Research article

The Three-Phase Methodology for IoT Project Development☆

Luiz Carlos B.C. Ferreira [a],*, Pedro R. Chaves [b], Raphael M. Assumpção [b], Omar C. Branquinho [c], Fabiano Fruett [b], Paulo Cardieri [a]

[a] *Communications Department, Campinas State University, Campinas, Brazil*
[b] *Semiconductors Instruments and Photonics Department, Campinas State University, Campinas, Brazil*
[c] *PoB-Tec and the Extension Program of Computing Institute, Campinas State University, Campinas, Brazil*

ARTICLE INFO

ABSTRACT

The development of IoT solutions requires the expertise from professionals from several areas. This multidisciplinary aspect can be better served with an integrated process to structure the development of these solutions. There are several software development methodologies, such as Ignite and ELDAMeth, that have been adapted for this end. However, most lack some fundamental functionalities. The Three-Phase Methodology for Project Development (TpM-Pro) was originally developed as an IoT teaching methodology that proved ideal for solution development. It is a generic, agile, interactive, technology-independent and incremental methodology that splits the development of solutions into three distinct phases: Business, Requirements, and Implementation. This segmentation has proved invaluable and can become a standard for IoT solution development. In this paper, we present the TpM-Pro, gauge it against five other methodologies proposed for IoT solution development and detail its application on a corporate IoT project.

1. Introduction

The Internet of Things (IoT) is a reality. Reports indicate that billions of objects are going to be connected in the coming years, exchanging information and interacting with the environment intelligently [1,2]. In March 2022, the author of [3] wrote at IoT Analytics, a German publication specialized in market information and strategic business intelligence for the IoT, that companies' expenditure on IoT grew 22.4% in 2021 to US$158 billion. The author also forecasts that the IoT market value, including security, hardware, services, and software will grow to US$525 billion by 2027.

However, there are also challenges, according to Why IoT Projects Fail report [4], a survey conducted with 637 representatives of 250 companies distributed across 42 countries, showed that one of the barriers to success in the development of IoT projects was the higher level of knowledge in design and architecture needed. This indicated that methodologies and architectures that simplify the development process can play a decisive role. Another important finding was that often the definition of the business, its objectives, and expected returns are not taken into account, seriously contributing to the failure of IoT projects. The report concludes stating that focusing the solution on technology only is a mistake.

Also, according to [5], "eight out of ten IoT projects fail even before they are launched" and most new projects of IoT solutions are "in search of a business problem". The author of [5] adds that companies are developing solutions first and then offering them to solve business problems that may not exist yet.

IoT solutions often have many features and architectures, as a multiplicity of sensors/transducers connectivity technology, reconfigurability, connectivity to the Internet, distributed functionality, flexibility, and interoperability. Although most of these features are present in IoT systems, the criticality of each depends on the application requirements, making it difficult to define one architecture for all IoT solutions [6].

Methods and frameworks for creating IoT applications have been objects of research over the years. There is a concern about how to create and develop IoT solutions that need to consider the heterogeneity of techniques and technologies that must be used in an integrated manner [7]. These include frameworks for the creation of solutions for IoT in smart cities [8], frameworks for health and safety monitoring [9] and methods and technologies for geologic hazard prevention [10].

Also, currently, most IoT solution development depends on conventional techniques adapted from software engineering. A project methodology that considers the specificity of the IoT would be a significant contribution to the industry.

The Three-Phase Methodology for IoT Project Development (TpM-Pro) presented in this work addresses these shortcomings. The TpM-Pro was specifically formulated for the IoT, it is flexible and can be used in the development of any kind of solution. Its main differential is that it focuses on a complete understanding of the business in need of a solution and on the end-user needs.

The TpM-Pro has its origins in academia as the TpM [11], originally published in November 2019. It is a technology-independent methodology, used by students to manage the many disciplines involved in developing IoT solutions. The TpM presents a generic way to structure and plan a project, as well as manage and integrate the development team.

The TpM follows a consolidated three-phase process that starts with the Considering the Business Phase, followed by the Gathering of Requirements Phase, and concludes with the Implementation Phase. The TpM encompasses the IoT Open-Source Reference Model (IoT-OSRM) [12]. This model helps to analyze the IoT solution logically thus facilitating requirement acquisition and identification of possible implementation difficulties. The open nature of the IoT-OSRM ensures the TpM's flexibility and adaptability.

The TpM has been extensively applied in the academic environment over the last three years, in extension and postgraduate courses, with more than a hundred students, as seen in [11,13,14]. During its academic use, it was observed that the methodology has great potential to be used in the development of corporate IoT solutions.

This led to the development of the TpM-Pro, an expansion of the original TpM. The TpM-Pro adds to the original TpM as it incorporates project specific attributes like agents, roles, artifacts, and the TpM-IoT-Canvas.

Thus, an contribution of this work is to present the TpM-Pro as a methodology for corporate IoT projects, drawing from the experience acquired during its use in the academic world.

The TpM-Pro could be used by teams developing IoT projects in virtually all areas, such as banking, finance, energy, manufacturing, mining, etc.

To test the TpM-Pro in corporate projects, a consultancy was offered to a team developing temperature and mechanical vibration monitoring devices for industrial rotating machines. The consultancy covered all aspects of the TpM-Pro. The results of a survey carried out with the development team, at the end of the project, showed that the methodology was well accepted.

Therefore, the main contributions of this work are:

- A literature review of development methodologies used for IoT is presented;
- Description of the TpM-Pro, an original, consolidated and generic methodology for developing IoT projects, focused on the business and the problem in need of a solution;
- An analysis of the TpM-Pro in relation to the development methodologies covered in the literature review;
- A description of its application in a corporate project.

The rest of this article is organized as follows: Section 2 presents five different development methodologies for IoT solutions, comparing them to the TpM-Pro, while Section 3 details the TpM-Pro for IoT project development. Section 4 presents a case when the TpM-Pro was used in the development of a corporate project. The article ends with the conclusions in Section 5.

2. System development methodologies - Literature review

The selection of works covered in this literature review favored the following criteria: shortest time since publication, higher publication impact factor, larger number of citations and better known authors, following what is presented in the work of [15].

The literature lists some characteristics considered by [16–18] when classifying IoT methodologies, namely: interoperability, autonomy, scalability, and smartness.

(a) Interoperability: "The extent to which systems or products can exchange and use the information that has been exchanged" [19]. In IoT, interoperability entails that, regardless of their characteristics, systems can communicate and interact with each other.

(b) Autonomy: The authors of [20] define autonomy as an operating domain constrained by parameters that guarantee the operation without supervision. In the IoT context, autonomy implies that systems are independent to operate and self-manage.

(c) Smartness: It is defined by [21] as the capability of a system to autonomously act on data gathering, information processing, knowledge acquisition, and actuation. For the IoT, smartness relates to emulating the human behavior and the processes of learning and adapting.

(d) Scalability: "Ability to handle an increasing amount of work while adapting to environmental change" [22]. Any IoT system must offer support for an increasing number of devices and applications without affecting its performance.

Table 1
Comparison between IoT Development Methodologies.

