programmable world: Software challenges in the IoT era. *IEEE Softw.* **2017**, *34*, 72–80. [[CrossRef](#)]
55. Fitzgerald, B.; Stol, K.J. Continuous software engineering: A roadmap and agenda. *J. Syst. Softw.* **2017**, *123*, 176–189. [[CrossRef](#)]