

Requirements Validation Process in the Context of Internet of Things (IoT) and Health Care

Walter C. S. S. Simoes

UEA – Amazonas State University,
Superior School of Technology-EST,
Campus Manaus – Amazonas – Brazil
CESAR Institute – Campus Manaus – Amazonas - Brazil

Vandermi da Silva

Institute of Computing, ICOMP,
Federal University of Amazonas,
UFAM – Campus Itacoatiara – Amazonas – Brazil

Williame Rocha

UEA – Amazonas State University,
Superior School of Technology-EST,
Campus Manaus – Amazonas - Brazil

Andreza Mourao

UEA – Amazonas State University,
Superior School of Technology-EST,
Campus Manaus – Amazonas - Brazil

Abstract— The increasing popularity of IoT in Healthcare projects has enabled diverse solutions that allow an increase in the quality of life of those who use the devices and the health professionals who recommend them. However, its development has required knowledge about existing requirements validation processes and techniques both in the context of software engineering and in IoT. This work proposes the construction of a requirements engineering (RE) process for the specification of IoT systems, implemented to support software development teams in defining their processes and techniques. The adopted methodology is a survey of literature in a survey to investigate the practices adopted in the industry and proposals for Healthcare solutions. The results found reveal that the most used technique for specifying requirements are meetings with interested parties and brainstorming; and to validate the requirements used, prototypes and use cases are used.

I. INTRODUCTION

Contemporary software systems, such as those inherent in the Internet of Things (IoT), Industry 4.0, Smart Cities, and HealthCare systems, among others, are complex when compared to traditional software systems. This complexity comes from the inclusion of new concerns and characteristics related to the network, software, hardware, context sensitivity, interface, and interoperability [1][2].

IoT-based software systems seek to promote the interweaving and alignment of technologies and devices through data networks, allowing their access points with the scenario, such as the adopted sensors, to capture and exchange data, make decisions, and act, uniting the real and virtual worlds. Because of its specific technological characteristics, building IoT systems is not simple, requiring adapted and/or innovative software technologies to create and guarantee the quality of the built product [1].

The quality applied in the development of contemporary software systems depends on technologies that respond to new concerns and characteristics related to these systems. As with any other product built on engineering principles, a fundamental activity of IoT software system development is building the requirements document. The defects present in the

requirements document can directly affect the increase of time, cost and effort for the project; dissatisfied customers and end users; low software system reliability; high number of failures; among others [3][4].

Requirements engineering (ER) defines the lifecycle of the requirements document, which covers the stages of conception, elicitation, negotiation, analysis, specification, verification, validation and management, and ensures its proper construction in all steps [4][5]. The literature presents several software technologies that aim to support requirements engineering, but it is still an open area for discussion, since not all of these tools cover the different stages of RE, especially when it is necessary to work with the specificities of information systems. IoT software.

This paper brings as a problem for investigation and proposed solution: How to build the IoT software set functional requirements for HealthCare in a more adequate way?

This problem arises from discussions about the relationships that software technologies need to have when the objective is to build IoT solutions, which increase the importance that a requirements document can offer in terms of stability, adequacy and quality of the project.

The hypothesis presented in this work brings as a strategy the application of a combination of requirements engineering directed to IoT problems, called REIoT (Requirements Engineering for software systems based on IoT).

The term technology refers to the methodological, technical and tooling support offered by REIoT, whose objective is to support the construction of requirements documents for IoT software systems.

The REIoT technology is composed of a requirements specification technique based on the description of IoT scenarios - SCENARIOT [6], an IoT scenario inspection technique - SCENARIOTCHECK [7], a construction process – initial version [8] and templates of artifacts (project scope, proposed solution and description of IoT use cases) that support the process activities and make up the requirements document. The SCENARIOT and SCENARIOTCHECK

techniques were previously evaluated through experimental studies which showed their viability [6][9] and have been used in IoT software systems projects developed by the Software Engineering Group of the GAIT/UEA project.

This paper aims to present the REIoT technology and a proof of concept involving the comparison of its templates with those used to support the construction of requirements documents for conventional software systems. The results indicate that the REIoT templates allow capturing the information for IoT projects and that they are ready to be evaluated in the construction of requirements documents for IoT software systems. In addition, it is possible to observe that the REIoT technology provides coverage for the main phases of the RE regarding the specificities of IoT software systems.