Methodology - Origin	Characteristics
TpM-Pro – Academy	**Objective:** to offer a flexible and adaptable technology independent project methodology for IoT project. **Main aspects:** considers the complexities of IoT solutions and the need for a manufacturer independent approach. Approaches the solution as a whole, with the business at its core. Covers agents and roles, business, requirements gathering and implementation. Emphasis on documentation. **Interoperability:** supported. **Autonomy:** supported. **Smartness:** supported. **Scalability:** supported.
Resilient-IoT – Academy	Objective: develop and implement components that will work at the edge of an IoT solution. **Main aspects:** proposes that some basic entities like Environment, Problems, Actors and Coordinators must be defined. **Interoperability:** limited. **Autonomy:** supported. **Smartness:** supported. **Scalability:** supported.
SPP – Academy	**Objective:** identify similarities between agents to develop an reference architecture; cost and implementation time reduction, and better quality. **Main aspects:** in-line production of software; use of agents and product line engineering software. Does not consider the business, nor the implementation. **Interoperability:** supported. **Autonomy:** limited. **Smartness:** supported. **Scalability:** supported.
HLAD – Academy	**Objective:** application development based on Model Driven Design approach and on the work in macro programming of sensor networks. **Main aspects:** identifies different interests in a conceptual model. Implements the methodology as a development framework. Segments the process into areas of interest. Does not consider business needs. **Interoperability:** supported. **Autonomy:** supported. **Smartness:** limited. **Scalability:** supported.
ELDAMeth – Academy	**Objective:** to enable rapid prototyping based on visual programming, validation and automatic code generation. **Main aspects:** defines high-level characteristics and technical aspects. Can include problem statement, define "things", agents, etc. Does not consider the business or the requirements. **Interoperability:** supported. **Autonomy:** limited. **Smartness:** limited. **Scalability:** supported.
Ignite – Industry	**Objective:** to deliver best practices in the form of reusable, technology-independent and open-source methodology. **Main aspects:** provides project templates, checklists and solution architecture schemes. Defines the process until delivery of the operational IoT system. Does not provide technical support. **Interoperability:** supported. **Autonomy:** limited. **Smartness:** limited. **Scalability:** limited.

The development of software systems benefits from the use of Systems Development Methodologies (SDMs), as they help the final product to meet functional and non-functional project requirements [23].

SDMs guide the development of systems, providing guidelines on different aspects such as stakeholders' roles and artifacts to be produced. SDMs also formalize procedures, ensuring consistency in the development phases, logically ordering processes, and preventing important aspects from being neglected [24,25]. SDMs provide a set of standards and goals to be achieved and make it easy to verify if the final solution meets the previously defined requirements, allowing the monitoring of the project's progress and contributing to the coordination of different teams [26].

The IoT involves software, hardware, and communication components. The development of IoT projects can capitalize on some aspects of the SDMs created for the development of software projects. The IoT, however, needs a broader approach, as its inherent multidisciplinary nature makes the development process of these projects more complex, as it involves professionals from different areas of knowledge [27].

The function of an IoT system is to connect a user to a virtual representation of an entity in the physical world, i.e. the representation of a "thing", be it a sensor, actuator, tag or physical magnitude [28].

In addition to SDMs, there are generic options for developing complex distributed systems. An example is the Reference Model of Open Distributed Processing (RM-ODP), defined by ITU-T Rec. X.901-X.904, ISO/IEC 10746 [29]. The RM-ODP is a framework for system specification that incorporates viewpoints, such as computational and enterprise viewpoints. However, it is a generic standard, developed in 1998, with several documents and specifications. It is an example of a framework that was not specifically created for the IoT that can be adapted to that end.

The literature lists some SDMs already adapted to the development of IoT systems, originating both in the industry and in academia. We have identified five SDMs that have a detailed structure that can serve as a benchmark for the TpM-Pro. Table 1 presents the comparison between the TpM-Pro and these methodologies according to their characteristics.

Interoperability was measured taking into account the level of abstraction and independence of technologies, provided by the methodology. Most of them call upon model-driven engineering (MDE), which offers an approach to engineering interoperable solutions by means of Object Management Group's (OMG) established modeling standards, such as the unified modeling language (UML) and metalanguages, such as XML-like.

The autonomy in IoT systems is largely acknowledged as a cross-domain and cross-layer key feature. It is interesting that the SDMs provide a methodological approach to this aspect. The scale, dynamics and complexity of IoT systems make a fully human management difficult, often, the management of the solution must be hybrid, divided into human aspects (administrators, developers, users) and software and hardware agents. To assess this characteristic, the SDMs were verified in order to find a way to define these responsibilities, either at a methodological level (high level of abstraction) or implementation (programming languages, technologies and techniques).

IoT methodologies address the issue of scalability from both programming and architectural perspectives. From a programming perspective, this view takes into account technology aspects, such as macro-programming approaches to group devices in hierarchical

clusters based on their spatial relationships. The architectural perspective involves how the methodology provides a generic framework for scalability to be thought through and planned. This approach is the focus of this work and was the focus of the analysis of the SDMs presented.

These were evaluated as a way to implement intelligence in IoT solutions. These must be generic, leaving space to plan the actions according to the needs of the problem to be solved by the solution. The more independent of specific techniques (e.g., deep learning and reinforcement techniques), the better, since techniques and technologies are constantly changing, for that reason, it is not interesting to set standards.

Development methodologies are basically a set of phases in the life cycle of a project. Some emphasize requirements engineering and development, not concentrating on the business aspects, such as SPP, which only covers software development aspects for components, or HLAD, which produces an implementation specification and support for the maintenance phase; or even ELDAMeth, composed of three distinct phases, which uses software operating autonomously. Other methodologies, such as Ignite, address the solution development cycle, including aspects of the business, project management, feasibility, requirements engineering, and design, but do not provide details on how to develop and test solutions.

2.1. Engineering Resilient Collaborative Edge-enabled IoT

The Engineering Resilient Collaborative Edge-enabled IoT (Resilient-IoT) was published by Casadei et al. [30] in 2019. It presents a methodology for resilient collaborative engineering systems, focused on the development of heterogeneous components that can be implemented on different devices. These devices can work collaboratively to perform functions. It is a specific methodology for developing and implementing components that will work at the edge of an IoT solution, i.e., closer to the target environment.

The methodology assumes that there are domains of knowledge and problems that are already known, which can serve as a standard for the use and reuse of implementations and solutions. It proposes a technical implementation based on Aggregate Computing, a formal programming paradigm able to capture adaptive behavior of agent collectives.

The work proposes some basic entities that must be defined:

- Environment: a domain where the solution is going to operate.
- Problems: challenges that must be overcome by the solution.
- Actors: entities that will interact in the IoT solution, such as humans, sensors, actuators, and robots.
- Coordinators: structure with greater computational capacity that acts at the edge, coordinating and managing actors.

This SDM does not present explicit interoperability characteristics. Although the model itself describes how interactions should occur between components, interoperability solutions are not presented. Regarding autonomy, the approach used says that IoT devices (re)act autonomously to stimuli, through data-oriented techniques, i.e., machine learning, artificial intelligence, etc., and coordination mechanisms (consensus). Scalability is achieved with a macro-programming approach to grouping devices into clusters, based on their spatial relationships, e.g., smart room objects. This scale abstraction is treated through spatial operators and middleware components, which allow logical scopes and support IoT systems ranging from a few to several components (horizontal scalability). The methodology meets the intelligence requirement, proposing the use of deep learning and reinforcement techniques to perform the detection of patterns and anomalies, in addition to the use of predictive analysis and resource optimization.

2.2. Software Production Process

The Software Production Process (SPP) for the development of agents has its origins in academia and was proposed in 2015 by Ayala et al. [31]. It can be understood as an in-line software product production process designed to improve the development of IoT applications using agents and Software Product Line Engineering (SPLE). SPP follows the two main processes established in SPLE: Domain Engineering, used to establish common characteristics in the platform, and Application Engineering, used to derive applications created specifically for the platform. The objective of the initiative is to identify similarities between agents and develop a common reference architecture, providing a reduction in cost and implementation time, and an increase in quality. The SPP is documented by journal articles and conference documents.

The SPLE seeks to develop software and identify common and distinct points in a family of software products. Common points ensure platform re-usability while distinctions refer to specific characteristics of certain software within the scope of a product family. A multi-agent IoT system architecture, which presents similarities and distinctions, is produced after a variability model is defined. Therefore, a multi-agent system architecture acts as a base model, forming the common platform and points of difference to be configured in each autonomous software component (agent). Developing software components for "things" using agents can be convenient due to their distributed nature.

