Internet of Things 34 (2025) 101805

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Software engineering in IoT: Insights from a survey of 361 experts

Maura Cerioli, Maurizio Leotta *, Gianna Reggio

Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS), University of Genova, Italy

ARTICLE INFO

Keywords:
Internet of Things
Personal opinion survey
Software engineering
Empirical study
Researchers
Practitioners

ABSTRACT

The Internet of Things (IoT) has rapidly integrated into almost every aspect of daily life. The exponential growth in connected devices has established IoT as a critical component in various domains, including industrial automation, building management, healthcare, transportation, and logistics. These complex systems combine digital and physical elements distributed across networks from the Edge to the Cloud, supporting real-time decision-making and automation. Software plays a fundamental role in IoT systems, underpinning every layer from sensor firmware to cloud infrastructure, and is essential for the design, development, and maintenance of robust and scalable IoT solutions. Effective Software Engineering (SE) practices are vital to meeting these systems' diverse and evolving demands. This study aims to gain a detailed understanding of IoT system development, focusing on identifying prevalent software technologies, adopted SE practices, and IoT-specific approaches. We also seek to explore the challenges and needs faced by SE experts in this field. To achieve this, we conducted a survey gathering insights directly from 361 IoT practitioners and experts across 53 countries. This diverse participation provided a comprehensive view of IoT systems, offering valuable insights into current practices and future trends in the industry.

1. Introduction

The Internet of Things (IoT) technology has become an integral part of nearly every facet of our daily lives, seamlessly embedding itself into numerous types of systems. The rapid proliferation of IoT systems has profoundly transformed how we live, work, and interact, impacting billions of people globally in recent years. The number of connected devices has surged exponentially (e.g., [1] reports 18.8 billion of IoT devices connect at the end of 2024 and forecasts 30 billion by 2030), with IoT applications now playing a critical role in diverse domains such as industrial automation, building management, healthcare, transportation, and logistics, e.g., almost 50 % of the firms in the 38 EOCD[1] (Organization for Economic Co-operation and Development) countries use at least two IoT devices or systems [2]).

IoT systems are inherently complex, consisting of both digital and physical components, including a vast array of interconnected objects and devices distributed across the network, spanning from the Edge to the Cloud. These systems not only facilitate communication and data exchange but also enable real-time decision-making and automation, driving efficiency and innovation in many sectors.

In this complex landscape, software serves as the cornerstone, permeating every layer of IoT systems. From the firmware embedded in small sensors to the sophisticated code running on cloud infrastructures, Software Engineering (SE) is crucial for the design,

* Corresponding author.
E-mail address: maurizio.leotta@unige.it (M. Leotta).

[1] https://www.oecd.org/en.html

https://doi.org/10.1016/j.iot.2025.101805
Received 8 April 2025; Received in revised form 8 October 2025; Accepted 16 October 2025
Available online 20 October 2025

development, deployment, and maintenance of high-quality IoT solutions. Effective SE practices ensure that these systems are robust, scalable, and capable of meeting the diverse and evolving demands of their respective domains.

We already published the results of a preliminary survey [3] aimed to understand the nature of IoT systems and the vital aspects of their development from a practical point of view, with a particular focus on identifying (a) the elements and characteristics that represent first-class entities of IoT systems in practice, and (b) the challenging aspects of high-quality IoT systems in practice. Specifically, in that preliminary survey we addressed the following four research questions through a questionnaire consisting of eight questions (excluding those aimed at collecting personal and company information): (1) what are the elements and characteristics considered vital parts of IoT systems? (2) what are the most popular IoT system types? (3) how is deployment managed for IoT systems? (4) what is the perceived relevance of quality attributes for IoT systems, and how difficult are they to achieve?

While the prior work identified key elements and quality attributes of IoT systems, this paper aims to provide a detailed understanding of the technologies, processes, and challenges that define modern IoT development.

More in detail, the study aims to investigate current practices in IoT system development through four main research questions, focusing on: (1) the most commonly used software technologies (e.g., protocols, languages, IDEs, platforms, and tools); (2) the approaches and processes adopted for IoT development; (3) the specific challenges and needs compared to other types of systems; and (4) how developers acquire knowledge about IoT development. To address these questions, we conducted a large-scale survey targeting IoT practitioners and experts to collect relevant and practical insights. We used an online questionnaire to collect data, inviting a broad spectrum of IoT experts to participate. Overall, our survey was administered to 361 IoT practitioners and experts across 53 countries.

The analysis of the collected answers reveals key trends in IoT development and a few nuances in how they are perceived by different roles in development. This paper contributes a data-driven overview of the state of software engineering in the IoT domain, with the goal of benefiting industrial practitioners, researchers, and educators alike.

The remainder of the paper is structured as follows. Section 2 describes the design and the procedure followed in conducting the survey. Section 3 provides an overview of the characteristics of the participants. Section 4 reports the results and discusses them, also analyzing the influence of the respondents' role (Management, Development, Research). Related work is described in Section 5, followed by conclusions and future works in Section 6.

2. Study goal and design

The present study undertook the execution of a survey employing an online questionnaire to investigate the practical application of software engineering practices in the realm of IoT system development. We based our study design on the principles described by Kitchenham and Pfleeger [4] as we did, for example, in Reggio et al. [5]. We started with the goal definition and then derived our research questions. After identifying the population, we designed and administered our questionnaire accordingly. The collected raw data can be found at https://sepl.dibris.unige.it/2025-IoT-SE-Survey.php.

2.1. Goal

Our primary interest is understanding the peculiarities of IoT System development in practice, its current trends, and the differences, if any, in software engineering practice application compared to the development of more traditional systems. Thus, the goal (Caldiera and H. D. Rombach [6]) of our survey is:

Goal *to understand the current status of IoT system development, the usage of software technologies and software engineering practices, IoT-specific approaches, challenges, and needs, if any*.

We searched the literature for available empirical studies on similar subjects, and to the best of our knowledge, this paper is the first to address the aforementioned specific goal.

The knowledge we are seeking has the potential to significantly benefit industrial practitioners, researchers, and professional educators in the field of IoT system development and software engineering. It will help *practitioners* to select the best SE practices and technologies for the IoT, *researchers* to identify and tackle interesting challenges, and *professional educators* to choose the most relevant topics and the most effective communication channels to reach their target, especially in the setting of life-long learning.

2.2. Research questions

We refined our goal into four main research questions.

RQ1 What software technologies (protocols, languages, IDEs, platforms, services, and code automation tools) are currently most used to develop IoT systems?

RQ2 What approaches are used nowadays to develop IoT systems concerning development processes and selected specific software engineering tasks?

RQ3 What are the challenges and needs related to developing IoT systems? Are they different from those for the development of other kinds of systems?

RQ4 How do people get their knowledge about IoT systems development?

Section 4 will detail the questions we included in our questionnaire to answer our research questions.

2.3. Target population

Our study target population mainly consists of practitioners working on IoT systems and researchers studying possible improvements for their development techniques. To reach a sizable sample (i.e., in the order of several hundred participants), following the principles described in [7,8], we (a) posted the survey on IoT-related groups on social networks for professionals like, for instance, LinkedIn, and on forums of well established IoT conferences to reach other experts. Moreover, we also applied (b) *Convenience sampling* by collecting contacts of experts in IoT system engineering from our industrial partners and personally asking the suggested practitioners[2] to participate in our survey. Then, we made our sample grow by adopting Snowball sampling by asking those practitioners for help distributing the questionnaire through their professional networks. In this way, most of the experts who participated in our survey were outside of our professional network. Thus, we avoided possible biases due to a closely knitted community of respondents.