The rest of the paper is organized as follows: In Section 2, an overview is presented on the construction practices of the requirements of Computing Systems aimed at IoT and HealthCare through a survey of papers. Section 3 presents a set of works related to requirements engineering practices and IoT. In Section 4, the process of gathering functional and non-functional requirements and business rules for building Healthcare systems is shown. Section 5 presents the criteria and metrics adopted to evaluate and validate the requirements adopted in the development of the systems. Section 6 discusses the topics worked on and used to establish a case study as a reference for this study.

II. EASE OF USE

Building a quality requirements document requires engineering. RE phases and activities may differ according to the application domain, people involved, processes, organizational culture, among others. However, it is possible to observe some recurring phases and activities in requirements engineering, such as: conception, elicitation, negotiation, analysis, specification, verification, validation and management.

Therefore, this section presents a set of related works found in the technical literature, which address technologies for the different stages of RE, mentioned above. This study was structured based on the GQM paradigm (Goal Question Metric) [27]. Table 1 presents the comparison of these works in chronological order).

Regarding the conception phase (CON), the GSEM-IoT [11], Fortino [12] and the technique by Laplante et al. [13] carry out an analysis of the interested parties involved in the system. Furthermore, Fortino and CORE [14] techniques provide mechanisms to perform business analysis. The feasibility analysis is partially handled by the IoT methodology technique [12].

Regarding the elicitation phase (ELI), Lim et al. [24] carried out a systematic review of the literature with the objective of listing the elicitation techniques for IoT systems. The techniques SCENARIOT [6], Fortino [12], IoT Methodology [12], Laplante et al. [13], CORE [14], Curumsing et al. [23], IoTReq [25] and TrUStAPIS [26] offer resources for gathering requirements. GSEM-IoT, IoTReq and IoT Methodology propose mechanisms to transform user needs into requirements.

For the negotiation phase (NEG), Lim et al. [24] presents a contribution to the analysis and resolution of conflicts. The Ignite technique and the technique by Kaleem et al. [17] address the impact analysis. Only the technique described by Lim deals with risk analysis [24].

In the analysis phase (ANA), Ignite, the technique by Laplante et al. [13], CORE [14], the technique by Curumsing et al. [23], IoTReq [25], the technique of Takeda et al. [18] and the technique by Touzani et al. [19] use UML diagrams to elaborate the analysis models. The technique by Silva et al. [15] and the IoT-RML technique [20] deal with artifacts and models that can be reused.

For the specification phase (ESP), IoTReq [25], CORE [14], TrUStAPIS [26], the technique of Takeda et al. [18] and IoT-RML [20] use formal (own) models for requirements specification. The technique by Silva et al. [15] and the technique by Mahalank et al. [16] provide templates for specifying requirements. The SCENARIOT technique [6] proposes the specification of IoT scenarios using interaction arrangements.

As for the verification phase (VER), SCENARIOTCHECK [7] and the technique by Naem et al. [22] propose mechanisms to verify requirements. The technique by Curumsing et al. [23], IoT-RML [20], the technique of Naem et al. [22] and Yamakami's technique [21] offer mechanisms to check conflicting requirements.

Regarding the validation phase (VAL), Fortino et al. [12], IoT Methodology [12] and Lim et al. [24] offer the prototyping technique to ensure that the product meets the needs of users.

Finally, regarding the management phase (GER), some studies offer mechanisms to enable traceability, such as the techniques by Curumsing et al. [23] and de Silva et al. [15]. The TrUStAPIS technique [26] provides traceability and requirements change management.

TABLE I. TECHNOLOGIES FOR REQUIREMENTS X PHASES OF THE RE

Technology / Phase*	C	E	N	A	S	V	L	G
Silva et al. [15]				x	x			x
Mahalank et al. [16]					x			
Kaleem et al. [17]			x					
Takeda et al. [18]				x	x			
Touzani et al. [19]				x				
IoT-RML [20]				x	x	x		
Yamakami [21]						x		
GSEM-IoT [11]	x	x						
Naem et al. [22]						x		
Curumsing et al. [23]		x		x		x		x
Fortino et al. [12]	x	x	x	x			x	
Laplante et al. [13]	x	x		x				
Lim et al. [24]		x	x				x	
IoTReq [25]		x		x	x			
CORE [14]	x	x		x	x			

Technology / Phase*	C	E	N	A	S	V	L	G
SCENARIOT [6]		x			x			
SCENARIOTCHECK [7]						x		
TrUStAPIS [26]		x			x			X
REIoT	x	x	x	x	x	x	x	x

* C (CON) – Conception; E (ELI) – Elicitation; N (NEG) – Negotiation; A (ANA) – Analysis; S (ESP) – Specification; V (VER) – Verification; L (VAL) – Validation; G (GER) – Management

The REIoT technology (presented in sections 3 and 4) permeates the different phases of the RE, offering methodological and technical support through a constructive process and its templates.