2.3. High-level Application Development for the IoT

The High-level Application Development for the IoT (HLAD) was presented by Patel and Cassou [32], originating in academia. HLAD is documented in a doctoral thesis and a journal article. This SDM aims to facilitate the development of IoT applications, based on the Model-Driven Design approach and the work in macro programming of sensor networks. The proposal identifies different interests represented in a conceptual model. The identified concepts are linked together in a well-defined methodology that can be used as a development framework. The HLAD segments the development process into four areas of interest, namely:

(1) Domain Area: deals with the specific concepts of IoT systems.
(2) Functional Interest Area: looks at the system architecture and how its implementation should take place.
(3) Deployment Area: specifies the "things" that will be addressed by the solution.
(4) Platform Area: addresses the development of specific drivers for each type of virtual entity or "thing".

2.4. Event-driven Lightweight Distilled State Charts-based Agents Methodology

The Event-driven Lightweight Distilled State Charts-based Agents Methodology (ELDAMeth) project [33] was conceived in academia and aims to provide a simulation-based methodology. This proposal uses an agent to drive software development into the "things" in IoT systems. These agents can be defined as software operating autonomously in a networked environment.

The ELDA model is based on three main concepts to enable distributed computing: Agent Behavior, driven by events that trigger reactive and proactive computing; Agent Interaction, based on multiple coordination spaces that are explored by agents; and Mobility, which allows agents to migrate autonomously or to be passively migrated between servers. This methodology comprises three distinct phases:

(1) The Modeling Phase produces the Multi-Agent System (MAS) specification. The ELDA-MAS can be converted to platform-independent code through the ELDA Framework, a set of Java classes that formalize modeling.
(2) The Simulation Phase produces the simulation assessment document, including performance indices to be assessed against functional and non-functional requirements.
(3) The Implementation Phase produces a code based on the MAS. Testing the MAS code implemented allows the assessment of functional and non-functional requirements. If issues are detected, this assessment can trigger a new iteration process.

The ELDAMeth focuses only on the technical aspects of the solution, it does not cover the business aspects or in-depth definition of requirements.

2.5. Ignite-IoT Methodology

The Ignite Methodology [34] had its origins in the industry and is documented through a book that defines the method and provides information on business projects. Ignite aims to deliver IoT best practices in the form of a technology-independent, reusable, open-source methodology. It aspires to support the design, configuration, and management of IoT projects by providing project templates, checklists, and solution architecture schemes.

Ignite, however, does not provide technical details on how to develop and test software. The methodology is divided into two phases.

In Phase A the strategy is outlined, the business opportunity is identified and the management is defined.

Phase B comprises the delivery of the IoT solution. In this phase, the individual IoT solution and related projects are analyzed. This phase is subdivided as follows:

(1) Solution Life-cycle - Comprises planning, construction, testing and commissioning of the IoT solution:

   (a) Initial Design: based on the elements defined in previous designs contained in the Generic Building Blocks.
   (b) Workflows: define the flows to be implemented in the project. A checklist for each workflow is provided, as well as a list of common dependencies between the workflows.

(2) Generic Building Blocks — Repository with reusable models of successful projects, including:

   (a) Project Dimensions: precursor of the project's formal requirements. Used for project evaluation, comparisons between projects and architectures, technologies selection, etc.
   (b) Architectural Schemes: builds on existing architectures, adds new perspectives needed to the design and provides a superstructure to integrate the different perspectives needed.
   (c) Technology Profiles: identifies and describes technologies for IoT projects. It uses the perspectives of the IoT architecture to describe where different technologies fit together.

(3) Project Database: a repository of reference projects analyzed to extract best practices for the Building Blocks and Solution Life-cycle perspectives.

Ignite allows different platforms to interoperate. Objects defined and written to Ignite by one platform can be read and used by another platform.

3. The Three-Phase Methodology for IoT Project Development

Traditionally, Methods Engineering deals with the processes of designing, building, and adapting methods directed to the development of information systems [35]. According to [36], an information system can be understood as a subtype specific to a work system. Objects that must be designed, integrated, or transformed through a methodology can be defined as a work system. An IoT solution involves the integration of several subsystems, making it a work system.
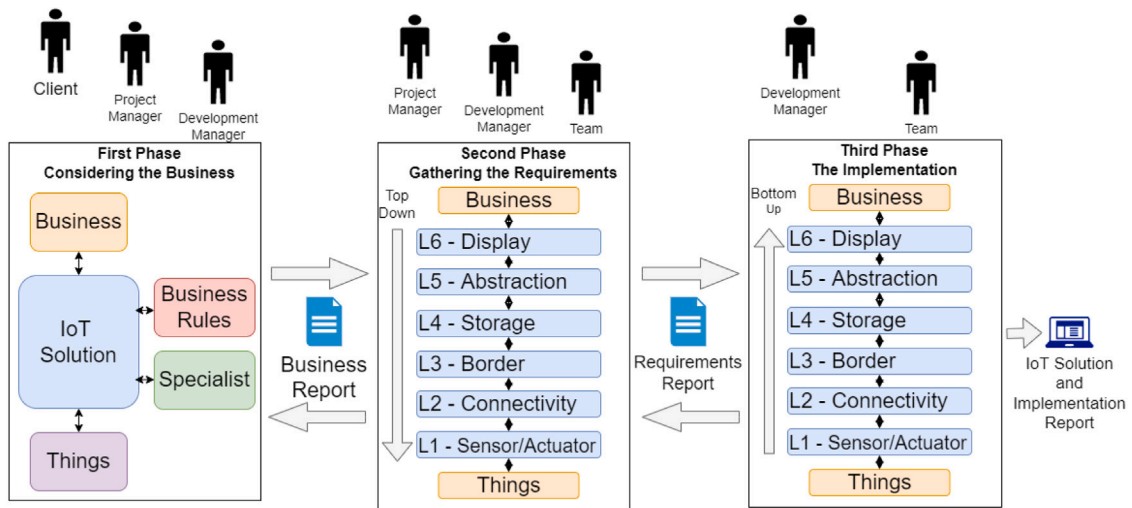
Fig. 1. TpM-Pro iterations diagram showing roles, phases and deliverables.

To be applicable to the development of IoT solutions, methods need to be adapted to the characteristics of the development or project situation. This approach is commonly called Situational Methods Engineering [37–39].

For the development of the TpM-Pro, the Situational Method Composition approach was employed [35]. The fundamental idea behind this method is the selection and orchestration of artifact fragments with respect to the specifics of the development of a class of projects. This methodology composition process aims at combining several levels to establish new construction results. This approach to methods engineering is widely used and discussed in the literature [37,39–41].

The Situational Method Composition follows three steps to create a methodology:

(1) Identify situational characteristics: these characteristics can be used to identify types of projects to be developed with the methodology, as well as artifacts and artifact fragments.
(2) Breakdown generic artifacts into artifact fragments: to elaborate the methodology, generic artifacts need to be broken into artifact fragments. In addition, artifact fragments and their interrelationships need to be identified.
(3) Assemble the artifact fragments in a methodology: the actual composition of a methodology occurs by choosing and orchestrating fragments of artifacts according to well-defined construction or composition principles, to suit the characteristics of the projects to be developed with the methodology.

On the TpM-Pro, step 1 is represented by IoT projects, as these have specific characteristics, i.e., heterogeneity of technologies and techniques, and the need for hardware/firmware/software integration. Artifacts in the TpM-Pro are defined as the three phases that make up the methodology.

In step 2, the phases (artifacts) are broken-down into artifact fragments. These fragments are represented by the information regarding the business and the IoT-OSRM reference model. The relationships between these fragments are also defined.

In step 3, the organization of the methodology happens. There the flows, actors, and products generated by TpM-Pro are defined.