We started our questionnaire by asking if the respondents had experience in IoT development and proposed the other questions only to those who had worked on some IoT systems. Thus, all our participants have hands-on experience with IoT projects, and their answers reflect that experience.

2.4. Questionnaire design

The questionnaire was developed through a multi-step process. First, we conducted a review of relevant (scientific and gray) literature in the IoT domain to identify key topics and feasible alternatives for answers. Based on this, we drafted an initial version of the questionnaire. This draft was then shared with a group of domain experts who reviewed the questions and provided detailed feedback, including suggestions for additional response options and clarifications. Their input was collected through annotated drafts, which allowed us to iteratively refine and expand the predefined answer lists. Finally, the revised questionnaire was tested in a pilot study with representative participants, whose feedback contributed to further improvements in question clarity and completeness.

More specifically, the *pilot study* was conducted to ensure that our questions did not contain ambiguities, to verify that the time required to complete the form was appropriate for an unsupervised online survey, and to fine-tune the questionnaire. Three Italian and one Swedish academics, plus two Swedish professionals, answered the initial version of the questionnaire (and did not participate in the final survey). They responded in under 10 minutes and were interviewed immediately afterward to gather their comments and suggestions that concerned mainly the question wording and, in much minor measure, missing items among the possible answers. We used those data to identify ambiguous terms and clarify our questions accordingly, and to complete our lists of possible predefined answers. After the minor changes recommended by the pilot participants and other experts contacted, our final questionnaire contains three parts.

Introduction: setting the context of the questionnaire itself, its goal, the expected participant background, and the average time required to complete the survey.

Core: asking for the technical answers needed to investigate our research questions. Table 1 presents such questions, including the available choices for closed-answer questions and the number of respondents. They are also grouped based on their contribution to each research question.

Personal and company info: collecting information about the participants, such as, for instance, their country, working position, and seniority, and their current employing company, if any, such as, for example, the company's core business, size, and years of presence on the market. Table 2 presents such questions, including again the available choices for closed-answer questions and the number of respondents. Section 3 will aggregate the answers to these questions to sketch our survey participant overview. Moreover, we will extensively use information about the role to cluster the received answers and see when those elements influence the participant's perceptions.

2.5. Data collection and analyses

As described above, we proposed an online questionnaire to collect responses, reaching a worldwide population sample in this way. Using the results of the pilot experiment conducted in September 2018, the survey opened in October 2018, and most answers were received within the first 12 months. Using a Web-based approach was instrumental in simplifying the question collection and getting a satisfying number of responses, as also suggested in the literature ([9]).

The responses to the questionnaire collectively capture the frequency and impact of some elements in our target population. Thus, this survey's nature is mainly descriptive and lends itself to descriptive statistics.

[2] Around 30 people.

Table 1
Questionnaire - Core section.

	Section on technologies used for IoT development
Q1	Within the IoT projects that you worked on, which languages/protocols were used? Please select all answers that suit you. *Answer type*: Multiple choice including 10 languages and 9 protocols + open ended 359 respondents (172 Development, 121 Management, 63 Research, 3 role unknown)
Q2	Within the IoT projects that you worked on, which Cloud Platforms were used, and how satisfactory were them? *Answer type*: For each of the five proposed platforms (1) *Azure Sphere (Microsoft Azure IoT Platform)*, (2) *AWS IoT Core (Amazon Web Services Platform)*, (3) *Bluemix (IBM Cloud)*, (4) *In house platform*, and (5) *Other* a single choice among [*Not used*, *Not satisfactory*, *Scarcely satisfactory*, *Satisfactory*, *Highly satisfactory*]; moreover, the possibility of detailing the last choice was offered 350 respondents (170 Development, 120 Management, 58 Research, 2 role unknown)
Q3	If you selected other in the previous question, which were the other used Cloud Platforms? *Answer type*: open ended 99 respondents (51 Development, 36 Management, 12 Research)
Q4	Within the IoT projects that you worked on, which IDEs were used? Please specify. *Answer type*: open ended 216 respondents (122 Development, 53 Management, 41 Research)
Q5	Within the IoT projects that you worked on, have services been used? *Answer type*: Single choice among [*No*, *Yes (e.g., RESTful services)*, *Other*] with the possibility of detailing the last choice 345 respondents (168 Development, 112 Management, 63 Research, 2 role unknown)
Q6	Within the IoT projects that you worked on, were tools for the automatic generation of code used? *Answer type*: boolean choice 340 respondents (165 Development, 112 Management, 61 Research, 2 role unknown)
Q7	If you answered yes, which one have you used? *Answer type*: open ended 51 respondents (28 Development, 15 Management, 8 Research)
	Section on SE methods and tasks for IoT development
Q8	Within the IoT projects that you worked on, which development process was followed? *Answer type*: Multiple choice among [*Agile/Scrum*, *Iterative*, *Waterfall*, *Other*] with the possibility of detailing the last choice 351 respondents (168 Development, 120 Management, 61 Research, 2 role unknown)
Q9	Within the IoT projects that you worked on, how were the needs/ requirements documented, and how were the used approaches satisfactory? *Answer type*: For each of the five proposed documentation styles (1) *Free-form natural language/graphical documents*, (2) *Natural language/graphical documents following specific formats*, (3) *Use cases*, (4) *Goals*, and (5) *Subsumed by other documents, e.g. contract with the client, vision* a single choice among [*Not used*, *Not satisfactory*, *Scarcely satisfactory*, *Satisfactory*, *Highly satisfactory*] 353 respondents (168 Development, 122 Management, 60 Research, 3 role unknown)
Q10	Within the IoT projects that you worked on, how was the design documented, and how satisfactory were the used approaches? *Answer type*: For each of the five proposed documentation styles (1) *Free-form natural language/graphical documents*, (2) *Natural language/graphical documents following specific formats*, and (3) *Models produced using a modelling notation (e.g., UML Class Diagram)* a single choice among [*Not used*, *Not satisfactory*, *Scarcely satisfactory*, *Satisfactory*, *Highly satisfactory*] 348 respondents (165 Development, 121 Management, 60 Research, 2 role unknown)
Q11	In case you have used models to document the design, which have you used? *Answer type*: open ended 64 respondents (31 Development, 15 Management, 18 Research)
Q12	Within the IoT projects that you worked on, which kinds of testing were performed and how satisfactory were them? *Answer type*: Single choice among [*Not performed*, *Not satisfactory*, *Scarcely satisfactory*, *Satisfactory*, *Highly satisfactory*] for each of the three testing activities and an extra open ended possibility 339 respondents (164 Development, 115 Management, 58 Research, 2 role unknown)
Q13	If you answered Other, please specify which. *Answer type*: open ended 42 respondents (22 Development, 17 Management, 3 Research)
	Section on specificities of IoT development
Q14	In your experience, is developing IoT systems different from non-IoT systems? *Answer type*: boolean choice 361 respondents (173 Development, 122 Management, 63 Research, 3 role unknown)
Q15	If you answered yes, please say which are the differences. *Answer type*: open ended 203 respondents (91 Development, 73 Management, 39 Research)
Q16	While developing an IoT system, do you feel that there are lacks and/or challenges within the following areas? Please, select all answers that suit you. *Answer type*: Multiple choice among [*Languages*, *Frameworks*, *Tools*, *Development processes*, *IDEs*, *Requirement capture and specification*, *Design*, *Testing*, *Other*] with the possibility of detailing the last choice 311 respondents (150 Development, 103 Management, 57 Research, 1 role unknown)
	Section on knowledge sources for IoT development
Q17	How did you get your knowledge about IoT? *Answer type*: Multiple choice among [*Reading books*, *Reading (web/printed) material provided by, e.g. companies producing IoT devices/software, blogs, tech survey sites*, *Attending IoT courses*, *Scientific literature*, *Other*] with the possibility of detailing the last choice 359 respondents (172 Development, 122 Management, 63 Research, 2 role unknown)

Table 2
Questionnaire - Section on personal and company info.