III. OVERVIEW OF REQUIREMENTS SURVEYS IN SOFTWARE ENGINEERING

This section presents the technological base of REIoT (Requirements Engineering for software systems based on IoT), built to support the RE of IoT software systems. This technology is inserted in the context of a systems engineering approach, which includes the major stages of development of contemporary software systems [1]. The technological base that composes it brings the SCENARIOT [6] and SCENARIOTCHECK [7] techniques that have an experimental indication of viability and have been used in IoT software systems projects under development by the GAIT/UEA Group.

A. SCENARIOT

SCENARIOT is an IoT scenario specification technique [6]. The technique adapts and develops conventional technologies of software scenarios to support the specification of scenarios in IoT software systems, considering characteristics (adaptability, connectivity, privacy, intelligence, interoperability, mobility, among others) and behaviors (identification, sensing and actuation) specific to the IoT paradigm [10].

The combination of the characteristics with the three behaviors allowed creating nine IoT interaction arrangements (AIIs). These arrangements guide software engineers to capture essential information about the software system that will be built, such as the data to be collected, identification of “things” in the IoT system, among others.

Furthermore, each AII has a catalog containing the information of all the features to be captured in the scenario description, as shown in Figure 1. The AIIs (isolated or combined) result in a scenario description artifact for IoT software systems.

The scenario specification artifact produced by SCENARIOT contains the following information:

- Identification of the main elements of the IoT software systems that will compose the solution, as well as the way these elements are organized;
- Description of the problem domain, such as health, smart cities, automotive engineering, among others;
- Description of the role of each actor within the scenario;

- Description of the interactions between the actors and the IoT system.

B. SCENARIOTCHECK

SCENARIOTCHECK is a scenario checking technique for IoT-based software systems [7]. This technique was created to support the verification of artifacts produced by the scenario specification technique - SCENARIOT [6]. The objective of SCENARIOTCHECK is to increase the quality of the specification of IoT scenarios performed with the SCENARIOT specification technique. It consists of a checklist, composed of two parts, which helps inspectors (with little or no knowledge of IoT software systems) to identify defects in the description of scenarios (Figure 1).



Fig. 1. AII example - collecting and displaying data and its catalog (based on [6])

Data producers are those responsible for collecting data, such as sensors, tag readers, wearable devices, etc. Data viewers are responsible for presenting data to users or other devices. The intermediary layer called Broker includes the entire infrastructure capable of receiving raw data and only forwarding or preparing it for display to users.

The first part of the checklist, based on the AIIs of the SCENARIOT technique, aims to verify general problems, capturing some characteristics such as problem domain, alternative and exception flow, interaction and identification between actors, system and devices. The second part of the checklist takes into account non-functional properties of IoT software systems. These properties are defined according to six facets: environment, things, behavior, connectivity, interactivity and intelligence [1]. Sousa presented the complete checklist (2019) in his research that sought to describe a scenario inspection technique to receive IoT approaches [9].

After specifying the IoT scenarios with the SCENARIOT technique, the inspectors apply the SCENARIOTCHECK technique to verify the description of the scenarios. Finally, after the discrimination meeting, the scenario specification document is corrected based on the defects found.

IV. REQUIREMENTS GATHERING PROCESS USING THE REIoT TECHNIQUE

The REIoT technology (Requirements Engineering for software systems based on IoT) is composed of a constructive process, and templates to support the construction of the requirements document (list of requirements, IoT use cases, IoT scenarios, IoT interaction arrangements including their respective catalogs) under RE principles.

A. REIoT Requirements Document Construction Process

The construction process of the REIoT requirements document is based on the main phases of the RE: conception, elicitation, analysis, specification, verification, validation and

management. However, REIoT adapts and includes new activities to meet the specificities of IoT software systems. An overview of the construction process is presented in Figure 2. In the next paragraphs, a summary of each phase is presented.

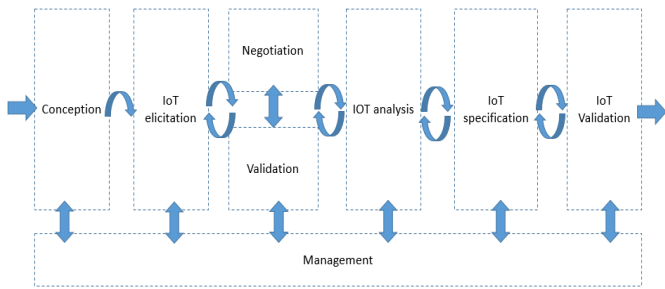


Fig. 2. Overview of the build process framework offered by REIoT.