Therefore, the TpM-Pro is based on a well-defined and consolidated engineering method.

Fig. 1 shows the diagram of the phases that make up the TpM-Pro, as well as the agents and roles identified in the methodology.

It is important to highlight that the TpM-Pro is an iterative and incremental methodology. There is a cycle that can repeat itself until a suitable solution is found. This feature makes the TpM-Pro flexible, as it can be used during the lifetime of solution, from delivery to updates and enhancements.

Fig. 2 shows this cycle, where the team can go from one phase to another, to adjust objectives, clarify doubts, meet new demands or for any situation where feedback is required. Therefore, the three phases that constitute the TpM-Pro are enough to meet all the stages of creation and development of an IoT solution. This makes the methodology simple, systematic and structured, making it an important contribution to the IoT.

A system to assist in the documentation of IoT projects according to the TpM-Pro has been developed, validated, and tested. It is meant to be a guide where development teams can collaboratively apply the TpM-Pro, documenting and organizing the project. In addition, the system also aims to connect the TpM-Pro using the community. There, professionals, academics, and others interested in IoT can exchange experiences and knowledge. The system can be accessed at http://www.iotm3f.cc. More information can be found in [14].
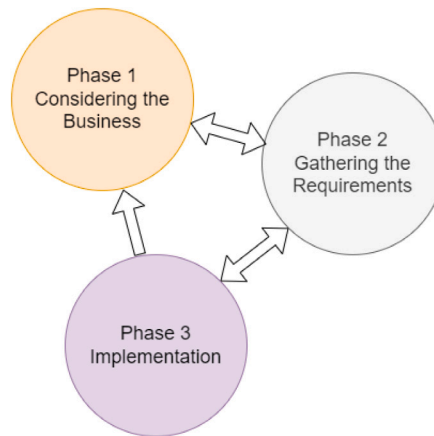
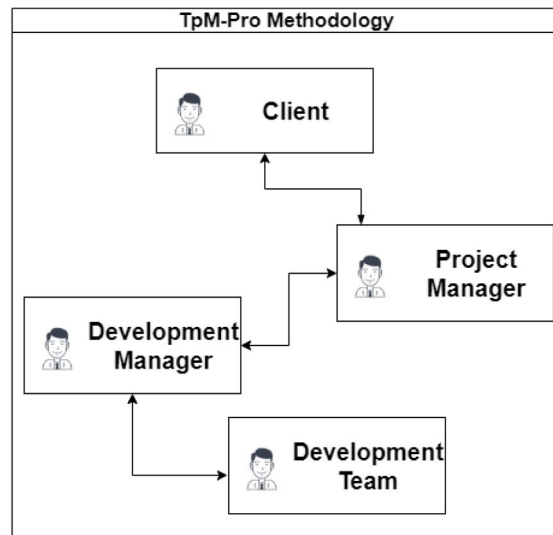**Fig. 2.** Iterations between the TpM-Pro phases.

Fig. 3. Interactions between agents in the TpM-Pro.

3.1. Agents and roles

Every development methodology needs to have its agents and roles well defined. This definition is important for the organization and structuring of solutions [42]. The TpM-Pro has the following agents performing the following roles, as shown in Fig. 3:

- Client: includes all parts served by the solution;
- Project Manager: acts as the link between the Client and the Development Manager. Responsible for extracting information of the business and verifying the feasibility of the solution together with the other players. All communication between the Client and the Development Manager is handled through the Project Manager;
- Development Manager: responsible for managing the multidisciplinary team. Defines the parameters related to the development of the solution. This role also defines responsibilities at every level of the IoT-OSRM and tracks the project development. Responsible for organizing the requirements gathering on Phase 2, acts as a bridge between the multidisciplinary team and the Project Manager. Can also be in contact with the Client;
- Multidisciplinary Development Team: formed by professionals with knowledge at the different levels of the IoT-OSRM, is responsible for the development process.

Defining roles is important for the organization and communication hierarchy in the development process. This organization helps to integrate the team, as the responsibilities and needs are made explicit and well delineated. Fig. 4 presents the flowchart of a TpM-Pro iteration indicating the relationship between agents, roles, processes, and deliverables.
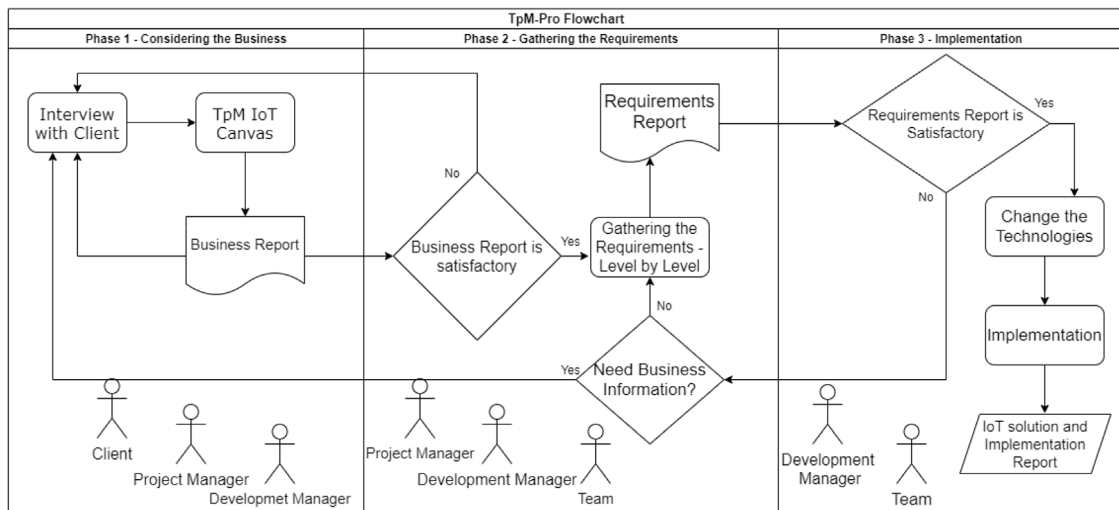
Fig. 4. Flowchart showing the IoT development solution process.

3.2. Phase 1 - Considering the business

In Phase 1, the business is understood and the problems to be addressed by the solution are outlined. The TpM-Pro is mainly concerned with the value brought by the IoT solution to the business. This is its main differential in relation to other IoT methodologies. The principle is that if the business is not well defined and understood, the solution developed will not meet the customer's needs. This approach's main objective is to meet the customers' needs in its entirety, understanding their anxieties and expectations. To define the problem to be solved, the following aspects are considered:

- The Business: varies from project to project, even when dealing with projects in the same area of activity, as each will have its peculiarities. If not well pondered, the end solution may not satisfy the end-user necessities;
- The Things: objects of interest to the business; entity or set of entities to be quantified, measured or controlled, can be physical or virtual. In this phase, the Project Manager must define what the "things" are;
- The Specialist: every solution requires the involvement of specialized personnel who will provide information to support the business decision-making processes. These can be in-house professionals or hired for this purpose;
- The Business Rules: assumptions and restrictions applied to the operation of any business. Therefore, these rules must be taken into consideration throughout the solution development process. Business rules can be determined by analyzing the target scenario in which the solution is to be employed. Developing an IoT solution begins with a clear understanding of the rules that govern the business and should be consistent with the customer's needs.

To carry out Phase 1, we developed the TpM-IoT-Canvas, a tool conceived to assist in extracting the necessary information about the business and expected solution. The TpM-IoT-Canvas was specifically designed for IoT projects.

The TpM-IoT-Canvas is based on the Project Model Canvas (PM-Canvas) [43]. It encourages the whole team to help planning a project with the use of post-its pasted on a board. The TpM-IoT-Canvas also simplifies the understanding of the business and solution by relying on small groups of key information presented in a visual model.