ID	Question
Q18	Which is your field of education? *Answer type*: Single choice among [*Computer Science*, *Telecommunication Engineering*, *Electronic Engineering*, *Physics*, *Other*] with the possibility of detailing the last choice 357 respondents (172 Development, 121 Management, 63 Research, 1 role unknown)
Q19	Which of the following would better describe your current roles/activities? *Answer type*: Single choice among [*Software Developer*, *Software Architect*, *Project Manager*, *Firmware Architect*, *Researcher*, *System Engineer*, *Other*] with the possibility of detailing the last choice 358 respondents (173 Development, 122 Management, 63 Research, 0 role unknown)
Q20	In which country are you currently working? *Answer type*: open ended 336 respondents (157 Development, 117 Management, 60 Research, 2 role unknown)
Q21	Which is the core business of your company? *Answer type*: Single choice among [*Manufacturing*, *Software*, *Transport*, *Telecom*, *Logistics*, *eHealth*, *Other*] with the possibility of detailing the last choice 348 respondents (168 Development, 119 Management, 60 Research, 1 role unknown)
Q22	Which is the size of your company? *Answer type*: Single choice among [*micro (1–9)*, *small (10–49)*, *medium (50–249)*, *large (more than 250)*] 351 respondents (168 Development, 120 Management, 60 Research, 3 role unknown)
Q23	How many years has been your company working in the field of IoT? *Answer type*: Single choice among [*less than 1*, *1–2*, *3–4*, *5–7*, *8–10*, *more than 10*] 353 respondents (169 Development, 120 Management, 61 Research, 3 role unknown)
Q24	How many years have you been working in the field of IoT? *Answer type*: Single choice among [*less than 1*, *1–2*, *3–4*, *5–7*, *8–10*, *more than 10*] 355 respondents (169 Development, 121 Management, 62 Research, 3 role unknown)
Q25	How many IoT projects have you worked in? *Answer type*: Single choice among [*1–3*, *4–6*, *7–10*, *11–15*, *more than 15*] 354 respondents (168 Development, 121 Management, 62 Research, 3 role unknown)

While our study involves a large cohort of participants - much larger than is typical for surveys in this sector (e.g., [5,10–14]) -, we cannot make a formal claim of statistical significance for our results. Indeed, in most cases, the answers are too complex, with several possible options, or are even open-ended, making standard statistical analysis procedures difficult or impossible to apply. Furthermore, the participants, being volunteers, were not selected to be representative of the IoT national communities nor role clusters.

2.5.1. Analysis of open-ended answers

As shown in Table 1, our questionnaire includes six open-ended questions and six further questions allowing an open-ended option. All of them but *Q15* collect lists of conceptual or technological tools. Thus, their analysis is straightforward: splitting each answer in its sub-parts, cleaning typos, and counting the elements in each group. The clustering may require a minimum of domain knowledge; for instance, we needed to identify different text-editors in *Q4*'s answers to group them together. However, given the technical nature of the answers, the process is fully objective and deterministic. The answers to *Q15* are more sophisticated, as represent the perceived differences between IoT and standard development. Also in this case the number of responses is sufficiently limited for the process to be human-manageable so that we were not forced to rely on automatic analysis and deal with its inherent limits. However, the analysis required was less straightforward and we had to devise an iterative, controlled process to ensure that the clustering was not biased by our beliefs. The details are described in Section 4.3, but in a nutshell, after splitting and cleaning the data, two of the authors iteratively refined a tagging system while the third author acted as control.

Our survey did not raise any ethical concern, because the questions only concern technical issues without political, military, or social implications, and we did not coerce nor enticed our participants into answering them. Moreover, the software used for the online survey recorded only the questionnaire answers, and such data do not allow the identification of the respondents. Thus, we are GDPR compliant.

2.6. Threats to validity

Following the classification proposed by Wohlin et al. [15], we discuss the main threats to the validity of our study and the mitigation strategies adopted.

Construct Validity. In our study, construct validity concerns how well the operational instruments (e.g., questionnaire items) and their design are adequate to measure the concepts intended to be studied. To mitigate threats in this area, the questionnaire was developed through a multi-step process starting from an informal review of relevant literature to identify key topics, terminology, and the most common technologies and techniques. As a second step, we designed an initial draft of the questionnaire with predefined answer options, based on gathered information. This draft was then reviewed by several domain experts, who provided annotated feedback with suggestions for clarifications and additional response options. This iterative process helped refine and disambiguate the questions. Furthermore, a pilot study with well-seasoned practitioners and experienced researchers tested the questionnaire to identify

ambiguous terms and missing answer categories. Open-ended questions following predefined lists allowed participants to suggest alternatives, reducing the risk of overlooking relevant aspects. Despite these efforts, we acknowledge that the use of predefined lists may have limited the emergence of entirely new responses and potentially introduced bias. However, the fact that most participants selected items from the predefined lists suggests these captured the choices most relevant to our sample.

Internal Validity. Internal validity refers to the degree to which causal conclusions can be drawn from the study. A threat in this study arises from potential low-quality or insincere responses, especially because participation was unsupervised and online. To mitigate this, we distributed the survey primarily to professionals known to the researchers, either directly or indirectly, and posted it only in selected professional forums. Participants with no IoT experience were excluded from the analysis. Additionally, all questions were optional to discourage random answers, and open-ended questions allowed participants to provide additional insights, improving data quality. We carefully checked for patterns that might indicate careless or automated responses and found no evidence of fake or insincere answers. A possible threat to internal validity concerns unreported misunderstandings of the survey questions. Although participants were explicitly invited to report issues with wording or answer options, some may not have done so. This may have introduced a response bias.

External Validity. External validity concerns the generalizability of the results. Our sampling technique, based on convenience sampling of IoT professionals, may limit the representativeness of the sample and thus the generalizability of findings. Given that the survey was mainly distributed within known professional circles and selected forums, the sample might not reflect the broader IoT developer population. While this constrains external validity, the focus on experienced practitioners ensures that the findings are directly relevant to the target domain.

Conclusion Validity. Conclusion validity relates to the appropriateness of data analysis and the robustness of conclusions. To reduce threats in this category, we designed the questionnaire to collect relevant and unambiguous data and applied analysis methods appropriate to the type and scale of the collected data (details in Section 2.5). The iterative questionnaire design and pilot testing helped minimize measurement errors.

