TEl-IoT: A Template for Eliciting IoT Software System Requirements

Sabrina R. de Souza
Federal University of Amazonas (ICOMP/UFAM)
Manaus, Brazil
sabrina.rocha@icomp.ufam.edu.br

Bruno P. de Souza
Federal University of Rio de Janeiro (COPPE/UFRJ)
Rio de Janeiro, Brazil
bpsouza@cos.ufrj.br

Anderson Uchôa
Federal University of Ceará (UFC)
Itapajé, Brazil
andersonuchoa@ufc.br

Daniella de O. Costa
Federal University of Amazonas (ICET/UFAM)
Itacoatiara, Brazil
daniellacosta@ufam.edu.br

ABSTRACT

Context: The Internet of Things (IoT) is a network of physical objects and system connected through mutual communication protocols. IoT systems have specific characteristics such as, self-configuration, dynamic changes, device and software heterogeneity. **Goal:** As IoT systems incorporate several components of the software, hardware, communication, and other features, building requirements documents to such systems become a challenge for Requirements Engineering (RE). Thus, this paper presents TEl-IoT, a template to aid developers during the requirements elicitation activities for IoT systems. **Method:** We conducted three evidence-based studies. We first performed a literature review aiming to identify artifacts that support requirements elicitation and specification for IoT systems. Second, based on the literature review, we proposed the initial version of the TEl-IoT. Finally, we performed two empirical studies to assess the TEl-IoT: (i) feasibility study with industry regarding the first version of TEl-IoT, and (ii) an observational study to understand how students apply the TEl-IoT in an IoT project. **Results:** Our results showed that TEl-IoT is viable, and its use reduces the time spent on requirements elicitation, in comparison with the ad-hoc way. In addition, our qualitative results also suggested that the use of TEl-IoT facilitates the requirements elicitation for IoT systems. **Conclusion:** We expect our template to guide requirements elicitation for IoT systems in practice. Our results showed that TEl-IoT can support developers and contribute to the body of knowledge about RE applicable in the IoT context.

CCS CONCEPTS

• **Software and its engineering**; • **Software creation and management**; • **Designing software**; • **Requirements analysis**;

KEYWORDS

Requirements Elicitation, Internet of Things, Empirical Validation

ACM Reference Format:
Sabrina R. de Souza, Bruno P. de Souza, Anderson Uchôa, and Daniella de O. Costa. 2022. TEl-IoT: A Template for Eliciting IoT Software System Requirements. In *XVIII Brazilian Symposium on Information Systems (SBSI'22), May 16–19, 2022, 2022, Curitiba, Brazil.* ACM, New York, NY, USA, 8 pages. https://doi.org/https://doi.org/10.1145/1234567890

SBSI'22, May 16–19, 2022, Curitiba, Brazil

1 INTRODUCTION

The Internet of Things (IoT) is a set of systems that have sensors, objects, and points that allow communication among devices. These points enable the collection of information, monitor processes, obey commands, carry out coordinated actions to meet the users' objective, manage "things" remotely and autonomously, and allow decision-making through the analysis of the collected data [2]. The purpose of IoT is to provide quality of life to end-users and benefit different areas, such as smart cities, smart homes, and others [21]. These applications have different characteristics, such as: self-configuration, large amounts of data [8], connectivity all the time [28], heterogeneity among the devices, context awareness, security and privacy of the end-actors [7] [17] [23] [25] [26].

In this context, developing and dealing with IoT software systems can be a challenging activity. The process of building an IoT software system requires a multidisciplinary approach as opposed to the monolithic structure used to build conventional systems (web application or mobile apps), which rules out the existence of physical objects interconnected by networks [13]. This multidisciplinary approach involves processes from Requirements Engineering (RE) to implementation. The RE is the process of defining software requirements that meet the needs of system users and stakeholders. In RE, a set of activities is performed: elicitation, analysis, specification, validation, and management of the requirements document [24]. This paper focuses on the requirements elicitation activity, which consists in identifying requirements that may initially be vague and confusing for software engineers. For this, analysts and software engineers work together with end-users, using techniques that facilitate communication among them [4].

Therefore, our motivation is to investigate and contribute to software engineering in the IoT context, considering the need to create new holistic approaches with a multidisciplinary vision to develop new solutions [13]. Thus, we present the TEl-IoT: a template for eliciting IoT software system requirements, which can initially be vague and confusing for software engineers. To build the TEl-IoT, we rely on an evidence-based methodology to construct our template. Specifically, we conducted a literature review to identify existing strategies and challenges regarding requirements elicitation for IoT software systems. Next, based on the literature review, we proposed the initial version of the TEl-IoT. Finally, we performed two empirical studies (feasibility, and observational) to assess the TEl-IoT. Our results point out a promising way to support IoT engineers plus some refinements to be applied and re-evaluated in future work. In summary, we expect TEl-IoT can help IoT requirements engineering in practice.

2 RELATED WORK

Paldês et al. [16] presented a systematic mapping to identify techniques used for eliciting functional requirements of IoT systems. As a result, the authors identified several techniques, methods, processes, and tools. The most used were interviews, brainstorming, and use case modeling, including the widespread use of models based on the Unified Modeling Language (UML) and scenarios. They also identified challenges related to the software and hardware technologies, user-centric design, structured documents to distinguish between functional requirements and hardware guidelines.

Silva [22] presented an approach called ScenarIoT to catalog information for supporting the description of scenarios for the development of IoT systems. The authors performed a structured literature review to obtain concepts and characteristics of IoT. Besides, they proposed a technique that support the specification of IoT requirements through scenario descriptions. Hence, the authors identified devices, non-functional requirements, and behaviors (such as identification, sensing, processing, and actuation), which allows the identification of requirements through interaction flows among IoT elements. Additionally, the authors have proposed nine IoT interaction arrangements (IIA). These IIAs are the composition of the Scenar-IoT technique. Each IIA has a specific catalog that guides the software engineers to describe the IoT scenarios [22].