There are 8 blocks that cover the following topics: business, justification, benefits, product, things, solution requirements, client, and team. Each block is directly related to its neighbor. This feature allows easier identification of the impacts changes in one area have on others; in case any inconsistencies are found in the project.

The TpM-IoT-Canvas shown in Fig. 7 provides a color coded visual understanding of the project's key information grouping, seeking to respond to three defining questions:

(1) Why? - (In orange) - Encompasses the justifications, objectives and expected benefits of the project. It should clarify the relevance of the solution, considering:

    (a) Business: defines the enterprise's area of expertise, core activity, area in which the proposed solution operates, and its relevance to the company's activities;

    (b) Justification: lists the problems that the solution must address;

    (c) Benefits: Tangible and intangible value proposition to be achieved with the project. These propositions should justify the development of the solution.

(2) What? - (In grey) - Describes the solution to be delivered to the customer. It is essential to consider the expectations of sponsors and customers, while also considering the following:
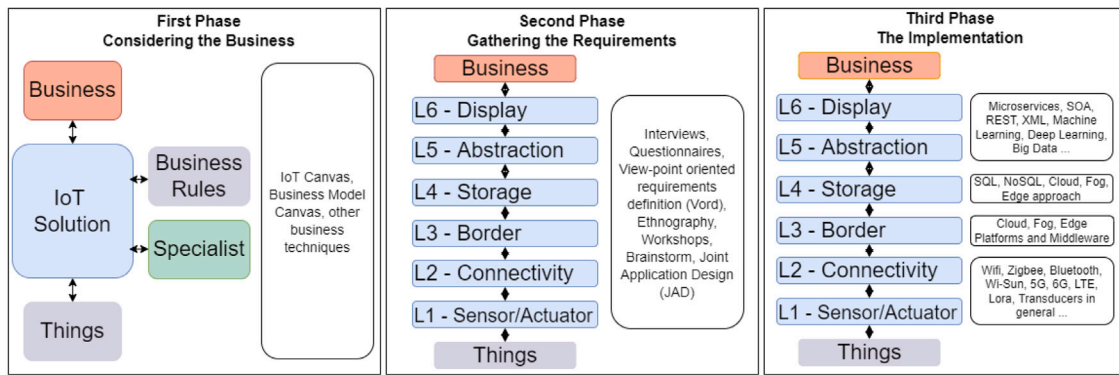
Fig. 5. Examples of techniques and technologies that can be used with the TpM-Pro.

(a) Product: definition of the solution in terms of its basic form e.g., equipment, software platform, systems integration, service platform;

(b) Things: in addition to a list of the things that interact with the solution per se, it is important to list what should be measured, controlled or quantified of each thing;

(c) Solution requirements: everything that is required from the solution must be listed here. This field reflects the business rules, and justifications. This field will also serve as a starting point for Phase 2.

(3) Who? - (In green) - Defines the incumbents, as previously defined in sub-section Agents and Roles.

The result of Phase 1 is consolidated in an artifact called Business Report, prepared by the Project Manager with the Development Manager and team. This document feeds into Phase 2 of requirements gathering.

The Business Report is then evaluated by those responsible. If approved, Phase 2 begins, otherwise, a new analysis of the business is carried out. Therefore, the Business Report is the artifact used to validate a Phase 1 interaction.

3.3. Phase 2 - Gathering of requirements

Phase 2 starts right after the approval of the Business Report. With the business well understood, it is possible to define the functional and non-functional requirements. The functional requirements define the functions or services that the system must encompass, that is, what the solution must do in terms of tasks and services. The non-functional requirements, on the other hand, define properties and restrictions on the system design, with no direct relation to the functionality, that need to be incorporated into the system, such as the technical standard to be followed and forms of system updating.

In this phase, the process is divided into six levels, following the IoT-OSRM reference model. Developers can use past cases or techniques such as interviews to get the information. This is an important step in the development process in which the structured format of the IoT-OSRM is an asset.

A top-down approach is used in this phase. It starts by defining the requirements on the level closer to the business, moving down the levels to the one closer to the "things". All information gathered must be documented:

- Level 6 - Display: handles how information is to be displayed to the end-user. Information can be presented in tables, graphs, or numbers, for example. Emergency and other urgent actions are also flagged and dealt with at this level;
- Level 5 - Abstraction: stored data is used at this level to convert raw data into information. At this level that the expertise of a specialist is required. Developers depend on expert information to decide how to extract abstraction from the available information. Techniques such as artificial intelligence, data mining, and machine learning, can be employed at this level;
- Level 4 - Storage: defines how the collected data should be stored. Data can be stored in the cloud, on the client's system, or both, depending on the required level of availability and security and the amount of data. Aspects such as redundancy, data security, and storage location are also defined at this level;
- Level 3 - Border: defines the characteristics of the element that connects the solution to the Internet, the so-called border element. This element warrants the connection between the "things" with the part of the system that is on the Internet. Therefore, the border element must always be present in an IoT system. In addition, this element may be required to perform functions as middleware and networking using more advanced strategies (including software-defined networks and virtualization of network functions). Developers should pay attention to this level, as it will guide the implementation phase;
- Level 2 - Connectivity: focus on the connectivity between "things" and border element. This level can be considered the bottleneck of any IoT system. The connectivity can be wired, wireless, or hybrid. Wireless technology is typically employed in the context of IoT, especially when many sensors/actuators are distributed over a large area. However, the decision on the type of technology to be employed must consider some other issues, including costs, number of items to be monitored or
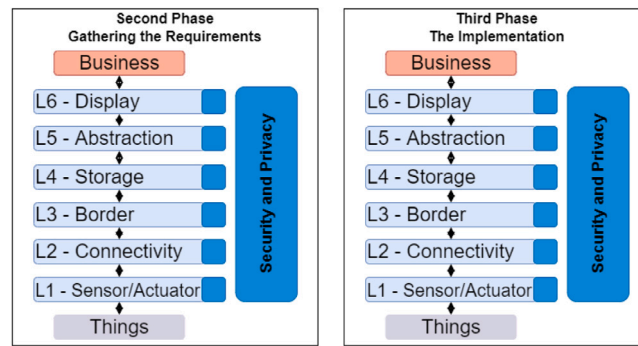
Fig. 6. Security and privacy in TpM-Pro.

controlled, operating environment, required capacity, and reliability of the transmission system. Requirements at this level are defined by developers with knowledge in the field of data communications;

- Level 1 - Sensor/Actuator Node: refers to the devices responsible for data collection or action in the environment. At this point, developers specify the capabilities expected from sensors and actuators, based on the customer necessities. Must consider local processing, use of edge or fog computing concepts, etc. Documentation must be created describing what is to be measured, controlled, etc.

Security and data privacy are perennial concerns regarding IoT solutions. This topic has been addressed by government and regulatory institutions, as in [44,45]. The TpM-Pro is prepared to meet this demand. In it, security must be defined at each level of reference model. There is no single tool that provides security for an IoT solution, but rather, a set of tools and techniques that, working together, provide the necessary security and privacy.

Fig. 6 shows that on the TpM-Pro, the security and privacy of the solution is transversal. Each level has its specific requirements of security. Proper definition of security for each level will ensure the privacy and security of the whole solution. Therefore, in the Gathering of Requirements phase, security and privacy concerns must also be considered.

Once each level of the IoT-OSRM reference model has been defined, a Requirements Report is prepared. It is a document that gathers all functional and non-functional requirements.

The Requirements Report is then validated by the Project Manager, Development Manager, and development team. If the report is approved, Phase 3 begins, otherwise, a new round of Phase 2 is carried out to resolve any pending issues. Therefore, the Requirements Report is the artifact used to validate the completion of an iteration of Phase 2.

The TpM-Pro has different characteristics from the other methodologies, for example, its requirements gathering method is not rigid. The path to be followed, i.e., the IoT-OSRM, is indeed systematic, but the techniques used can vary, as shown in Fig. 5. This brings flexibility. The actors involved can use knowledge previously acquired to use the TpM-Pro.