Other Threats. Terminology ambiguity is a specific concern in the IoT domain, where terms may have multiple meanings (e.g., *fog nodes* vs. *edge nodes*). We addressed this by providing clear definitions and clarifications within the questionnaire and verified comprehension through pilot testing.

3. Participants overview

We detail the questions asked to collect data about the participants in Table 2.

In all, 433 individuals responded to the initial question about their experiences. Nonetheless, only 361 affirmed that they had worked on at least one IoT project and were eligible to continue participating. All data discussed from here on refer to those 361 IoT professionals.

Participants come from 53 different countries. To delve into specifics, here are the top ten most frequently represented nations with the respondents: India (78: 11 Research, 22 Management, 44 Development, 1 no answer on role), Italy (78: 17 Research, 31 Management, 30 Development), Germany (23: 6 Research, 6 Management, 11 Development), the United States (20: 2 Research, 7 Management, 11 Development), the United Kingdom (19: 1 Research, 10 Management, 8 Development), Brazil (13: 4 Research, 4 Management, 5 Development), France (13: 9 Research, 3 Management, 1 Development), Sweden (10: 0 Research, 3 Management, 7 Development), Canada (8: 3 Research, 2 Management, 3 Development), and Spain (7: 0 Research, 4 Management, 3 Development)[3]. Since none of the questionnaire questions were mandatory, a few respondents (25) chose to keep their country of employment private.

We categorized the participants' working roles into three distinct categories. – *Development* (total 173), including software architect (51), software developer (50), system engineer (35), firmware architect (8), and consultant (8); – *Management* (total 122), including project manager (68), manager (19), product manager (11), and sales (15); and – *Research* (total 63), that is, researcher (58) and university professor (5). Only three respondents did not provide information about their working role.

The majority of our participants have a strong foundation in technical fields, with the education field of the majority being computer science (163). This was followed by electronic and telecommunication engineering (90 and 57, respectively) and physics (13), demonstrating the technical expertise of our sample.

Regarding the participants' years of experience in the IoT field, it is evident from the chart on the left in Fig. 1 that the majority of them have 3–4 years of experience (31 %), followed by 1–2 and 5–7 years (26 % and 16 %, respectively). The percentages refer to the answers received, which were 355 out of 361 respondents.

Turning our attention to the count of IoT projects in which they were involved, the chart on the right in Fig. 1 illustrates that most participants worked on 1–3 and 4–6 IoT projects (37 % and 24 %, respectively). Several participants (18 %) gained their experience in IoT by working on more than 15 projects. The percentages refer to the answers received, which were 354 out of 361 respondents.

When we examined the characteristics of the companies where the survey participants work, we found that more than half are classified as large (52 %). In contrast, the medium, small, and micro are 12 %, 19 %, and 17 %, respectively (see Fig. 2, left chart).

[3] The following countries received less than 5 votes: Albania, Argentina, Australia, Austria, Bangladesh, Belgium, China, Croatia, Denmark, Egypt, Greece, Iran, Ireland, Israel, Japan, Luxembourg, Malaysia, Mexico, Morocco, Netherlands, Norway, Pakistan, Poland, Portugal, Qatar, Romania, Russia, Saudi Arabia, Serbia, Singapore, Slovenia, South Africa, Sudan, Switzerland, Tunisia, Turkey, Ukraine, United Arab Emirates, Uzbekistan, Venezuela, Vietnam.

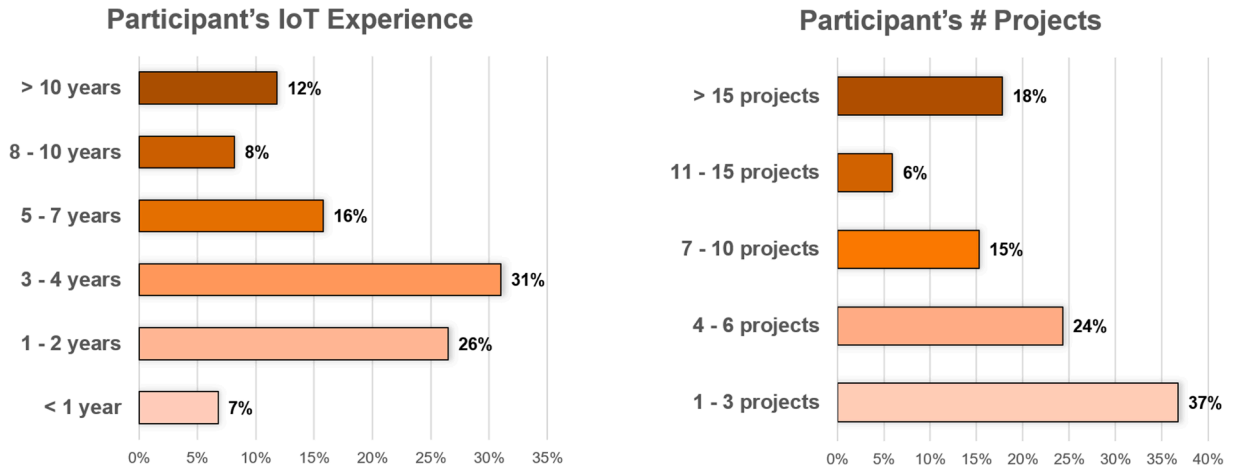

Fig. 1. Participant's experience overview.

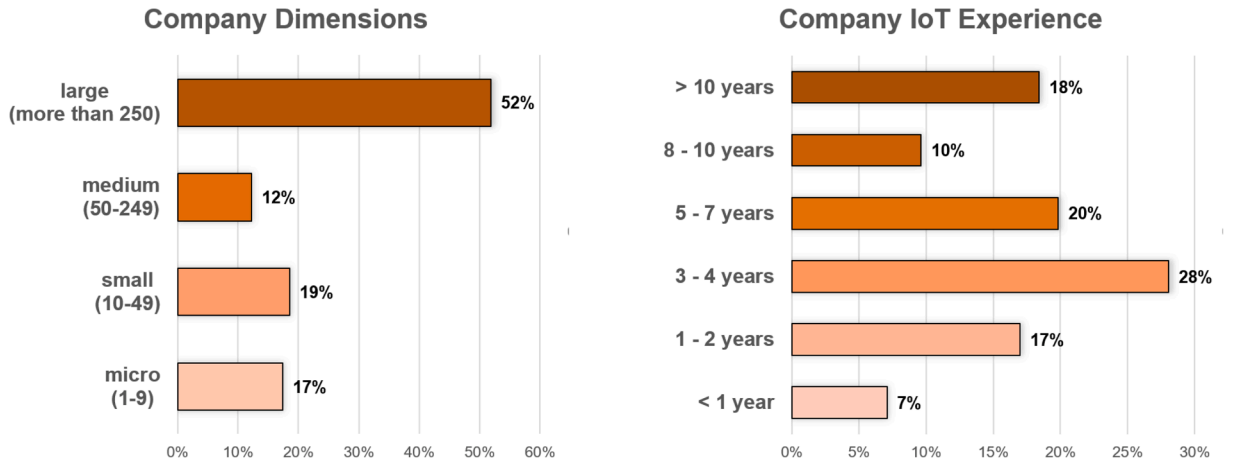

Fig. 2. Participant's company overview.