Aziz et al. [1] introduced a technique that adapts the conventional concept of Use Case for the IoT systems context. This technique aid the elicitation and specification of requirement in a systematic manner during the construction of systems for intelligent spaces. This technique defines a sequence of steps that must be monitored by system designers and consider the feedback from the end-users. The elicited requirements are documented in a standard format and can be reused in other development projects. The technique was validated during the development of an intelligent system. Reggio [19] proposed the IoTReq, a method that combines service-oriented UML modeling and ES practices to aid the elicitation and specification of functional requirements. The method was validated during a case study, in the construction of a real IoT system. The method enabled the definition and separation of functional requirements from those related to the technologies used.

In comparison with the prior studies, our main contribution is in the monitoring of requirements elicitation activities for IoT software systems. Our differential is the presentation of an empirically-derived template to support the requirements elicitation that considers the characteristics of IoT software systems.

3 LITERATURE REVIEW

In order to obtain a more accurate characterization of IoT software systems, we conducted a Literature Review (LR) together with the snowballing technique, which contributed to the coverage of the final set of selected papers. Our Literature Review adapted the guidelines proposed in [11]. Snowballing approach is a technique for searching studies of literature, in which its application occurs through a list of references of works or citations [29]. The LR aimed to identify techniques, tools, templates, and guides that support the software engineers during the requirements elicitation process.

LR Planning: consists of preparing the research protocol, including the definition of research questions (RQs), search string, control papers, search engines, among others. Besides, this phase also presents the definition of the search strategy, using the PICO approach [15]. PICO is divided into four levels of filtering: P (Population), I (Intervention), C (Comparison), O (Outcome). Due to the purpose of the study, we did not apply any comparison. To assess the quality and scope of the search string, we performed an exploratory research in which three control papers were defined, which are marked in Table 1 with the symbol (*). We also defined the inclusion (IC) and exclusion criteria (EC) to select works. The selected papers must answer, at least, one research question. Table 2 summarizes the literature review protocol.

LR Execution: the literature review was performed in November 2019 by three researchers. The search strings were executed again in February 2020 to obtain a greater number of publications. Thus, we selected the papers according to the IC. The search was conducted in the following search engines: Scopus, ACM, Science Direct, and Google Scholar. The search resulted in 71 papers after removing duplicates, posters, and proceedings. Then, a researcher applied the IC in the papers' abstract, resulting in 43 selected papers. After, the selection based on full reading was applied, in which 32 papers were excluded. As a result, 11 papers were included after this phase to compose our final set. Finally, we applied the snowballing technique (backward), in which three papers were added, resulting in 14 papers for the final set. Table 1 lists the selected papers.

Table 1: Selected Papers

ID	Paper Title	Search Engine	Year
S1	Data Visualization in Internet of Things: Tools, Methodologies, and Challenges	ACM	2020
S2	An Exploration to Determine Essential Requirements for Smart Home Application	SCOPUS	2019
S3*	TrUStAPIS: a trust requirements elicitation method for IoT	SCOPUS	2019
S4*	A Requirements Engineering Process for IoT Systems	ACM	2019
S5	A Systematic Mapping study on Internet of Things challenges	ACM	2019
S6*	A UML-based Proposal for IoT System Requirements Specification	ACM	2018
S7	Internet of things security: challenges and perspectives	ACM	2018
S8	Opportunistic Interaction in The Challenged Internet of Things	ACM	2017
S9	A Toolkit for Construction of Authorization Service Infrastructure for the Internet of Things	ACM	2017
S10	Internet of Things (IoT): A Survey on Architecture, Enabling Technologies, Applications and Challenges	ACM	2017
S11	Requirement Engineering Technique for Smart Spaces	ACM	2016
S12	Augmenting Requirements Gathering for People with Special Needs using IoT: A Position Paper	ACM	2016
S13	A Novel Approach for Specifying Functional and Non Functional Requirements using RDS (Requirement Description Schema)	Science Direct	2016
S14	Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges	Google Academic	2016

LR Results: We describe the results by RQ as follows.

RQ_1: What are the strategies used to elicit requirements for IoT software systems? As strategies, we mean any approaches, methods and techniques, guides, or templates provided in the literature. In summary, we identified the following strategies: **IoTReq (S6)** [19]: is a method that combines the service-oriented paradigm and Software engineering practices for eliciting and specifying functional requirements for IoT systems. The method presents preliminary evidence of non-functional requirements as well. To validate the IotReq, the authors performed a case study, in which their results showed readability about the specification. Besides,

Table 2: Literature Review Planning

Research Questions	**Search String**
RQ_1: What are the strategies used to elicit requirements for IoT software systems?	Population - IoT Software System | (IoT OR "Internet of Things")
RQ_2: What are the challenges faceted when building IoT software systems?	Intervention - Requirements Elicitation | ("requirements elicitation" OR "requirements engineering")
	Comparation | None
	Output - Techniques, tools, methods or guides to support requirements elicitations | (Framework* OR Template OR Tool* OR Guid* OR Catalog* OR Technique)
Inclusion Criteria	**Exclusion Criteria**
IC1: The study must be in the context of software engineering;	EC1: Papers being ideas or posters;
IC2: The study must be in the context of an IoT-based system;	EC2: Duplicate papers;
IC3: The study must report an evidence-based study (primary studies);	EC3: Self plagiarism;
IC4: The study must be written in English;	EC4: Studies that were not published between the years 2015 to August 2020.
IC5: The study must provide data to answer at least one of the research questions in the review;	
Technical Report Protocol	**Full LR report access is available at:** https://anderson-uchoa.github.io/SBSI2022/

IotReq allowed the professionals to specify requirements more accurate. **Requirement Description Schema (RSD) (S13)** [20]: is an XML-based approach, as XML defines specific vocabularies to serve different domains. To validate the approach, a case study was performed by comparing the use of RDS and approaches from other languages. The results revealed that RSD meets functional, non-functional requirements such as security and privacy. In addition, it emphasizes requirements metadata.