3.4. Phase 3 - Implementation

Phase 3 only starts once the Requirements Report is approved. Developers must now evaluate the technology options available and choose what best meets the specifications defined in the previous phase. The Implementation Phase also follows the IoT-OSRM reference model, however, a bottom-up approach is used in this phase, as it is first necessary to define how the "things" are to be monitored and controlled and how this information is to reach the end-user. The evaluation follows the sequence:

- Level 1 - Sensor/Actuator Node: at this level the sensors/actuators that will be used are defined and developed. The micro-controllers, memories, connectors and transducers are chosen here.
- Level 2 - Connectivity: the transceivers, communication protocols, antennas and all equipment needed to establish the communication between the Things and the edge element are selected at this level.
- Level 3 - Border: the edge element is selected at this level. It can be a single board computer if an edge computing approach is chosen, or it can be a cloud server if a inter cloud computing approach is decided for. According to the requirements raised in Phase 2, the choice of technology to be used is defined at this level.
- Level 4 - Storage: based on the requirements, the data storage strategy that best fits the solution is chosen at this level. It could be a single local database or a local database and others in the cloud. These can be relational or non-relational databases.
- Level 5 - Abstraction: data-handling techniques are implemented at this level. These can be machine learning, deep learning or big data, to mention just a few. Here the data is processed to generate useful information for the end user, i.e., the Client.
- Level 6 - Display: this is where the tools that will deliver the results to the Client are put in place. These can be apps, websites, phone messages, etc., showing graphs, alerts, tables, etc.

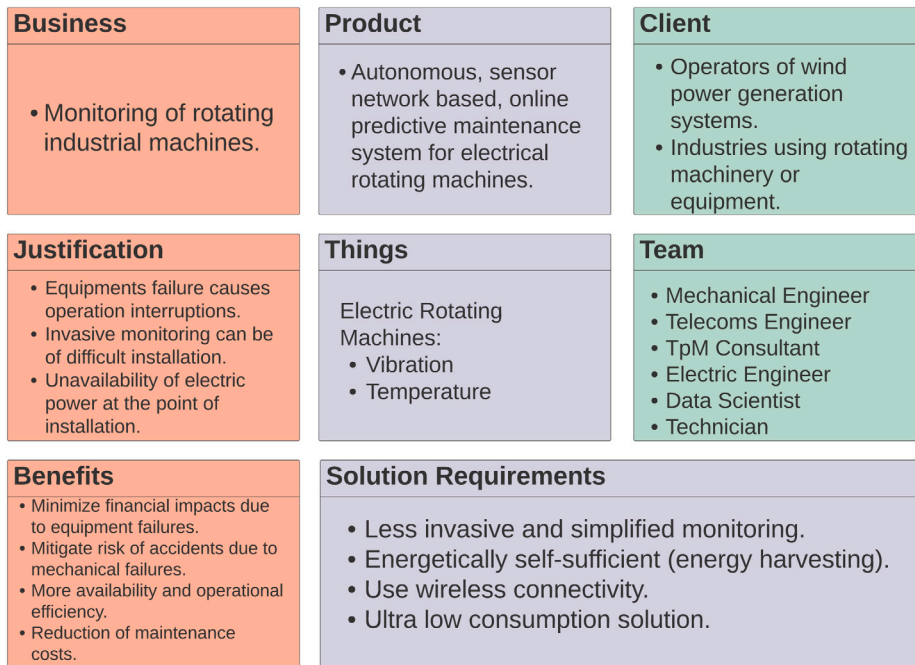| Business	Product	Client
• Monitoring of rotating industrial machines.	• Autonomous, sensor network based, online predictive maintenance system for electrical rotating machines.	• Operators of wind power generation systems. • Industries using rotating machinery or equipment.
Justification	**Things**	**Team**
• Equipments failure causes operation interruptions. • Invasive monitoring can be of difficult installation. • Unavailability of electric power at the point of installation.	Electric Rotating Machines: • Vibration • Temperature	• Mechanical Engineer • Telecoms Engineer • TpM Consultant • Electric Engineer • Data Scientist • Technician
Benefits	**Solution Requirements**	
• Minimize financial impacts due to equipment failures. • Mitigate risk of accidents due to mechanical failures. • More availability and operational efficiency. • Reduction of maintenance costs.	• Less invasive and simplified monitoring. • Energetically self-sufficient (energy harvesting). • Use wireless connectivity. • Ultra low consumption solution.	

Fig. 7. The i-Machine project TpM-IoT-Canvas.

It is important to emphasize that techniques of security and privacy are implemented at each level. From Level 1, where physical security measures are applied to sensors/actuators, up to Level 6, where user access controls are implemented. All tools and techniques implemented at each level should provide the appropriate level of security, as shown in Fig. 6.

The TpM-Pro is technology independent, it differs from other methodologies as it is flexible, adaptable, and can make use of documentation and implementation techniques already consolidated in the literature. However, it always follows the same structure, and always uses the business as a starting point. This concept integrates all techniques and knowledge needed to develop a solution.

Phase 3 results in the delivery of the Implementation Report and the implementation of the solution itself. If needed, the process of refinement and improvement can then begin, going through all three phases again. The TpM-Pro is iterative and incremental, through it, the process can be repeated as many times as necessary until the solution meets the business needs.

In summary, as shown in Fig. 5, the TpM-Pro can be a guide for the development team. Known techniques and architectures can be used during Phases 1, 2, and 3. The main differential, however, is that the TpM-Pro makes it clear how these techniques must be integrated to meet the objective of any IoT solution, that is to attend to the needs of a business.

4. Results

The mainly descriptive and exploratory case study presented in this section followed the Association for Computing Machinery guidelines for conducting and reporting case studies [46]. These guidelines set down the structure a study should have, i.e., objectives, data collection and analysis mechanisms, presentation and interpretation practices, and the way information should be gathered, i.e. qualitative and/or quantitative methods.

Differently from the methodologies presented in Section 2, the TpM-Pro follows the premise that before any choice of technology is made, it is necessary to clearly define the problem and understand the business in need of the solution. The TpM-Pro is a generic, manufacturer, and technology-independent methodology that facilitates the identification of possible difficulties during the project's development cycle. The TpM-Pro encompasses the main aspects that must be considered in the development of any solution, namely: business, "things", team, requirements, technologies, implementation strategies, and documentation.

The TpM-Pro was adopted by a team developing *i-Machine*, a government-funded project to provide an online predictive maintenance service for electrical rotating machines, with the use of fuzzy logic in the analysis of vibration and temperature parameters. The service relies on autonomous wireless sensor networks. Each sensor node incorporates a thermoelectric generator (TEG) to power the system, a DC-AC converter to manage the power, a super-capacitor to store the energy, and a micro-controller (MCU) to manage the communication module, as shown in 8. Potential customers include enterprises that rely on difficult to reach rotating machines in their processes, such as wind power operators and companies that employ industrial pumps and cranes.

As previously indicated, when using the TpM-Pro, it is essential to start with a well-defined problem that needs to be addressed. In the *i-Machine*'s project, the problem the solution had to address was "the detection of physical signals that indicate possible future
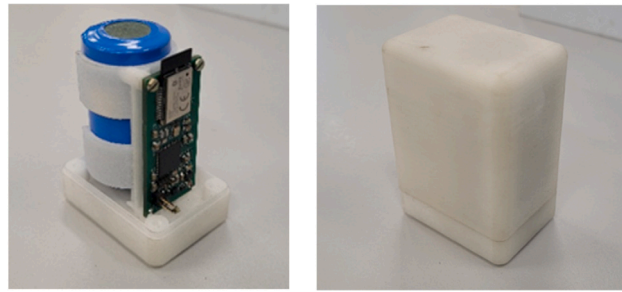
Fig. 8. On the left, the complete i-Machine sensor node, including the battery, on the right, the encapsulated sensor node.
Source: Adapted from the i-Machine project.