This REIoT architecture model allows the definition of requirements to have an evolutionary character, as shown in Figure 3.

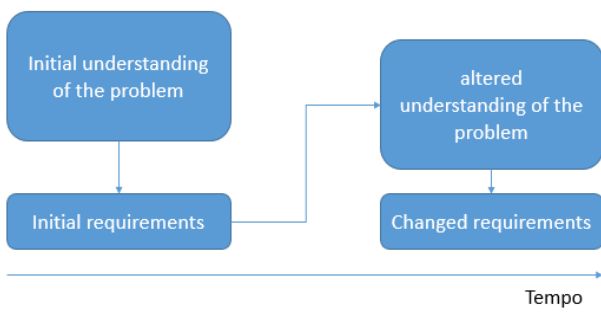


Fig. 3. Evolution of requirements gathering

During the design phase, one should seek to understand whether the problem or opportunity can be solved using IoT technologies. A feasibility study should also be carried out to determine if there are technologies available and qualified staff to solve the problem.

The IoT elicitation phase seeks to refine and transform business needs and stakeholders into requirements. During this phase, the SCENARIOT technique [6] is used to support the identification of requirements. In addition, IoT scenarios can be described to help understand the system’s behavior. Last, the final requirements list should organize and classify the requirements into “IoT requirements” and “non-IoT requirements”.

This phase is important to the process because a difficulty in eliciting and analyzing requirements is that stakeholders usually do not know what they want from a computer system, especially when it involves the development of hardware layers [24][26]. Still, on the elicitation definition process, stakeholders express requirements in their own terms and with implicit knowledge of their domain on the subject. That is, the customer’s domain can bring great affects to the requirements. To reduce the impacts of these stakeholder domain influences, the Requirements Elicitation and Analysis model follows a cyclical process, as shown in Figure 4.

The validation and negotiation phases occur in parallel. During the validation phase, requirements are validated,

attesting that a common understanding of the system has been achieved. At the same time, the negotiation phase seeks, together with the interested parties, to prioritize requirements and resolve conflicts, considering dependencies and precedence between requirements. The cost, effort and risk analysis for building the system must also be performed.

During the validation and negotiation phase, an immersion (ethnography) is applied, which applies observation technique to increase the understanding of operational processes and helps to extract and confirm the requirements about the system to be developed.

During the IoT analysis phase, system models are generated. Possible IoT use cases and system IoT interaction arrangements are listed. The IoT use-case diagram and catalog filling of identified interaction arrangements should be produced during this phase. The IoT specification phase encompasses the description of the identified IoT use cases. This phase runs parallel to the analysis phase and iterative flows between phases can occur.

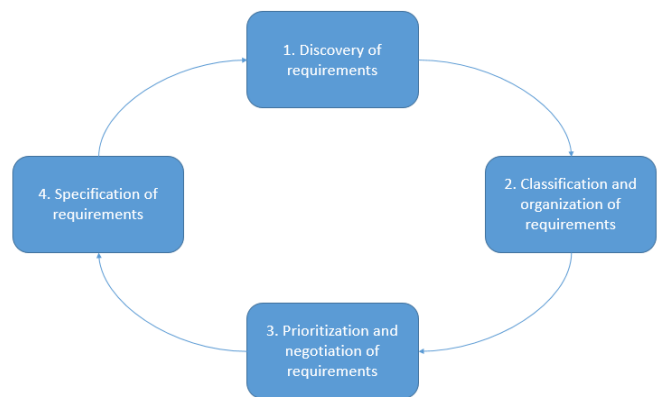


Fig. 4. Elicitation model and analysis of requirements in a cyclical way.

Ethnography (immersion) was also applied to the process of prototyping and refinement of the criteria to be built in the physical layer. The applied immersion scheme is shown in Figure 5.

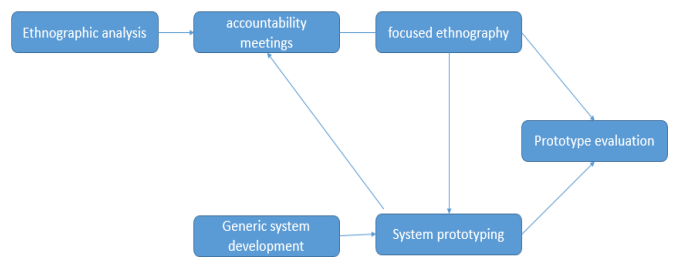


Fig. 5. Phases from the initial ethnographic analysis to the prototype evaluation model.