Questionnaire (S2) [9]: was applied to elicit requirements for smart home applications. A questionnaire was used for a group of people who use IoT technologies. The authors did not mention the advantages or disadvantages of using the questionnaire as a requirements elicitation technique. However, the authors were able to conclude that time, usability, security and mobility requirements can be elicited using one of the traditional elicitation techniques to IoT systems. **TrUStAPIS** proposals and **JSON Model(S3)** [6]: presents a method for eliciting requirements for an IoT scenario. This method must create domains and divide them into seven different types (usability, identity, security, availability, privacy, protection, and reliability). For the documentation, the authors proposed using JSON as a template. In addition, UML diagrams are adopted to assist in the requirements elicitation and specification phases in IoT. **RETIoT: A Requirements Engineering Technology for the IoT Systems (S4)** [21]: is an adapted and harmonized process from ISO IEC and IEE 12207: 2017 for IoT systems. This approach focuses on three subjects Business analysis; Stakeholder requirements; and System Requirements Definition.

The approach complements traditional methods and IoT (S12) [5] to support requirements elicitation for people with special needs. But, the amount of data generated is a challenge in terms of the security and privacy of users. The work of Aziz et al. [1] (S11) proposed the **technique that adapts the conventional use of Use Cases in the context of IoT-based systems** to identify stakeholder needs in an intelligent spatial system. For this, one must: (i) identify possible actions of actors with use cases; (ii) describe the functionality in each use case; (iii) identify the actors that perform the functionality actions; (iv) develop use case models that represent actors, use cases and their interactions; and (v) provide a description of use cases.

RQ_2: What are the challenges faced when building IoT software systems? Some concerns of IoT systems are related to non-functional requirements such as security, privacy, and reliability of the collected data. In this context, opportunities and challenges guided the template construction. The identified challenges were: **Environment not controlled (S7, S9)** [8] [10]: related to "things" that navigate to environments not reliable and unsupervised; **Evaluation (S1)** [18]: engineers evaluate the user interface based on the user perception and analytical reasoning; **Interoperability (S14)** [17]: on how to use and process the collected, exchanged or changed information; **Connectivity (S5, S8)** [12] [28]: related to maintaining the connection between physical objects and users via communication networks; **Creating insights (S1)** [18]: on the creation of meaningful results, that will leverage human capacity and facilitate data exploration, analytical reasoning, and better insights.

Cost versus usability (S14) [17]: The connection of physical objects to the internet generates costs with support resources, such as sensors and control mechanisms; **Scalability (S1, S7, S9, S10)** [18] [8] [10]: the ability to manage devices is critical, therefore, it is necessary to have the ability to monitor the health of multiple devices; **Data Management (S14)** [17]: it becomes a challenge for generating data at high scale. In this way, the handling of this data becomes critical. **Heterogeneity (S7, S9)** [8] [10]: IoT interconnects heterogeneous devices to provide modern systems to improve the user's quality of life. **Infrastructure (S1)** [18]: It is not always possible to reuse data, as there are different types. The challenge is to develop a common infrastructure, as the existing ones are specific to a particular domain.

4 TEL-IOT AT A GLANCE

The elaboration of the TEl-IoT template[1] was based on the body of knowledge obtained from the LR results. The TEl-IoT aims to support software engineers during the requirements elicitation, by performing a sequence of steps to make explicit and obtain the maximum information related to the system context, to provide the most correct and complete understanding of what is required for the system. Each step has a goal and result that fulfill the purpose of the RE process for IoT systems. We overview the TEl-IoT as follows.

Definition of the organization's context. Understanding the application domain helps to identify requirements [31]. Thus, to organize the context, the developers need to specify the: project name, responsible for creating the document and creation date, system domain (e.g., smart city, smart home, industry, among others), actors (e.g., physical and eventually human devices), type of data to be collected (e.g., physical dimensions of the system: volume, weight, temperature, unit, among others), as shown in the Figure 1.

Definition of the system scope. This step starts the requirements elicitation, in which problem, impact, and a possible solution must be described. Stakeholders and constraints related to the system must also be defined (e.g., tasks to be performed, monitoring,

[1] All artifacts are available at https://anderson-uchoa.github.io/SBSI2022/

Basic System Information	
Project name:	[project name]
Responsible:	[name of the person responsible for the project]
Creation date:	[yyyy/mm/dd]
System domain:	[describe the system domain, e.g.: smart city, smart home, industry, leisure, health, among others]
Type	[IoT system, cyber-physical, ubiquitous]
Actors:	[describe the actors of the IoT system as users, things, software systems, among others]
Type of data collected:	[describe the types of data collected, such as: temperature, weather, humidity, pollution, among others]

Figure 1: Definition of the Organization's Context

controlling, and initiating functions). As well as activities of the actors (e.g., create/save/change/delete/read), and communicate if external information is needed or if there are modules in the system. Figure 2 shows the TEl-IoT fields for the system scope definition.

System Scope	
Problem	[problem description]
Who affects	[Stakeholders affected by the problem]
Impact	[what is the impact of the problem?]
Possible solution	[list some of the main benefits of a good solution]
Scope Limitations	
What are the main tasks that the system must perform?	
Does the system need to perform any monitoring functions?	
Does the system need to perform any control functions?	
Does the system need to perform some boot function?	
Will an actor need to create, save, change, delete or read data on the system?	
Does an actor need to inform the system of external information?	
Does an actor need to be informed about certain occurrences in the system?	
Are there modules in the system?	