Table 2
I-Machine team survey results.

Statement	Agree	Somewhat agree	Neither agree nor disagree	Somewhat disagree	Disagree
1	80	20	0	0	0
2	80	20	0	0	0
3	100	0	0	0	0
4	80	20	0	0	0
5	100	0	0	0	0
6	100	0	0	0	0
7	100	0	0	0	0
Question	Very easy	Easy	Adequate	Hard	Very hard
8	60	40	0	0	0
9	80	20	0	0	0
10	80	0	20	0	0

rotating equipment failure''. The main difficulties faced by the team were the possible unavailability of electric power at the point of installation and the impracticality of using intrusive sensors.

There were seven members in the *i-Machine* team, all experienced professionals who had previously worked with IoT. There was a data scientist, a technician, and electrical, telecommunications and mechanical engineers. The roles of project manager and a development manager were taken by the senior members. The remaining members worked as the development team and consultants. The project's main expected results were:

- Increased availability and operational efficiency of electrical rotating machines;
- Early detection of failures and reduced maintenance costs;
- Reduction of accidents with personnel.

The project was split into three main attributes, namely:

(1) Continuous vibration and temperature sensing.
(2) Energy self sufficiency (environmental energy harvesting).
(3) Wireless communication capability.

During Phase 1, the team produced a TpM-IoT-Canvas where business, solution description, justifications, objectives, and expected benefits were laid down, as shown in Fig. 7.

This information allowed the preparation of the Business Report.

On Phase 2, the main requirements documented in the Requirements Report were:

- Graphical interfaces to show the evolution of equipment wear and tear;
- Intelligent and accurate data processing;
- Local and cloud storage;
- Low energy consumption;
- Wireless connectivity capability;
- Level of precision and accuracy in sensor data;
- Devices small size and low mass;
- Low cost.

Phase 3 concludes with the writing of the Implementation Report, which details the implementation and technological choices. The actual implementation details and technologies used are not presented in this paper due to confidentiality issues. However, this

fact does not influence the results of this work as the objective is to evaluate the use of the methodology. The *i-Machine* project started in November 2020 and has just cleared the prototyping stage. The project is set to finish by the end of October 2022.

The team's perception regarding the TpM-Pro was assessed once Phase 3 was over.

Following the guidelines set down by the Association for Computing Machinery, we used methodological triangulation, i.e., we combined qualitative (interview and observation) and quantitative (survey) methods to increase the precision of this empirical research.

First, each member was interviewed. There were no questions as such but an open conversation that gave them the opportunity to express their opinion about the methodology.

Then a 5-point Likert Scale questionnaire was applied to all members of the team. In these questionnaires the team members could rate specific aspects of the methodology. It included the following statements and questions:

(1) The TpM-Pro was easy to understand and assimilate.
(2) The TpM-Pro helped me to better understand what IoT is.
(3) The TpM-Pro helped the team to better understand the project.
(4) The TpM-Pro helped integrate the team members.
(5) The TpM-Pro contributed to increasing the team's productivity.
(6) The TpM-Pro provided a better quality of service (technical competence, reliability, delivery time).
(7) The iterative and incremental nature of the TpM-Pro provided the necessary flexibility for the development of the project.
(8) What was the level of difficulty applying Phase 1 of TpM-Pro?
(9) What was the level of difficulty applying Phase 2 of TpM-Pro?
(10) What was the level of difficulty applying Phase 3 of TpM-Pro?

Approval of the TpM-Pro as a development methodology was attested by the answers given by most interviewees. During the conversations, it was a recurrent observation that the TPM-Pro provided a clear path to be followed, giving them the confidence that no short cuts were being taken, albeit inadvertently. Some respondents mentioned that the TpM-Pro's flexibility and manufacturer and technology independence were its main contributions and that the TpM-Pro served them well as a project guide. There was however, one interviewee that expressed a certain resistance to use the methodology indicating that it was difficult to follow.

The survey results also indicate that the methodology was well accepted by the *i-Machine* team, as presented in Table 2.

In the following analysis of the data resulting from the survey, we refer to the ratings Agree and Somewhat Agree as positive ratings.

The results pointed out that the general perception was positive. The same rates were assigned to statements 1 and 2, indicating that the TpM-Pro was helpful and easy to comprehend. Respondents also wrote that the TpM-Pro allowed for a better work organization.

There was a consistency of positive ratings for statements 3, 4, 5, and 6 concerning the benefits of the methodology with regard to cohesion, productivity, and QoS, corroborating our belief that the TpM-Pro delivers a better-integrated team. All respondents agreed with statement 7, that the TpM-Pro granted flexibility to the development process.

The answers to questions 8 to 10, about the level of difficulty in handling the phases, told us that the users considered this the hardest aspect to grasp. Probably this was caused by the novelty of the methodology and shortage of training time.

In addition, our perception regarding the team's take on the TpM-Pro was that it indeed helped them to see the project as a whole, bridging the business necessity to the solution developed.

5. Conclusion

Developing IoT systems is a process more complex than traditional software development, as these systems often encompass software, hardware, and communication components. Validation of development methodologies that seek to attend IoT systems needs are also an important topic that needs to be addressed.

Of the six methodologies presented in this article, only Ignite and the TpM-Pro have been validated in real-world projects. Although the original TpM has been used by over 100 students in extension and postgraduate IoT courses, the TpM-Pro had not yet been tested for projects in a corporate environment. Its adoption by the *i-Machine* team allowed us to test its applicability in real-world projects. This validation has allowed us to assess and improve the TpM-Pro.

None of the methodologies identified in this paper can be considered a complete blueprint that contemplates everything in the development process of IoT solutions. However, we feel that, out of these methodologies, the TpM-Pro could be considered the more holistic one, as it presents the clearest way to develop IoT solutions, starting at the business, all the way to a running solution, while keeping full project documentation (activities, artifacts, roles, requirements, technologies and testing). The TpM-Pro presents a way of organizing the flow of decisions made during all stages of an IoT project.

The development of the TpM-IoT-Canvas within the TpM-Pro domain granted the methodology an unique way to easily extract information about the business and expected solution, while, at the same time, encouraging the team to get involved in the project planning process.

The TpM-Pro is devoid of any commercial bias, focusing foremost on the client's needs. The methodology proved to be a useful instrument, especially for teams with little experience in the IoT area. It is not necessary to learn new techniques or specific tools, as the TpM-Pro offers the possibility of using all the team's previous knowledge, guiding it through a common and integrated path.