The second chart in Fig. 2 reveals the duration of company involvement in the IoT sector. The majority have 3–4 years of experience (28 %), followed by 5–7 and more than 10 years (20 % and 18 %, respectively). The percentages refer to the answers received, which were 353 out of 361 respondents.

Concerning the companies' core business, the most popular answers were Software (40 %), Telecom (14 %), Manufacturing (13 %), and Education & Research (11 %); the other alternatives were selected by less than 10 % of the participants. Interestingly, the responses that specified a particular domain, such as eHealth, Energy, or Automotive, received fewer than ten responses each. These modest figures align with the fact that more than half of the companies are categorized as large, suggesting that they likely have a broader, generic core business, such as software or manufacturing.

In summary, the survey participants have considerable experience in IoT, making them capable of offering diverse perspectives and valuable insights into the current state of IoT practices. They constitute a satisfactory large and heterogeneous sample for our survey.

4. Results

We devote each of the following subsections to one of our four research questions, presenting and analyzing the collected answers to the pertinent questions, and discussing the insight we get into the matter. The reader can refer to Table 1 for the list of survey questions and their possible answers, grouped into subsections dedicated to each research question.

We cluster the answers to each specific survey question by the respondents' roles and present the results for the whole sample and the clusters. We discuss only the most notable differences among the clusters and propose some interpretations of the variance. As reported in the previous section, we provide all the data online to allow the interested reader to investigate minor discrepancies.

Because, by design, any question could be left unanswered, the respondent number may slightly differ from question to question. Therefore, the raw answer numbers may be misleading. Hence, instead of discussing the choices for a specific option, we focus on

Protocol	ALL	Development	Management	Research
MQTT	68.0%	76.2%	66.9%	47.6%
WiFi	65.2%	64.0%	68.6%	63.5%
Bluetooth	44.8%	41.9%	49.6%	42.9%
Bluetooth Low Energy (BLE)	42.1%	37.2%	50.4%	36.5%
Zigbee	31.8%	31.4%	33.1%	28.6%
Near Field Communication (NFC)	24.2%	20.3%	35.5%	12.7%
CoAP	20.9%	23.3%	17.4%	20.6%
6LowPAN	18.7%	18.0%	20.7%	15.9%
Z-Wave	9.5%	8.1%	11.6%	9.5%

Fig. 3. Protocols: proposed answers - distribution of choices.

Language	ALL	Development	Management	Research
Python	61.0%	62.8%	60.3%	58.7%
Javascript	59.6%	63.4%	58.7%	50.8%
Java	44.8%	39.5%	48.8%	52.4%
C	44.8%	52.9%	37.2%	38.1%
C++	34.5%	32.6%	39.7%	30.2%
Node-Red	25.9%	25.0%	29.8%	19.0%
PHP	18.1%	16.9%	15.7%	25.4%
Go	7.5%	12.2%	3.3%	3.2%
Swift	6.1%	5.2%	8.3%	4.8%
Rust	1.7%	0.6%	2.5%	3.2%

Fig. 4. Languages: proposed answers - distribution of choices.

the percentage that those choices represent of the respondents (having a particular role) who elected to answer the corresponding question. Thus, the results are comparable across clusters of different sizes, such as the subsets of the sample in some role, or the respondents to different questions.

All questions had a substantial number of answers. Of the 361 responses given by IoT professionals, the number of answers to questions proposing a list of possible options ranged from 203 (*Q15*), which is 56.2 %, to 361 (*Q14*), which is 100 %. We had lower numbers (but still adequate) for the open-ended questions, whose answers ranged from 42 (*Q13*), that is, 11.6 %, to 222 (*Q4*), that is, 61.5 %. Given that most open-ended questions appeared in the questionnaire to let participants suggest missing options, the few answers indicate that most of the respondents have been satisfied with our selection of possible choices.

4.1. *Software technologies for IoT systems*

To answer our first research question, we use the questions presented in the first section of Table 1.

Protocols and languages

Question *Q1* investigates the most used protocols and languages for IoT development. Figs. 3 and 4 present the predefined choices for protocols and languages, respectively.

Besides the listed items, an open-ended choice was available, whose results we will discuss later. The second column contains the percentages of choices disregarding the participant role, and the following ones are the percentages for each role. We used a heat map to highlight the interest arisen by the different protocols and languages. For each category, we list the items from top to bottom in decreasing order of interest by the overall population.

Among the protocols, the most used is MQTT[4] (68.0 %), an open-source, application layer protocol based on the publish-subscribe pattern that is extremely popular for connecting devices. Our result agrees with the widespread appreciation given to MQTT on the Internet (see, e.g., [16]). WiFi (65.2 %) takes second place, and it is such a widely available pervasive technology that it is hardly surprising that it is the most selected choice for the physical/data-link layer. Then, we have Bluetooth (44.8 %) and its optimized version for IoT, Bluetooth Low Energy (42.1 %). As both protocols are well suited for low-range but low-power connections, their

[4] https://mqtt.org/

usage in so many projects does not come as a surprise. The others, Zigbee[5] (31.8 %), Near Field Communication[6] (24.2 %), CoAP[7] (20.9 %), 6LowPAN[8] (18.7 %), and Z-Wave[9] (9.5 %) are used by less than a third of the respondents.

We can catch discrepancies among the different clusters by reading the table by row. For example, we notice that MQTT is of greater interest for Development (76.3 % vs. 68.0 % of the overall population) and vice versa Research is much less into it, with only 47.6 %. Analogously we can see that WiFi is the most consistently chosen protocol over all clusters, being used by at least 63.5 % and that CoAP (20.9 %), 6LowPAN (18.7 %), and Z-Wave (9.5 %) are by far less popular, being used by much less than a quarter of the respondents.

By reading the table by columns, we can compare the interests of a specific cluster for the different protocols. The trends are similar, with the only notable inversion being WiFi, which comes out second for the overall population (65.2 %) and Development (64.0 %) but it is first for both Management (68.6 %) and Research (63.5 %). Moreover, for Research it is the only protocol used by at least half the population.

Finally, in general, Management percentages are above and Research percentages are below those of the overall population. This trend is likely due to the number of projects the respondents are involved in. Indeed, participants in Management role probably supervise more projects than people in Development role help to implement, and respondents in Research role only focus on a few highly challenging systems. Thus, Management has more opportunities for seeing a larger spectrum of protocols in use than the other categories, and Research is restricted to few instances, hence to few protocols.

If we look at languages, Python is preferred by the overall sample (61.0 %), Management (60.3 %) and especially Research (58.7 %), whose rather distant second choice is Java (52.4 %). Such a position reflects the centrality of Python as an easy-to-learn-and-use language that is specifically well suited to support data analysis for the bounty of data collected by many IoT systems. Development, on the other hand, favors Javascript more (63.4 % vs 62.8 % of Python), which is the close second for Management (58.7 %) and the overall sample (59.6 %). That might be given by Node.js[10] being a favorite with programmers building servers and infrastructures to collect and store information as well as by the popularity of Node-RED[11] to low-code programming for event-driven applications. Its popularity could also derive from an existing huge population of proficient programmers that prefer to stick to their preferred language, due to its increasingly widespread support by microcontrollers.