Figure 2: System Scope Definition

Definition of stakeholders and user environment. The characteristics of stakeholders and their needs must be described, classified by degrees of priority (high, medium, and low), as well as a solution proposal for each need. This step was developed considering the challenges obtained in the LR, as shown in Figure 3.

Definition of system requirements. IoT systems are divided into hardware and software requirements. Electronic and mechanical engineers are commonly involved in hardware decisions, and software engineers are involved in building the software. Therefore, TEl-IoT allows elicitation of requirements for hardware and software by defining the degree of priority. These can be classified into requirements for IoT systems or requirements for classic systems (e.g., web and mobile systems). TEl-IoT also allows the description of the documentation requirements (i.e., online guide, installation guide, configuration, or "README" file). Figure 4 shows the fields for the definition of system requirements.

Definition of system restrictions: In this step, the limitations concerning the execution of activities for the development of the project should be described (e.g., deadlines, scope, and costs). The deadlines refer to the delivery schedule, i.e., how long it will take for the final project to be delivered. The scope is a set of well-defined goals, deliverables, and tasks. The TEl-IoT serves to detail features and restrictions of the project, thus, it can serve as a contract between stakeholders and therefore includes the cost estimate. This estimate is the values of hiring, software acquisition, hardware equipment that must be defined before the project is approved.

Stakeholders	Necessity	Concerns	Solution Proposal	Priority
[Stakeholder name]	[What's the problem?]	[What are the causes of this problem?]	[What solutions does the stakeholder or user want?]	[Degree of development and influence]

User Environment	
Description of the environment	[The target user's desktop]
Number of users	[How many people are involved in completing the task? Is it being changed?]
Simultaneous accesses	[How many users can access the system at the same time?]
Duty cycle	[How long does a task loop take? How much time do users spend on each activity? Is it being changed?]
Reliability	[What is the probability that an item, component, equipment, machine, or system will perform its specified function in the project?]
Environmental restrictions	[What unique environment constraints affect the project? For example, do users require remote devices, work offsite, or work while traveling?]
Interface with other systems	[What system platforms are currently in use? Are there future platforms planned?]
Need of interested parties	[What other apps are in use? Does your app need to integrate with them?]
Power consumption characteristics	[What is the form of powering the product (e.g., batteries and external source)]
Identify physical and mechanical characteristics	[What are the physical and mechanical characteristics (size of the product, physical designer, mechanical functions of buttons and size of sensors and actuators for communication with the user)]
Interface	[Is there external communication? How is it performed? E.g.: USB, CAN, Serial]
Indicate critical situations	[May occasionally occur, e.g.: falls, shocks and crashes]
Connectivity	[Example: PAN (low scale), MAN (medium scale), WAN (large scale), among others.]

Figure 3: Definition of Stakeholders and User Environment

Hardware requirements
To define: • Sensors • Actuators • User interaction • HW interruptions • Buttons • Memoirs • External communication ports • Component requirements

Software requirements	Type
To define: • Variables • Determine SW functions • Delimitation of exceptions • Interrupt functions • Language requirements • Communication interface (software) • Monitoring functions • Data storage functions	[IoT or Classic]

Documentation Requirements	
User manual	[Not required - the system should be easy enough to use that there is no need for a user manual.]
Online Help	[General and context-specific help will be available for all functions contained in the system.]
Installation and Setup Guides, and "Readme" file	[A manual with installation instructions and configuration guidelines will be given to the company manager and the "Readme" file will be included as a standard component.]

Figure 4: System Requirements Definition

5 STUDY 1: FEASIBILITY STUDY

We followed the Goal Question Metric template [30] to define our evaluation goal: **analyze** the TEl-IoT template; **with the purpose of** characterizing its application feasibility; **from the point of view of** developers; **in the context of** a real medium-sized IoT system. Thus, we defined our research question (RQ) as follows: **RQ:** *The use of TEl-IoT assists software engineers during requirements elicitation for IoT systems?* Aiming to understand the degree of perception and the time spent to elicit requirements with TEl-IoT. In this study, we do not have defined any hypotheses, since we do not aim to compare the proposed template with another technology.

5.1 Study Phases and Artifacts

Phase 1: Preparation for Evaluation. We highlighted that we have conducted a pilot study with three students from the last year of the Software Engineering Course at ICET/UFAM to verify the quality of the instruments and procedures before applying them in the feasibility study. This phase consists in distribution of the *Consent Form* and the *Characterization Form*. The Characterization Form aimed to collect the experience level of each participant about four skills (see Table 3). We also trained the participants about concepts of software engineering in the context of the IoT (two hours spent) and the evaluation procedures (30 minutes spent). We spent 30 minutes answering questions without compromising the evaluation. **Phase 2: Run the Activities**. We first distributed the *Experiment Activity* to all participants, which discriminated against the one experiment activity: eliciting requirements to develop the IoT system using the TEl-IoT template. For this, the participants received the script for performing the tasks and the instruments defined to perform the study. Among them, two projects[2] within a real software context, with the goal of monitoring the production of materials for two "smart tracks" using sensors. After, the end of the elicitation activity, we also distributed the *Activity Experiment Form* composed of open and closed questions. The latter relies on five-point Likert scales [14] when possible to evaluate TEl-IoT. **Phase 3: Finish the Evaluation.** After the requirements elicitation with the TEl-IoT, the participants answered an *Follow-up Form* composed of open and closed questions, questions with Likert scales of five points [14] when possible (half an hour), aimed at collecting data on the participant's perception regarding the use of TEl-IoT[1].