The IoT verification phase encompasses checking the requirements document to ensure its quality. The SCENARIOTCHECK technique [7] is applied in this phase. In the management phase, technology proposes version control of artifacts and traceability between requirements, IoT scenarios, IoT interaction arrangements and IoT use cases. Furthermore, REIoT offers change management so that changes to requirements can be reflected in artifacts.

Once the milestones on the requirements and the cyclical application of understanding and verification of their changes have been established, through the SCENARIOTCHECK technique, the requirements are consolidated for the construction of the other stages of an IoT project. Figure 6 shows the requirements change management, indicating that previous discussions are not eliminated, they are stored so that it is possible to identify the evolution of each requirement raised to be used in the system.



Fig. 6. Requirements change management.

B. RIOT Templates

The templates provided by REIoT support the elicitation (ELI) (“Project scope” artifact), analysis (ANA) (“Solution proposal” artifact) and specification (ESP) (“IoT use case description” artifact) phases. The “Project Scope” template minimally meets the design (CON), negotiation (NEG) and validation (VAL) phases.

The “Proposal for a solution” and “Description of IoT use cases” templates support the management phase (GER) by maintaining traceability between requirements and analysis models. Furthermore, the techniques described in section 3 support the elicitation (ELI), specification (ESP) and verification (VER) phases.

The subsection that follows presents the global description of the templates (project scope, proposed solution and description of IoT use cases) defined by the REIoT technology.

1) “Project Scope” Template

This template supports documenting the project’s initial activities, the problem to be solved, those involved in the project, user profiles and user needs that encompass business needs. It includes the identification and description of system requirements (functional, non-functional, restrictions, among others) and business rules. In addition, validation of the requirements document is made possible through explicit agreement (signature or email copy). The proposed template is used in the initial phases of the project, that is, the conception (CON), elicitation (ELI), negotiation (NEG) and validation (VAL) phases.

2) Template “Description of IoT Use Cases”

This template describes IoT use cases. Use cases are identified and described, providing a broad view of the system’s behavior. The use case diagram of the project is inserted in this template. Traceability between requirements, IoT scenarios, IoT interaction arrangements, and IoT use cases is maintained. This template must be used in the analysis (ANA), specification (ESP), verification (VER) and management (GER) phases. The SCENARIOTCHECK [7] technique is applied during the verification phase to identify inconsistencies in the description of IoT scenarios and their components and in the choice of AIIIs.

V. EVALUATING THE FEASIBILITY OF TEMPLATES

REIoT aims to support software engineers during RE activities. Two of the technologies used by REIoT have already been experimentally evaluated and have been used in IoT software system projects. However, the inclusion of new facilities to support RE with REIoT requires initial feasibility observation, before carrying out more robust experimental studies. Thus, this section presents a proof of concept of the feasibility of using REIoT templates in IoT software system projects.

A. Design

The proof of concept considers the structure of two artifacts (list of requirements (LR) and description of IoT use cases (DUC1)) for conventional systems that were used in IoT software systems projects; compared with the structure of the REIoT templates (project scope (EP), solution proposal (PS) and description of IoT use cases (DUC2)) described in section 3.

The LR and DUC1 artifacts were built from conventional templates in two projects that represent problems to be solved using the IoT paradigm. They are: i) Project A - software system to support the collection of data from an insole (eg: foot pressure sensors (piezoelectric), inertial sensors); ii) and Project B - software system to support the monitoring of a user’s walking in a Dashboard environment (data center, which concentrates data presented in graphic and textual form).

Researchers from the GAIT project produced the LR and DUC1 artifacts during research on possible hardware and software approaches to be applied in monitoring a user’s walking, carried out at UEA. The activity had the participation of 18 Junior researchers and 3 Senior researchers from the areas of software engineering, IoT and AI (Artificial Intelligence). These researchers were organized into three development teams, with seven participants each. The teams were balanced and contained participants with equivalent levels of software and hardware knowledge. The researchers had training and expository guidance on different software engineering topics and mentoring throughout the project. There was no intervention by the mentors in the content of the artifacts.

The treated problems represented real demand and a stakeholder (completely external to the research group) worked with the developers, including the acceptance of the requirements. Some topics covered were: requirements engineering; IoT scenarios focused on wearable devices; verification technique for IoT systems, UML diagrams, among others. The SCENARIOT and SCENARIOTCHECK techniques were presented to the participants, although they were not conditioned to use. Teams were free to organize their projects. The requirements document represented one of the project milestones.