The feedback received from the *i-Machine* team suggests that the TpM-Pro is a promising development methodology for IoT. Nevertheless, we recognize that it is a preliminary assessment. Further testing should bring new contributions, consolidating and validating the methodology. However, this will happen only when developers consider the TpM-Pro as an option and make use of it on their IoT projects.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

References

[1] M. Lombardi, F. Pascale, D. Santaniello, Internet of things: A general overview between architectures, protocols and applications, Information 12 (2) (2021) 87.
[2] E. Tunstel, M.J. Cobo, E. Herrera-Viedma, I.J. Rudas, D. Filev, L. Trajkovic, C.L.P. Chen, W. Pedrycz, M.H. Smith, R. Kozma, Systems science and engineering research in the context of systems, man, and cybernetics: Recollection, trends, and future directions, IEEE Trans. Syst., Man, Cybern.: Syst. 51 (1) (2021) 5–21.
[3] P. Wegner, Global IoT market size grew 22 in 2021 — these 16 factors affect the growth trajectory to 2027, 2022, URL https://iot-analytics.com/iot-market-size/. (Accessed 31 March 2022).
[4] R.D. Woolley, Why IoT projects fail, 2020, URL https://www.whyiotprojectsfail.com/ (Accessed 31 April 2022).
[5] S. Jha, 8 out of 10 IoT projects fail even before they are launched, 2016, URL https://cio.economictimes.indiatimes.com/news/internet-of-things/8-out-of-10-iot-projects-fail-even-before-they-are-launched/524488. (Accessed 31 March 2022).
[6] V.S. Chakravarthi, Internet of Things and M2M Communication Technologies, Springer, 2021.
[7] J.P. Dias, A. Restivo, H.S. Ferreira, Designing and constructing internet-of-things systems: An overview of the ecosystem, Internet Things 19 (2022) 100529, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660522000312.
[8] J. Jin, J. Gubbi, S. Marusic, M. Palaniswami, An information framework for creating a smart city through internet of things, IEEE Internet Things J. 1 (2) (2014) 112–121.
[9] S. Hayward, K. van Lopik, A. West, A holistic approach to health and safety monitoring: Framework and technology perspective, Internet Things 20 (2022) 100606, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660522000889.
[10] G. Mei, N. Xu, J. Qin, B. Wang, P. Qi, A survey of internet of things (IoT) for geohazard prevention: Applications, technologies, and challenges, IEEE Internet Things J. 7 (5) (2019) 4371–4386.
[11] L.C.B.C. Ferreira, O.C. Branquinho, P.R. Chaves, P. Cardieri, F. Fruett, M.D. Yacoub, A PBL-based methodology for IoT teaching, IEEE Commun. Mag. 57 (11) (2019) 20–26.
[12] A.L.B. Déo, Proposal of an Open Source Reference Model for IoT (Master's thesis), Pontifical Catholic University of Campinas, Brazil, 2018.
[13] P.R. Chaves, R.M. Assumpção, L.C. Ferreira, P. Cardieri, O.C. Branquinho, F. Fruett, A remote emulation environment for the teaching of low-power wireless communications, Comput. Appl. Eng. Educ. (2021).
[14] L.C.B. Ferreira, R. Yamaguti, O.C. Branquinho, P. Cardieri, A tpm-based collaborative system to teach IoT, Comput. Appl. Eng. Educ. (2021).
[15] W. Edwards, F.H. Barron, SMARTS and SMARTER: Improved simple methods for multiattribute utility measurement, Organ. Behav. Hum. Decis. Process. 60 (3) (1994) 306–325.
[16] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac, Internet of things: Vision, applications and research challenges, Ad Hoc Networks 10 (7) (2012) 1497–1516.
[17] Y. Wang, M. Hou, K.N. Plataniotis, S. Kwong, H. Leung, E. Tunstel, I.J. Rudas, L. Trajkovic, Towards a theoretical framework of autonomous systems underpinned by intelligence and systems sciences, IEEE/CAA J. Autom. Sin. 8 (1) (2020) 52–63.
[18] G. Fortino, C. Savaglio, G. Spezzano, M. Zhou, Internet of things as system of systems: A review of methodologies, frameworks, platforms, and tools, IEEE Trans. Syst., Man, Cybern.: Syst. (2020).
[19] M. Noura, M. Atiquzzaman, M. Gaedke, Interoperability in internet of things: Taxonomies and open challenges, Mob. Netw. Appl. 24 (3) (2019) 796–809.
[20] J. Albus, P. Antsaklis, Panel discussion: Autonomy in engineering systems: What is it and why is it important? Setting the stage: Some autonomous thoughts on autonomy, in: Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) Held Jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell, IEEE, 1998, pp. 520–521.
[21] S. Alter, Making sense of smartness in the context of smart devices and smart systems, Inf. Syst. Front. 22 (2) (2020) 381–393.
[22] A. Gupta, R. Christie, P. Manjula, Scalability in internet of things: Features, techniques and research challenges, Int. J. Comput. Intell. Res. 13 (7) (2017) 1617–1627.
[23] R. Pressman, B. Maxim, Engenharia de Software-8ª Edição, McGraw Hill Brasil, 2016.
[24] S. Saeed, N. Jhanjhi, M. Naqvi, M. Humayun, Analysis of software development methodologies, Int. J. Comput. Digit. Syst. 8 (5) (2019) 446–460.
[25] G. Fortino, W. Russo, C. Savaglio, W. Shen, M. Zhou, Agent-oriented cooperative smart objects: From IoT system design to implementation, IEEE Trans. Syst., Man, Cybern.: Syst. 48 (11) (2017) 1939–1956.
[26] X. Song, L.J. Osterweil, Experience with an approach to comparing software design methodologies, IEEE Trans. Softw. Eng. 20 (5) (1994) 364–384.
[27] I. Lee, K. Lee, The internet of things (IoT): Applications, investments, and challenges for enterprises, Bus. Horizons 58 (4) (2015) 431–440.
[28] S. Greengard, The Internet of Things, MIT Press, 2021.

[29] A. Vallecillo, et al., RM-ODP: The ISO reference model for open distributed processing, DINTEL Ed. Software Eng. 3 (2001) 66–69.
[30] R. Casadei, C. Tsigkanos, M. Viroli, S. Dustdar, Engineering resilient collaborative edge-enabled IoT, in: 2019 IEEE International Conference on Services Computing, SCC, IEEE, 2019, pp. 36–45.
[31] I. Ayala, M. Amor, L. Fuentes, J.M. Troya, A software product line process to develop agents for the iot, Sensors 15 (7) (2015) 15640–15660.
[32] P. Patel, D. Cassou, Enabling high-level application development for the internet of things, J. Syst. Softw. 103 (2015) 62–84.
[33] G. Fortino, W. Russo, ELDAMeth: An agent-oriented methodology for simulation-based prototyping of distributed agent systems, Inf. Softw. Technol. 54 (6) (2012) 608–624.
[34] D. Slama, F. Puhlmann, J. Morrish, R.M. Bhatnagar, 9Enterprise IoT: Strategies and Best Practices for Connected Products and Services, " O'Reilly Media, Inc.", 2015.
[35] T. Bucher, M. Klesse, S. Kurpjuweit, R. Winter, Situational method engineering, in: Working Conference on Method Engineering, Springer, 2007, pp. 33–48.
[36] S. Alter, Work systems and IT artifacts: Does the definition matter?, 2006.
[37] A.F. Harmsen, M. Ernst, U. Twente, Situational Method Engineering, Citeseer, 1997.
[38] K. Kumar, R.J. Welke, Methodology engineeringr: A proposal for situation-specific methodology construction, in: Challenges and Strategies for Research in Systems Development, 1992, pp. 257–269.
[39] K.v. Slooten, B. Hodes, Characterizing IS development projects, in: Working Conference on Method Engineering, Springer, 1996, pp. 29–44.
[40] U. Baumoel, Strategic agility through situational method construction, in: Proceedings of the European Academy of Management Annual Conference, vol. 2005, Munich, 2005.
[41] S. Brinkkemper, M. Saeki, F. Harmsen, Assembly techniques for method engineering, in: International Conference on Advanced Information Systems Engineering, Springer, 1998, pp. 381–400.
[42] J. Vlietland, H. van Vliet, Towards a governance framework for chains of scrum teams, Inf. Softw. Technol. 57 (2015) 52–65.
[43] J. Finocchio Júnior, Project model canvas: Gerenciamento de projetos sem burocracia, São Paulo (2013).
[44] R.L. Kelly, H.R.1668 - IoT cybersecurity improvement act of 2020, 2019, https://www.congress.gov/bill/116th-congress/house-bill/1668. (Accessed 12 September 2022).
[45] NIST, NIST cybersecurity for IoT program, 2022, https://www.nist.gov/itl/applied-cybersecurity/nist-cybersecurity-iot-program. (Accessed 12 September 2022).
[46] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empir. Softw. Eng. 14 (2) (2009) 131–164.