Java and C, with 44.8 % for the overall sample, are in third position, with Development preferring C (52.9 % vs. Java 39.5 %) and Research favoring Java (52.4 % vs. C 38.1 %). The capability of Java to run in different environments and, on the other hand, the great C performance for low-level programming can be the reasons for such placements.

Platform-specific and less well-established languages, like Go (7.5 %), Swift (6.1 %), and Rust (1.7 %), are probably more used in restricted development environments and rated poorly in our questionnaire, possibly due to our world-spread sample.

The last option for *Q1* was an open-ended answer. 77 participants proposed 156 entries, 26 protocols, 6 languages, and 124 other elements in categories close to protocols and languages, like, for instance, networks and frameworks. Out of all the items proposed by the participant, only three got at least ten votes (LoRaWAN, 23 votes, that is, 6.6 % of participants, C# and Narrowband IoT, both 11 votes, that is, 3.1 %). Considering that these votes were autonomously volunteered, not simply passively selected from our proposals, we believe they are interesting.

The personal opinion survey (27 respondents) presented in [17] also investigated the technologies used to develop IoT systems. However, it did not distinguish between languages, protocols, platforms, and tools.

Integrated Development Environments (IDEs)

To investigate which IDE, if any, were most popular, we asked an open-ended question (*Q4*), answered by 216 participants. The majority of them suggested multiple IDEs (or similar kind of tools that could be used as such), ten claimed to use many different ones and five flatly stated that they did not use any. Even after cleaning the data and clustering similar answers, we still have about sixty alternatives, most of which used by just one respondent. In Fig. 5, we list the alternatives suggested by at least ten participants[12].

The most favored by far is Eclipse (35.6 %), particularly liked by Research (48.8 %), as expected because it is a free and open source tool widely used by (senior) programmers in the Java community. The second choice is Visual Studio (17.1 %), which is mainly appealing to the Microsoft community and quite costly, and the third is the specialized IDE for Arduino[13] (15.3 %). Arduino's position in this ranking is a specificity of IoT development, where device programming plays an important role.

[5] http://www.techtarget.com/iotagenda/definition/ZigBee
[6] en.wikipedia.org/wiki/Near-field_communication
[7] coap.technology
[8] en.wikipedia.org/wiki/6LoWPAN
[9] http://www.z-wave.com
[10] https://www.nodejs.org/
[11] https://nodered.org/
[12] IDEs selected by less than ten participants: (9) Android Studio and NetBeans; (7) Keil; (5) Atmel studio; (4) Node-Red, Spyder, Texas Instruments Code Composer Studio and Webstorm; (3) Mbed Studio, MPLAB, Spring Tool Suite and Xcode; (2) Anaconda, GoLand, IAR, in house, Mendix, SAP Web IDE and ThingWorx; (1) AVR Studio, Brackets, Cloud9 IDE, Codeanywhere, CodeBlock, Contiki, Contineo, ESP IDF, Intel System Studio, Jupyter, MIIMETIQ Composer, PaizaCloud, Particle Dev, PhpStorm, PlatformIO IDE, Postman, PSoC Creator, Python IDLE, QT Creator, RStudio, Simplicity Studio, Sketch, SW4STM32, Turbo C++, WebForm, XCTU, Zelitron, Zerynyth.
[13] http://www.arduino.cc/en/software

IDE	ALL	Development	Management	Research
Eclipse	35.6%	32.0%	34.0%	48.8%
Visual Studio	17.1%	20.5%	17.0%	7.3%
Arduino	15.3%	16.4%	15.1%	12.2%
Visual Studio Code	13.0%	18.0%	7.5%	4.9%
Text editors	11.6%	13.9%	3.8%	14.6%
IntelliJ	6.5%	9.0%	3.8%	2.4%
Many different	4.6%	4.1%	3.8%	7.3%
Pycharm	4.6%	5.7%	3.8%	2.4%

Fig. 5. The most popular IDE choices, in percentage of participants.

Platform	ALL	Development	Management	Research
In house platform	65.7%	64.7%	65.0%	70.7%
AWS IoT Core	58.0%	57.1%	66.7%	41.4%
Azure Sphere	42.6%	41.8%	45.8%	36.2%
Bluemix	28.9%	29.4%	29.2%	25.9%
Other	28.0%	29.4%	29.2%	20.7%

Fig. 6. Percentage of participants using the different cloud platforms.

We differentiate Visual Studio Code (13.0 %) from generic text editors (11.6 %) due to its extensive plug-in ecosystem, which gives it capabilities closer to an IDE. While Visual Studio Code ranks between full-fledged IDEs and plain text editors, the combined share of these two categories is 24.6 %, making (enriched) text editors the second most popular toolset after Eclipse. This finding aligns with the dominance of Visual Studio Code and the presence of many text editors at the top of the Stack Overflow rankings for IDEs[14] in the wider software development community. Given that our respondents are IoT professionals, this suggests a strong preference for lightweight development tools in this domain. However, this preference may involve a trade-off between simplicity and efficiency. It is plausible that adopting more comprehensive, full-fledged IDEs could lead to marked performance gains. Validating this hypothesis requires a dedicated study to investigate the impact of tool choice on developer productivity and code quality in IoT projects, which represents an interesting direction for future research.

Cloud Platforms

The answers to question *Q2* clearly show that most participants used some cloud platform. Indeed, 97.1 % of the overall population claimed they used some. The percentage is even higher for the participants in Development (97.6 %) and Management (100.0 %), while respondents in Research using any platform were 93.1 %. Research's much lower platform usage could be due to its involvement in experimental projects not conveniently supported by existing platforms or worth developing an *ad hoc* one, as they are not intended for production.

In Fig. 6, we see the details about the usage of the different proposed platforms.

The most popular choice is an in-house platform (65.7 %), which leads the ranking for Developers (64.7 %) and Research (70.7 %). Management, instead, favors AWS IoT Core (66.7 % vs 65.0 % of in-house), which is the second choice for the overall population (58.0 %), Development (57.1 %), and Research (41.4 %). Interestingly, Research is strongly oriented toward in-house platforms, with a gap of almost 30 % between that and their second choice, AWS IoT Core. Such polarization might derive from the experimental nature of Research projects, which have special requirements not supported by standard industrial platforms, or from the fact that, in some cases, the project's goals included the development of innovative platforms. It could also reflect a mere budgetary concern, because using a in-house platform can save money on short-term projects. At the bottom of the ranking, we find Bluemix[15], used by only 28.9 % of the participants.

In addition to the listed alternative, *Q2* also offered the choice *Other*, which scored 28.0 % on the overall population, about 29 % on Management and Development, and as low as 20.7 % on Research. Question *Q3* asked participants who had selected that choice for further details about the other platform adopted. We received 91 answers clustered in 51 alternatives; three-quarters of them received only one vote, a few from two (2.2 %) to seven (7.8 %). The only values worth mentioning are the Google Cloud Platform, with 19 votes (21.1 %), and the cluster of custom-built platforms, with 14 votes (15.6 %).