B. Execution

After delivery of the requirements document by the teams, the LR and DUC1 artifacts produced by the two projects were analyzed. The structures of the generated artifacts, extracted based on the information found in them, were submitted to a comparison with the information structure of the EP, PS and

DUC2 templates. A checklist was used to compare the templates, which will be presented in the next subsection.

C. Results and Discussion of the Proof of Concept

Table 1 presents the checklist used to compare the structure of each of the templates (conventional systems and REIoT technology) and the result of the analysis. Analyzing Table 2, it is possible to observe that:

- The design/system objective and problem domain are not addressed by the LR template. Having knowledge about the problem domain is essential information for IoT systems [1][2];
- The LR template presents a partial description of the interested parties. It does not describe the different user profiles which is important for system and user interface development;
- The LR model does not cover the description of business needs/stakeholders. Identifying business/stakeholder needs represents the initial step of the project. At this stage, we seek to understand the customer’s real need, which in the future will be transformed into system requirements;
- Unlike the LR template which does not identify the IoT requirements, the REIoT technology identifies from the beginning the requirements that will drive the IoT solution;
- The DUC1 template treats information about IoT scenarios and IoT arrangements partially, and does not deal with the description of IoT components. Differently, the REIoT technology fully treats this information in the templates “proposed solution” (PS) and “description of IoT use cases” (DUC2);
- Traceability is partially handled by the DUC1 template and fully handled by the REIoT technology in two templates;
- The LR template presents a field for references (other documents), which is not addressed by the REIoT technology.

Project/system information	Conventional templates		REIoT Technology Templates		
	LR	DUC1	EP	PS	DUC2
Non-functional requirements	T		T		
Negotiation of requirements	T		T		
Business rules	N	T	T		T
Project analysis	N		P		
IoT scenarios	N	P		T	T
Description of IoT components		N		T	T
IoT Arrangements		P		T	T
IoT use case diagram		T			T
Description of IoT use cases		T			T
Traceability		P		T	T
References (other documents)	T		N		

^a. * P - Partially Collect; T - Collect Totally; N - Does not collect information; Gray - Does not apply

The different points of convergence and divergence between the templates (LR and DUC1) of conventional systems and the REIoT templates (EP, PS and DUC2), offer evidence that the REIoT technology is more robust, since it deals with information related to the IoT from the beginning of the project.

According to the results, it was observed that REIoT technology has significant potential for its use in IoT software systems. This is due to the fact that it incorporates specific information from IoT software systems in its templates, unlike conventional templates. The lack of this information can cause problems in the construction of the requirements document. However, to ensure the validity of the technology, an experimental evaluation is necessary to verify, among other things, whether the chain of the construction process as well as the templates are useful, complete, correct and intuitive.

D. Threats to Validity and Limitations

One limitation is the proof of concept itself, even though it is part of the technology evaluated and validated by experimental studies. However, the results show evidence that templates can capture relevant information when compared to a conventional technology, regarding the use of project artifacts.

An external threat concerns the participants (undergraduate students) who were invited to participate in the proof of concept. Regarding construct validity, there was no control over the creation of artifacts produced during the proof of concept. However, the projects were equivalent in terms of size, complexity, and IoT technologies to be used. It can be highlighted that the groups received training and mentoring in RE. Finally, the threat of conclusion is related to the study itself and the small size and homogeneity of the sample.

VI. CONCLUSION

In this work, the REIoT (Requirements Engineering for software systems based on IoT) technology was presented, which aims to provide methodological support through a constructive process (including artifact templates), technical (specification techniques and verification of IoT scenarios) and tooling for the RE of IoT software systems.

TABLE II. TEMPLATE INFORMATION STRUCTURE MAPPING CHECKLIST

Project/system information	Conventional templates		REIoT Technology Templates		
	LR	DUC1	EP	PS	DUC2
Project Name/Project Responsible	T	T	T	T	T
Version control	T	T	T	T	T
Explicit agreement	T		T		
Purpose of the project/system	N		T		
Problem domain	N		T		
Project scope	T		T		
Glossary of terms	T		T		
Description of interested parties	P		T		
Description of business/stakeholder needs	N		T		
Functional requirements	P		T		

A proof of concept was performed to compare the templates (artifacts) defined by REIoT technology with templates (artifacts) from conventional systems (not specific for IoT systems). Comparing the templates offers evidence that the REIoT technology artifacts can be more complete from the point of view of information related to IoT.