5.2 Execution and Participant Characterization

The study was conducted remotely and asynchronously due to the COVID-19 pandemic, during the PROCAP-ICET project, which consists of a partnership between PROCAP, a company in the Electronics and Informatics sector, and the Institute of Exact Sciences and Technology - ICET/UFAM, Brazil. The objective of the PROCAP-ICET project is to promote the automation of the PROCAP production process, using technologies such as IoT. The feasibility study was performed in August 2020, divided into two sessions, one per day, conducted with members of the PROCAP-ICET project. We conducted two evaluation sessions due to the availability of our industrial partner. Each session took two hours to be completed. The first session aimed to address topics related to software engineering for IoT systems, and the second session aimed at the execution of requirements elicitation activity. As a feasibility evaluation, we recruited four participants (P1, P2, P3, P4). The set of participants was different from the pilot study. Table 3 overviews the participant's description according to the self-assessed experience in four skills, in which 1, 2, 3, and 4 indicate *No experience*; *Less than a year*; *I participated in projects only in the classroom*; and *I participated in industrial projects during one year or more*, respectively.

5.3 Evaluation Results

About the Time Spent. In summary, the P1, P2, P3, and P4 have spent 4 hours, an hour and a half, 1 hour, and 30 minutes, respectively, while using the TEl-IoT template. As mentioned, the study lasted four hours. Overall, we did not observe a significant difference in time spent between participants. Except for P1, which considered the training time. **Participants' perception of the support provided by TEl-IoT and the need for an additional artifact**. We asked the participants to measure the degree of satisfaction when using TEl-IoT: *How did TEl-IoT help you during the elicitation of IoT requirements?*. P2 considered himself *neutral* and would have met the same requirements if he had not used TEl-IoT. Participants P1, P3, and P4 consider *good*, perhaps they would not have identified some requirements if they had not used it.

[2]We omitted this form due to intellectual-property constraints.

Table 3: Characterization of Participants in the First Study

Participant	Software Engineering	Software Development	Requirements Elicitation	Development of IoT Systems
P1	3	3	3	Basic
P2	4	4	4	Advanced
P3	2	2	3	Intermediary
P4	1	3	3	Intermediary

About the perception of the degree of difficulty. We asked the participants to measure their degree of difficulty when using TEl-IoT: *How do you rate the degree of difficulty in applying TEl-IoT?*. This question was based on five-point Likert scales [14]. We observed that participant P1 and P4 considered *"very easy"*, and *"easy"*, respectively. While the participants P2 and P3 considered *"medium"*. We also asked participants to describe the difficulties faced in using TEl-IoT: *What are the difficulties faced in using TEl-IoT to support IoT requirements elicitation?*. We observed that the inexperience with software engineering concepts, when applied to IoT, makes it difficult to fill in all the fields of the proposed template. As mentioned by P1 and P4 as follows.

> *P1: "The difficulty is not related to the template, but only with **new information** that the template presents that were previously **unknown**."*

> *P4: "[...] I found it difficult to identify **many fields**, and to visualize the system as a whole."*

We also asked the participants to assess their perception of whether the use of TEl-IoT helps in eliciting IoT system requirements: *Do you think that using TEl-IoT facilitates the elicitation of IoT requirements?*. This question was also was based on five-point Likert scales. Participant P1 stated that the use of TEl-IoT *totally* facilitates the requirements elicitation for IoT systems, and the participants P2, P3, P4 considered that it facilitates *most of the time*.

About suggestions for improvements. We asked participants to indicate suggestions for improvements that make TEl-IoT easier to use: *Do you believe that TEl-IoT can be improved to make it easier to use? How?*. In summary, we noted that there is a need to improve the fields of TEl-IoT by making the terms more clearly. As mentioned by P2 and P4 as follows.

> *P2: "Yes, with **clearer topics** and **easy to understand**."*

> *P4: "Yes, the TEl-IoT is good, but a person who has not had any experience with IoT systems, would feel a little **difficulty** with **actor specifications** [...]."*

6 STUDY 2: OBSERVATIONAL STUDY

Although the first experimental study showed evidence of the feasibility of the template to elicit IoT requirements, the qualitative results also revealed difficulties in the use of TEl-IoT. Based on these results, we improved the proposed template. Here, we highlighted some difficulties. First, we observed that some participants did not understand what the 'system type' field meant. This field was not explained in the template. To solve this problem, we added a brief description of what should be filled in this field. We also noticed that the participants were in doubt on the 'actors' field. Here, we added some descriptions to make this field clear to the participants.

Aimed to assist the TEl-IoT template maturation process, in sequence, we conducted an observational study. We also followed the GQM template [30] to define our evaluation goal: **analyze** the improved version of TEl-IoT template; **for the purpose of** characterizing the easy of use and usefulness; **with respect to** eliciting IoT system requirements; **from the viewpoint of** novice software developers; **in the context of** a modeled medium-sized IoT system. We defined our research questions (RQs) as follows.

RQ_1: *How easy was it for the students to use the TEl-IoT template?* We aim to understand if the TEl-IoT template is easy-to-use in terms of perception on: the degree of understanding, the need for additional artifacts, and usefulness. In this case, the presence or not of certain fields of the template may hinder easy-to-use. These cases would be opportunities for refining our template by adapting it to the developers' needs.

RQ_2: *What are the positive and negative aspects of the TEl-IoT template for eliciting IoT system requirements?* Through **RQ_2** we aim to identify opportunities for improving the template to make developers more comfortable with using the TEl-IoT template. Additionally, we aim to understand to what extent the improvements incorporated in the TEl-IoT help the developers.

6.1 Study Phases and Artifacts

Phase 1: Prepare for the Evaluation. We distributed the *Consent Form* and the *Characterization Form*. The training was also conducted to level the participant knowledge on the main concepts of software engineering when applied in IoT (two hours were spent). We overview the evaluation procedures and answer questions without compromising the evaluation (an hour spent). **Phase 2: Run the Activities.** This phase consisted of performing the activity to assess the TEl-IoT. For this, we distributed to the participants the *Experimental Activity*: elicit requirements for IoT systems using the TEl-IoT template. Then, the participants received the activity script and the project modeled based on the work of Wanzeler et al. [27] to elicit requirements. The main idea consists of a home automation system divided into an alarm system, temperature system, and lighting system modules, each of which controls or monitors a residence, and should issue a notification to the user. We distributed the *Activity Experiment Form* composed of open and closed questions. The latter relies on five-point Likert scales [14] when possible. The form should be filled out right after completing the activity.