Fig. 7 shows users' appreciation for the different proposed platforms. The horizontal lines highlight the first, second (median) and third quartile. The labels are the number of received answers in each category, with the mode in bold, larger text. Although, median and mode are *Satisfactory* for all platforms, the analysis of the first and third quartiles clearly shows that Bluemix is not only the less

[14] See, e.g., https://survey.stackoverflow.co/2024/technology#1-integrated-development-environment

[15] http://www.ibm.com/support/pages/overview-ibm-bluemix

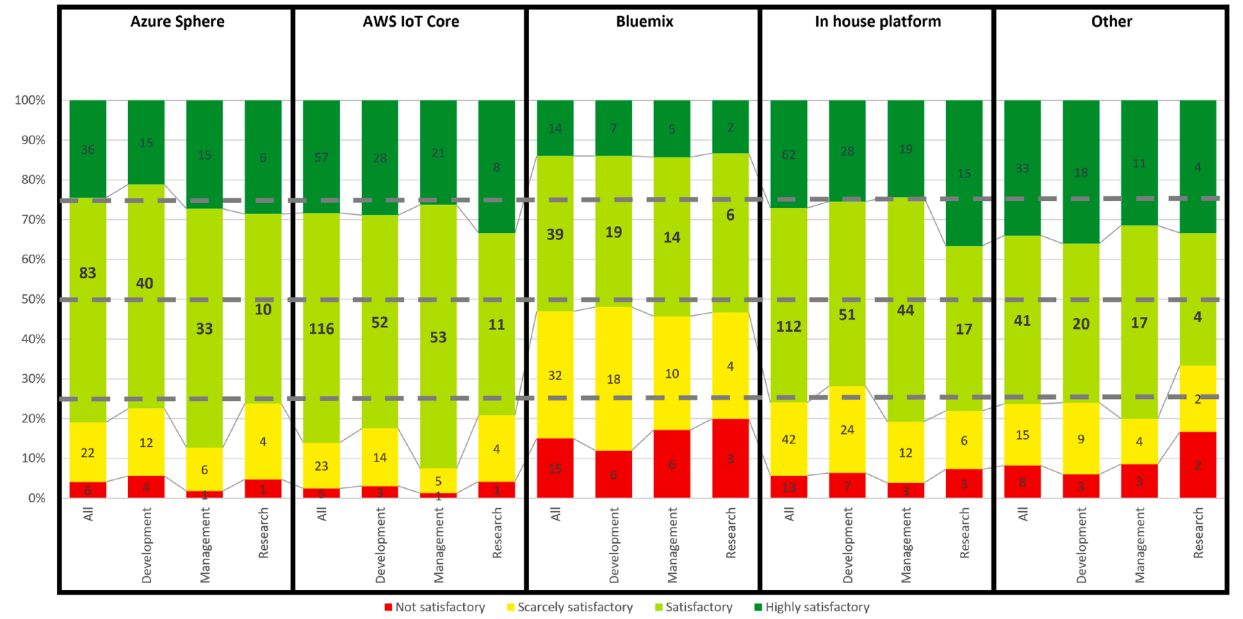

Fig. 7. Satisfaction of participants using the different platforms.

used, but also the less appreciated by its users, with the first quartile falling well within in the *Scarcely satisfactory* range and the third quartile in the *Satisfactory* range. In other words, less than a quarter of Bluemix users are very satisfied with it and more than a quarter (actually almost half) are scarcely satisfied or not satisfied at all. Vice versa, the first quartile corresponds to *Satisfactory* for all other platforms, and the third quartile to *Highly satisfactory* for AWS IoT Core, In house platforms, and Other. The third quartile for Azure sphere corresponds to *Satisfactory* but the *Highly satisfactory* votes are 36 and 25 % of all votes (147) is 36.75; thus, the result is very close to *Highly satisfactory*.

In Fig. 7, we also show the same analysis for the different clusters (Development, Management, Research) but there are no notable discrepancies between a cluster and the overall population worth discussing.

Services

Services are central to IoT system development. Indeed, from question *Q5*, we learned that 86.7 % of the overall sample used some kind of service. The percentage is even more impressive for Development (90.5 %), while it is slightly less substantial for Management (84.8 %) and Research (79.4 %). We conjecture that the impressive drop in Research usage mostly depends on the lack of availability of out-of-the-shelf services that can be of use in innovative systems and the adoption of easier, more traditional software architectures for the proofs-of-concepts so common in research. The last option of *Q5* was an open-ended answer for participants wanting to give extra details. However, only a handful of people took advantage of the opportunity, and it was impossible to gather information from the few scattered responses.

Code automation tools

The last technical aspect we analyzed is the usage of automatic code-generation tools. Question *Q6* asked whether the participant used automated tools to help with code generation and *Q7* provided an open answer for those who did it to detail which tool(s) they used.

We can conclude that automatic code generation is not central to IoT development, because barely 24.1 % of the population uses any.

The low response rate to *Q6* resulted in only 51 responses to *Q7* (28 Development, 15 Management, 8 Research), a mere 14.1 % of the surveyed population. Among the responses, Node-RED received the most votes, but still only 3, while SDK, SAP, Swagger Codegen, Thingworx, Docs, JAXB, Mendix, STM32, and CubeMx each received two votes. Ten participants answered that they used self-developed tools (mostly scripts).

4.2. *Approaches for software engineering tasks*

To investigate the software engineering practices and related technologies adopted in IoT development, we use the questions presented in the second section of Table 1.

Development process	All	Development	Management	Research
Agile/Scrum	69.8%	75.0%	74.2%	45.9%
Iterative	41.0%	38.1%	41.7%	49.2%
Waterfall	19.9%	19.6%	23.3%	14.8%

Fig. 8. Percentages of development processes usage by role.

Requirement documentation	ALL	Development	Management	Research
Use cases	91.5%	91.1%	95.9%	83.3%
Goals	88.1%	91.1%	90.2%	75.0%
Natural language/graphical documents following specific formats	78.8%	81.5%	79.5%	70.0%
Subsumed by other documents	77.3%	80.4%	82.8%	58.3%
Free-form natural language / graphical documents	70.3%	71.4%	68.9%	73.3%

Fig. 9. Requirement documentation - usage of different styles.

Development processes

We start our analysis with the overall development process. Question *Q8* asked our participants to identify the development method they adopted in their projects. As rough categories to choose from, we proposed *Agile methods*, (other) *Iterative methods*, *Waterfall*, and an open-ended answer. Fig. 8 details the responses received.

Considering the overall population, iterative methods, particularly agile ones, have the lion's share, as expected. In fact, *Agile methods* scored 69.8 % and (other) *Iterative methods* 41.0 %. This result is consistent with modern trends in software development. Instead, the fact that *Waterfall* scores 19.9 % is more interesting. We could have expected such a rigid method to be totally inapplicable in an ever-shifting domain like IoT. However, IoT systems often incorporate a far greater variety of heterogeneous hardware (e.g., sensors, actuators, custom boards) compared to traditional PC or server-based software. This hardware diversity intensifies the challenge of integration and can lead teams to favor the perceived stability and predictability of the Waterfall model for upfront planning, especially to solidify hardware-software interfaces early in the lifecycle. Furthermore, it is also plausible that this prevalence is influenced by the background of the respondents. Practitioners trained in more traditional engineering disciplines, as opposed to specialized software engineering, may be more accustomed to the Waterfall process, potentially leading to its selection by default.