Although the area of requirements engineering is already well established, the increasing domain of techniques that increase the robustness of decisions in new contexts, such as implementing IoT systems for HealthCare solutions. In these systems, additional care is needed to ensure that the proposals are both compatible with the specifications of the hardware and software used, as well as contemplate their use in critical contexts by very specific users.

The application of the ethnography technique (immersion) proved to be very effective in discovering and confirming the types of requirements, as it allowed reflecting on the relationship in which requirements are derived from the way people actually work and not from the way process definitions say that they should work.

No sample set of related works is perfect, even when applying literature survey techniques such as systematic review; however, when the processes are well defined and clear, it is possible to build more agile and robust studies. For this reason, this survey also sought to raise the advantages and disadvantages of the techniques described, to guide the decision-making process to be adopted in new projects.

Some of the future work for REIoT technology is directed towards its use in academic and industrial projects. So it is possible to indicate some of these possibilities, such as:

- i) Development of support tools that integrate the construction process and REIoT models; and
- ii) From the requirements document generated by REIoT, create test cases to support the development of more complex and robust IoT systems.

AUTHOR CONTRIBUTIONS

The individual contributions of each author were: conceptualization, methodology: W.C.S.S.S., W.R.; validation, writing—proofreading and editing: W.C.S.S.S., V.S. and A.M.; writing—preparation of original draft: W.C.S.S.S., W.R.; supervision. V.S.; project administration, finance procurement: A.M.

ACKNOWLEDGEMENTS

This work was developed at EST – UEA – Superior School of Technology, in Manaus – Brazil.

FUNDING

This research was organized by UEA – University of the State of Amazonas. In accordance with Article 48 of Decree No. 6008/2006, it was also financed by Samsung Electronics da Amazônia Ltda, pursuant to Brazilian Federal Law No. 8387/1991, through Agreement No. 004, signed with UEA.

REFERENCES

- [1] Motta, R.C., Oliveira, K.M., Travassos, G.H.: On challenges in engineering IoT software systems. *J. Softw. Eng. Res. Dev.* 7, 5:1-5:20 (2019). <https://doi.org/10.5753/jserd.2019.15>.
- [2] Nguyen-Duc, A., Khalid, K., Shahid Bajwa, S., Lønnestad, T.: Minimum Viable Products for Internet of Things Applications: Common Pitfalls and Practices. *Future Internet*. 11, Paper 50 (2019). <https://doi.org/10.3390/fi11020050>.
- [3] Hussain, Azham, Emmanuel OC Mkpojiogu, and Fazillah Mohamad Kamal. "Eliciting user satisfying requirements for an e-health awareness system using kano model." (2015): 156-165..
- [4] Vegendla, A., Duc, A.N., Gao, S., Sindre, G.: A Systematic Mapping Study on Requirements Engineering in Software Ecosystems: *J. Inf. Technol. Res.* 11, 4:1-4:21 (2018). <https://doi.org/10.4018/JITR.2018010104>.
- [5] Akbar, Muhammad Azeem, et al. "Success factors influencing requirements change management process in global software development." *Journal of Computer Languages* 51 (2019): 112-130..
- [6] Silva, V.M.: SCENARIoT Support for Scenario Specification of Internet of Things-Based Software Systems, (2019).
- [7] Paiva, Joseane OV, Rainara M. Carvalho, and Rossana MC Andrade. "Towards a Process for NFRs Evaluation in IoT Applications." *Anais do XIV Simpósio Brasileiro de Computação Ubíqua e Pervasiva*. SBC, 2022..
- [8] Silva, D., Gonçalves, T.G., Rocha, A.R.C.: A Requirements Engineering Process for IoT Systems. In: XVIII Brazilian Symposium on Software Quality. pp. 204–209. ACM Press, Fortaleza, Brazil (2019). <https://doi.org/10.1145/3364641.3364664>.
- [9] Souza, B.P., Motta, R.C., Costa, D.O., Travassos, G.H.: An IoT-based Scenario Description Inspection Technique. In: XVIII Brazilian Symposium on Software Quality. pp. 20–29. ACM Press, Fortaleza, Brazil (2019). <https://doi.org/10.1145/3364641.3364644>.
- [10] Motta, R.C., Silva, V.M., Travassos, G.H.: Towards a more in-depth understanding of the IoT Paradigm and its challenges. *J. Softw. Eng. Res. Dev.* 7, 3:1-3:16 (2019). <https://doi.org/10.5753/jserd.2019.14>..
- [11] Giray, Görkem, Bedir Tekinerdogan, and Eray Tüzün. "IoT system development methods." *Internet of Things*. Chapman and Hall/CRC, 2017. 141-159.
- [12] Fortino, Giancarlo, et al. "Internet of things as system of systems: A review of methodologies, frameworks, platforms, and tools." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51.1 (2020): 223-236.
- [13] 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.* 12, 1589–1597 (2018). <https://doi.org/10.1109/JSYST.2016.2558449>.
- [14] Hamdi, M.S., Ghannem, A., Loucopoulos, P., Kavakli, E., Ammar, H.: Intelligent Parking Management by Means of Capability Oriented Requirements Engineering. In: Wotawa, F., Friedrich, G., Pill, I., Koitz-Hristov, R., and Ali, M. (eds.) *Advances and Trends in Artificial Intelligence*. From Theory to Practice. pp. 158–172. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-22999-3_15.
- [15] Silva, Danyllo, Taisa Guidini Gonçalves, and Ana Regina C. da Rocha. A requirements engineering process for IoT systems. *Proceedings of the XVIII Brazilian symposium on software quality*. 2019.
- [16] Mahalank, S.N., Malagund, K.B., Banakar, R.M.: Non Functional Requirement Analysis in IoT based smart traffic management system. In: *International Conference on Computing Communication Control and Automation*. pp. 1–6. IEEE, Pune, India (2016). <https://doi.org/10.1109/ICCUBE.A.2016.7860147>.
- [17] Kaleem, Sarah, et al. "A review on requirements engineering for internet of things (IoT) applications." *2019 Sixth HCT Information Technology Trends (ITT)* (2019): 269-275..
- [18] Takeda, A., Hatakeyama, Y.: Conversion Method for User Experience Design Information and Software Requirement Specification. In: Marcus, A. (ed.) *Design, User Experience, and Usability: Design Thinking and Methods*. pp. 356–364. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-40409-7_34.