Phase 3: Finish the Evaluation. We left an hour for participants to answer the *Follow-up Form.* for collecting the data on participants' perceptions using the TEl-IoT. This form was distributed after the participants completed the elicitation activity[1]. **Phase 4: Collect and Analyze Data**. We conducted the experiment data analysis in two steps. In **Step 1** we have applied the *open coding* procedure for labeling the positive, negative and improvement suggestions mentioned by developers [3]. In **Step 2**, we performed the descriptive analysis [30] in the closed questions.

6.2 Execution and Participant Characterization

The observational study also was conducted remotely and asynchronously due to the COVID-19 pandemic. The study was performed in a short course "Software Engineering Practices" promoted by "III Ada Day 2020", an event aimed at collaboration with the scientific community. The short course served as training for participants, addressing topics related to SE in the context of IoT systems. The short course had a duration of four hours. We recruited 10 participants (P1 to P10). After, we grouped them into two groups with three members and another group with four members. This division served to facilitate the execution of the activity. However, the evaluation of the TEl-IoT occurred individually. We also counted with a researcher to moderate and monitor the requirements elicitation activity. Table 4 overviews the participant's description according to the self-assessed experience in three skills, in which 1, 2, 3, and 4 indicate *No experience*; *Less than a year*; *I participated in projects only in the classroom*; and *I participated in industrial projects during one year or more*, respectively.

Table 4: Participant Characterization in the Second Study

Group	Participant	Software Development	Requirements Elicitation	Development of IoT Systems
Group 1	P1	3	3	Basic
	P2	4	4	Advanced
	P3	2	1	Basic
Group 2	P4	4	2	Basic
	P5	3	3	Intermediary
	P6	3	3	Basic
	P7	3	3	Basic
Group 3	P8	2	2	Basic
	P9	3	3	Basic
	P10	3	3	Intermediary

6.3 Evaluation Results

About RQ_1. We asked the participants to assess their perception on the **ease of understanding** of the fields presented in the TEl-IoT template: *"The fields contained in the Tel-IoT template were easy to understand?"*. This question relies on five-point Likert scales. The participant's response, in which 40% of participants have considered that the fields of TEl-IoT have a *medium* difficulty to be understood. Additionally, we asked participants to report whether they felt the need for any **additional artifacts** during the use of TEl-IoT: *"Did you feel the need for any additional support in the use of TEl-IoT?"*. In summary, we observed that 40% of the participants felt "no" need for any additional artifacts. Another 30% answered "yes" felt the need for an artifact, 30% said "maybe".

In order to identify whether the TEl-IoT template is **useful** for eliciting IoT software system requirements, we asked the participants *"How did TEl-IoT help you elicit requirements?"*. For this question, 100% participants strongly agreed that the template was useful in eliciting the requirements of the IoT system and that they might not have identified some requirements if they hadn't used TEl-IoT.

About RQ_2. To identify **positive aspects** of TEl-IoT, we asked participants to justify their perception on whether TEl-IoT helped in eliciting requirements: *Do you consider that TEl-IoT helps in eliciting requirements? If "yes" or "no"?*. All participants said "yes". Down we present the justifications of some participants.

*P2: "Yes, because with the TEl-IoT, the identification of requirements, description of the environment, limitation of scope, was done **quickly** and with **simple recognition**."*

*P4: "Yes, because the way to describe the requirements are more **clear** and **objective**."*

*P5: "Yes, [...] it would **certainly** be a template that I would **look for** and **even indicate**, as it would also fit other types of systems."*

*P9: "Yes, I believe it **helps a lot**, since there is no existing template to assist in this process. [...] I also believe that its **application** is of **paramount importance** for researchers and developers in the area."*

Additionally, we also asked participants to justify their perception on which makes the use of TEl-IoT easy or difficult: "In your opinion, what makes the use of TEl-IoT easy or difficult?". We present some participants' responses as follows.

*P1: "It **helps in communicating** with the IoT **stakeholders**, facilitating the elicitation of requirements."*

*P2: "The way in which each section was described and defined, **facilitated the identification of requirements**, e.g., in each section small questions were asked, helping in the recognition and speed when filling in the fields."*

*P6: "TEl-IoT is **easy to use**, as it has a **clear** and **concise language**. It allows you to quickly understand what must be completed in each field."*

P9: "Sample texts make it a more easier"

To identify the **negative aspects** and improve the suggestions, we asked participants to cite points they considered negative: "List all the points you judge negative for the template". We present some negative aspects mentioned by participants as follows.

*P1: "In item 5. Connectivity and interface look similar in the template comments. It is **not understandable** the part of the stakeholders' needs and the interface with other systems [...]"*

*P2: "Need to **improve the suggestions** for the user"*

From these responses obtained in this study, suggestions for improvement emerged, such as: (1) reformulate questions and guidelines to avoid ambiguity; (2) simplify the language used; (3) define questions and guidelines to assist the software engineer during the requirement bidding. These qualitative data built a crucial foundation for the improvement of a new version of TEl-IoT.

7 TOWARDS A SECOND IMPROVEMENT VERSION OF THE TEL-IOT TEMPLATE

The obtained results of our feasibility and observational studies help us to obtain indicators of the evolution of the TEl-IoT[1]. To this end, we have identified that: the fields to define the scope and the project limitations must present guidelines to facilitate their fill. In this context, we have added guidelines to limit the project scope definition. Figure 2 shows the TEl-IoT in its first version, and Figure 5 illustrates its improved version. In addition, the guidelines contained in the definition of the user environment needed to be improved to facilitate understandability and avoid ambiguities. Table 5 shows the modified version of Figure 3.

According to our second study, the participants needed guidance in addition to the suggestions. To minimize this need, we added a description of what the hardware requirements are about. Figure 6 illustrates the evolution of this field.

What are the main tasks that the system must perform?	*[eg: monitor engine temperature]*
Will the system need to perform any monitoring functions? Which?	*[eg: monitoring a patient's condition]*
Will the system need to perform any control functions? Which?	*[in this field, control is related to the administration of people, products, among others. Eg: control the number of patients who have been admitted to the inpatient room]*
Will the system need to perform any initialization functions? Like?	*[eg: "click of a button.", "resistors and transistors need electrical current to function"]*
Will an actor need to create, save, change, delete or read data on the system? For what?	*[eg: The system should capture information from the temperature sensors and send this data to a display]*
Will an actor need to inform the system about external information? Which are?	*[eg: The system must inform the user of the temperature of the engine in order to carry out preventive maintenance of the machine]*
Does an actor need to be informed about certain occurrences in the system? Which are?	*[eg: The system should inform users of the machine's motor status (stopped / running)]*
Are there modules in the system? Which are?	*[eg: The residential system should be divided into temperature and alarm monitoring modules]*

Figure 5: User Environment Description

Table 5: Modified Version of the User Environment

Definition	Improvement suggestions
Connectivity	Describe the type of connectivity needed, for example: wireless, wired, among others. Also describe the size of the network coverage and the type, for example: PAN (low scale), MAN (medium scale), WAN (high scale), e.g.
Environmental Restrictions	What unique environment constraints affect the project? For example, do users require remote devices, work offsite, or work while traveling?
Power consumption characteristics	What is the form of feeding the product? (e.g. cells, batteries and external source).
Identify physical and mechanical characteristics	What are the physical characteristics? (dimension and location of components) What are the mechanical characteristics? (e.g. mechanical functions of buttons, size of sensors and actuators for user communication).
Interface	Is there external communication? Where will the data be presented? How is it performed? E.g.: USB, CAN, Serial, dashboard.

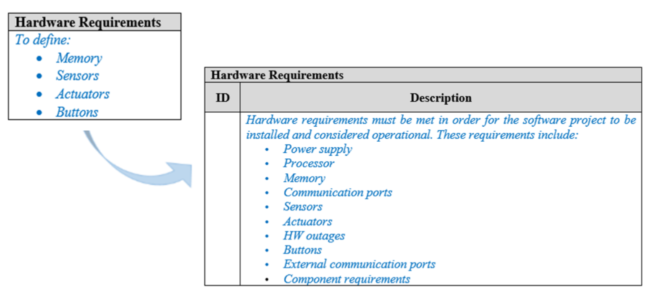

Figure 6: Improvement Made in Item 7 of the TEl-IoT

8 THREATS TO VALIDITY

Construct and Internal Validity. We designed our studies artifacts (e.g., study protocol and forms) before the experiment execution. Thus, we expected to avoid changing the study procedures as we analyze the experimental data. We wrote and validated the forms in pairs in various review rounds. We designed an experiment activity that involved the complete use of the TEl-IoT template. We have followed strict procedures for running the experiment with developers, including the training session. We collected the developer background before the experiment and we have grouped developers according to their level of experience. The developers conducted the experiment remotely and asynchronously due to the

COVID-19 pandemic. However, all developers were trained about the experimental procedures. Additionally, we have addressed the doubts of developers whenever possible.

Conclusion and External Validity. We tabulated and validated all the extracted data in a pair. Thus, we expected to avoid missing and incorrect data. Our analysis followed guidelines of descriptive data analysis [30], and open coding procedure [3]. All coding was performed by one researcher and validated by another researcher. We solved ties and treated divergences through discussions. We counted on the participation of 10 developers in our second experiment. Our participant set is limited but diversified. Additionally, two threats were identified: (i) the participation of students rather than software engineers; (ii) execution in an academic environment harms the reality of the industrial context.

9 CONCLUSION AND FUTURE WORK

This paper introduces the TEl-IoT template that aims to guide software engineers during requirements elicitation for IoT software systems. For building the TEl-IoT, we first conducted a literature review, aimed at identifying the main strategies used to elicit requirements for IoT software systems, and the challenges faced during the development of such systems. Second, based on the obtained results from the literature review, we derived and improved the TEl-IoT template, by conducting two studies: feasibility and observational. In the first study, we evaluated the template feasibility in terms of effort reduction in eliciting requirements with four developers of a real medium-sized IoT system. Additionally, we identified improvement opportunities. In the second study, based on the identified improvement opportunities, we improved the template and conducted an observational study, in which we evaluated the template in terms of the ability to understand, and positive and negative aspects with 10 developers.

Our results suggest that TEl-IoT is quite useful to support the elicitation of requirements for IoT systems. Additionally, the participants provided us with insights for future refinements that can drive a future template improvement and re-evaluation. As future work, we plan to improve our template based on industrial feedback. We also intend to expand the focus of the template to include other phases of RE (analysis, specification, validation, and management).

Acknowledgments. The authors would like to thank all participants in both experimental studies. In addition, we also thank CALCOMP (no. 728/02 – IIoT and Industry 4.0). This work was also partially funded by CNPq and CAPES - Financial Code 001.