- [19] Touzani, M., Ponsard, C.: Towards Modelling and Analysis of Spatial and Temporal Requirements. In: 24th International Requirements Engineering Conference. pp. 389–394. IEEE, Beijing, China (2016). <https://doi.org/10.1109/RE.2016.60>.
- [20] Costa, B., Pires, P.F., Delicato, F.C.: Specifying Functional Requirements and QoS Parameters for IoT Systems. In: 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. pp. 407–414. IEEE, Orlando, FL (2017). <https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2017.83>.
- [21] Yamakami, T.: Horizontal Requirement Engineering in Integration of Multiple IoT Use Cases of City Platform as a Service. In: IEEE International Conference on Computer and Information Technology (CIT). pp. 292–296. IEEE, Helsinki, Finland (2017). <https://doi.org/10.1109/CIT.2017.54>.
- [22] Naem, Mohammed Sharief Abdelrahman, Mouloud Koudil, and Zineeddine Ouldiam. "Product Quality Assessment in the Internet of Things: A Consumer-Oriented Approach." *Sensors* 22.6 (2022): 2215..
- [23] Curumsing, M.K., Fernando, N., Abdelrazek, M., Vasa, R., Mouzakis, K., Grundy, J.: Emotion-oriented requirements engineering: A case study in developing a smart home system for the elderly. *J. Syst. Softw.* 147, 215–229 (2019). <https://doi.org/10.1016/j.jss.2018.06.077>.
- [24] Lim, T.-Y., Chua, F.-F., Tajuddin, B.B.: Elicitation Techniques for Internet of Things Applications Requirements: A Systematic Review. In: VII International Conference on Network, Communication and Computing. pp. 182–188. ACM Press, Taipei City, Taiwan (2018). <https://doi.org/10.1145/3301326.3301360>.
- [25] Reggio, G.: A UML-based proposal for IoT system requirements specification. In: 10th International Workshop on Modelling in Software Engineering. pp. 9–16. ACM Press, Gothenburg, Sweden (2018). <https://doi.org/10.1145/3193954.3193956>.
- [26] Ferraris, D., Fernandez-Gago, C.: TrUStAPIS: a trust requirements elicitation method for IoT. *Int. J. Inf. Secur.* 19, 111–127 (2020). <https://doi.org/10.1007/s10207-019-00438-x>
- [27] Basili, V., Heidrich, J., Lindvall, M., Munch, J., Regardie, M., & Trendowicz, A. (2007, September). GQM strategies - aligning business strategies with software measurement. In First international symposium on empirical software engineering and measurement (IEEE - ESEM 2007) (pp. 488-490).