REFERENCES

[1] Muhammad Waqar Aziz, Adil Amjad Sheikh, and Emad A Felemban. 2016. Requirement engineering technique for smart spaces. In *1st ICC*. 1–7.
[2] Mauro Conti, Ali Dehghantanha, Katrin Franke, and Steve Watson. 2018. Internet of Things security and forensics: Challenges and opportunities. (2018).
[3] Juliet Corbin and Anselm Strauss. 2008. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (3 ed.). SAGE Publications.
[4] Priscila Basto Fagundes, Gislaine Parra Freund, Luciane Paula Vital, Camila Monteiro de Barros, and Douglas Dyllon Jeronimo de Macedo. 2020. Taxonomias, ontologias e tesauros: possibilidades de contribuição para o processo de Engenharia de Requisitos. *Em Questão* 26, 1 (2020), 237–254.
[5] Mexhid Ferati, Arianit Kurti, Bahtijar Vogel, and Bujar Raufi. 2016. Augmenting requirements gathering for people with special needs using IoT: a position paper. In *9th CHASE*. 48–51.
[6] Davide Ferraris and Carmen Fernandez-Gago. 2020. TrUStAPIS: a trust requirements elicitation method for IoT. *Int. J. Inf. Secur. Priv.* 19, 1 (2020), 111–127.
[7] Arindam Giri, Subrata Dutta, Sarmistha Neogy, Keshav Dahal, and Zeeshan Pervez. 2017. Internet of Things (IoT) a survey on architecture, enabling technologies, applications and challenges. In *1st IML*. 1–12.
[8] Amina Harit, Abdellah Ezzati, and Rachid Elharti. 2017. Internet of things security: challenges and perspectives. In *2nd ICC*. 1–8.
[9] Fazilah Ismail, Sabrina Ahmad, and Ummi Rabaah Hashim. 2019. An Exploration to Determine Essential Requirements for Smart Home Application. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* 8 (2019).
[10] Hokeun Kim, Eunsuk Kang, Edward A Lee, and David Broman. 2017. A toolkit for construction of authorization service infrastructure for the internet of things. In *2nd IoTDI*. 147–158.
[11] Barbara A Kitchenham, Tore Dyba, and Magne Jorgensen. 2004. Evidence-based software engineering. In *26th ICSE*. IEEE, 273–281.
[12] Aleksandr Lepekhin, Alexandra Borremans, Igor Ilin, and Sami Jantunen. 2019. A systematic mapping study on internet of things challenges. In *1st SERP4IoT*. IEEE, 9–16.
[13] Rebeca C Motta, Káthia M de Oliveira, and Guilherme H Travassos. 2018. On challenges in engineering IoT software systems. In *XXXII SBES*. 42–51.
[14] Geoff Norman. 2010. Likert scales, levels of measurement and the "laws" of statistics. *Adv. Health Sci. Educ.* 15, 5 (2010), 625–632.
[15] Madhukar Pai, Michael McCulloch, Jennifer D Gorman, Nitika Pai, Wayne Enanoria, Gail Kennedy, Prathap Tharyan, and John M Colford Jr. 2004. Systematic reviews and meta-analyses: an illustrated, step-by-step guide. *Natl. Med. J. India* 17, 2 (2004), 86–95.
[16] Roberto Avila Paldês, Edna Dias Canedo, Fernando de Albuquerque Guimarães, and Angélica Toffano Seidel Calazans. 2020. Functional Requirements Elicitation in IoT Systems: a follow-up study. In *19th SBQS*. 1–10.
[17] Keyur K Patel, Sunil M Patel, and others. 2016. Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges. *Int. J. Eng. Sci.* 6, 5 (2016).
[18] Antonis Protopsaltis, Panagiotis Sarigiannidis, Dimitrios Margounakis, and Anastasios Lytos. 2020. Data visualization in internet of things: tools, methodologies, and challenges. In *15th ARES*. 1–11.
[19] Gianna Reggio. 2018. A UML-based proposal for IoT system requirements specification. In *10th MiSE@ICSE*. 9–16.
[20] Tejas Shah and SV Patel. 2016. A novel approach for specifying functional and non-functional requirements using rds (requirement description schema). *Procedia Comput. Sci.* 79 (2016), 852–860.
[21] Danyllo Silva, Taisa Guidini Gonçalves, and Ana Regina C da Rocha. 2019. A requirements engineering process for IoT systems. In *XVIII SBQS*. 204–209.
[22] Valéria Martins da Silva. 2019. ScenarIoT: support for scenario specification of internet of things-based software systems. (2019).
[23] Amanda Sousa, Anderson Uchôa, Eduardo Fernandes, Carla IM Bezerra, José Maria Monteiro, and Rossana MC Andrade. 2019. REM4DSPL: A requirements engineering method for dynamic software product lines. In *XVIII SBQS*. 129–138.
[24] Richard H Thayer, Sidney C Bailin, and M Dorfman. 1997. *Software requirements engineerings*. IEEE Computer Society Press.
[25] Anderson G Uchôa, Carla IM Bezerra, Ivan C Machado, José Maria Monteiro, and Rossana MC Andrade. 2017. ReMINDER: an approach to modeling non-functional properties in dynamic software product lines. In *16th ICSR*. Springer, 65–73.
[26] Anderson G Uchôa, Luan P Lima, Carla IM Bezerra, José Maria Monteiro, and Rossana MC Andrade. 2017. Dymmer-nfp: Modeling non-functional properties and multiple context adaptation scenarios in software product lines. In *16th ICSR*. Springer, 175–183.
[27] Tiago Wanzeler, H Fulber, and Bruno Merlin. 2016. Desenvolvimento de um sistema de automação residencial de baixo custo aliado ao conceito de Internet das Coisas (IoT). In *XXXIV SBrT*. 40–44.
[28] Hanno Wirtz, Jan Rüth, Martin Serror, Jó Ágila Bitsch Link, and Klaus Wehrle. 2014. Opportunistic interaction in the challenged internet of things. In *9th CHANTS@MobiCom*. 7–12.
[29] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *18th EASE*. 1–10.
[30] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.
[31] Didar Zowghi and Chad Coulin. 2005. Requirements elicitation: A survey of techniques, approaches, and tools. In *Engineering and managing software requirements*. Springer, 19–46.