The results of a small personal opinion survey (60 respondents among managers, QA analysts, and developers) reported in [18] confirm that waterfall is widely used in IoT. Hybrid methods combining waterfall with agile are also popular.

A discrepancy worth noting is the Research ranking, where *Iterative methods* are first (49.2 % vs. 41.0 %), followed by *Agile methods* (45.9 % vs. 69.8 %) and *Waterfall* (14.8 % vs. 19.9 %). A scarce interest in user involvement in experimental projects might justify, at least partially, the preference for iterative non-necessarily agile methods.

About 20 participants selected the open-ended answer, with one vote for DevOps, one for Kanban, one for Co-Creation, eight for personalized methods, and eight for development without any method.

Requirement Documentation

The starting activity of any development process is understanding the project requirements and somehow capturing them in the project documentation. Documenting requirements can be done at three different levels of formality, depending on the development method adopted. At the most basic level, requirements are implicit and captured, for example, in contracts or project plans. In most methods, though, they are described in specific documents. Such documents may be written in free natural language, possibly complemented by freestyle visual representations, like mock-ups. This is the second level of documenting requirements, isolating requirements from other project issues, but leaving the choice of the document format to the developers. At the third and topmost level of formality, a specific language/notation structure for such documents is prescribed by the development method. For instance, many agile methods employ user stories that follow the schema *I, as a…want to…so that*. These three levels appear as possible answers for *Q9*, together with two popular specific techniques for requirement documentation: *Use cases* and *Goals*. Although less well-established than use cases, goal-based requirement specifications seem promising for IoT systems. For example, in a recent systematic mapping study on requirement engineering for IoT, six of the 24 primary studies selected are goal-based (see [19]).

Fig. 9 shows that all choices are rather popular. All clusters favor *Use cases*, scoring over 90 % for the overall population (91.5 %), Development (91.1 %), and Management (95.9 %), and a close 83.3 % with Research.

Except for free-form documents in natural language and graphics, which lag behind at 70.3 %, the other choices also collect an impressive number of votes, ranging from 88.1 % for *Goals* to 77.3 % for describing the requirements through other documents. In particular, *Goals*, though less well-established in general than *Use cases*, are a close second, possibly because they do not rely on the concept of user for requirement description and are appropriate for IoT systems enriching the environment with functionalities not tailored for any specific user. As expected, Research is much less prone to confining requirements to other documents because it is less involved in standard industrial project management providing contracts and project plans that implicitly describe requirements.

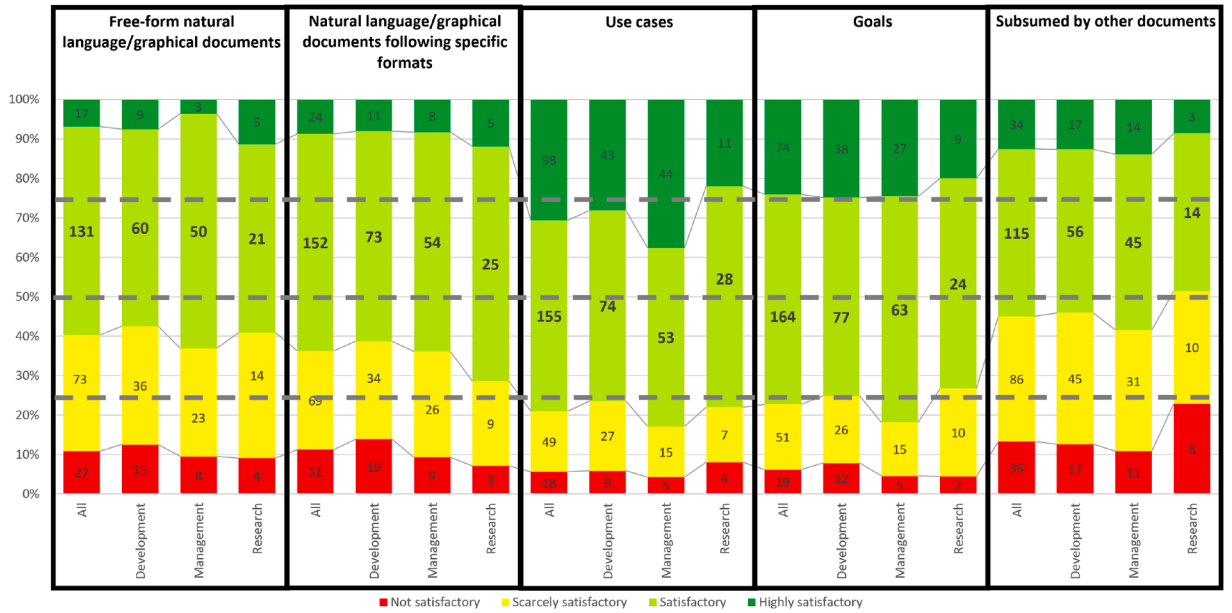

Fig. 10. Requirement documentation - user satisfaction for different approaches.

Design documentation	*ALL*	*Development*	*Management*	*Research*
Natural language/graphical documents following specific formats	**84.5%**	86.1%	86.8%	75.0%
Free-form natural language / graphical documents	**83.0%**	84.8%	82.6%	80.0%
Models produced using a modelling notation (e.g., UML Class Diagram)	**72.1%**	73.3%	70.2%	71.7%

Fig. 11. Design documentation - usage of different styles.

Use cases are the most satisfactory techniques for requirement documentation, being the only choice having the first quartile in the *Satisfactory* range and the third quartile in the *Highly satisfactory* range. *Goals* are not far behind, the first quartile in the *Satisfactory* range and the third quartile in the upper end of the *Satisfactory* range, with 74 votes for *Highly satisfactory* out of 308 (24 %).

Fig. 10 also details the satisfaction levels for the different clusters (Development, Management, and Research), which differ marginally from those of the overall populations.

Design documentation

Analogously to the case of requirement documentation discussed previously, with question *Q10* we investigated the diffusion and appreciation of documentation techniques for design.

Fig. 11 shows that structured documents in natural language/graphics are the preferred choices (84.5 %), followed by those in free format (83.0 %), while modeling notations lag behind (72.1 %). Research, instead, prefers free format documents (80.0 %) over structured ones (75.0 %).

Fig. 12 shows the appreciation levels for the different kinds of design documentation, both for the overall populations and for the Development, Management, and Research clusters. The three alternatives have similar appreciation levels by the overall population, with mode and median corresponding to *Satisfactory*, first quartile to *Scarcely satisfactory* and third quartile to *Satisfactory*. However, if we consider the position of the quartiles within the ranges, we can see that modeling notations are the least used in practice but also the most appreciated, and that free-form documents are *Scarcely satisfactory* for a much larger population than structured ones. Research is more critical on all techniques, but notably so on free-form documents being the only cluster with the first quartile well within the *Scarcely satisfactory* range. That is peculiar, considering that such a technique is the most popular for Research.

The finer grained analysis of appreciation, showing the superiority of modeling notations and the limits of free-format documents, suggests that investing some effort in adopting modeling notations and reducing the usage of free-form documents could substantially improve the design quality of innovative systems.

Sixty users answered *Q11*, leaving feedback on the specific language/notation used to model design. As expected, the most popular was the UML, with 37 votes. The other answers are rather scattered but show two aggregation clusters:

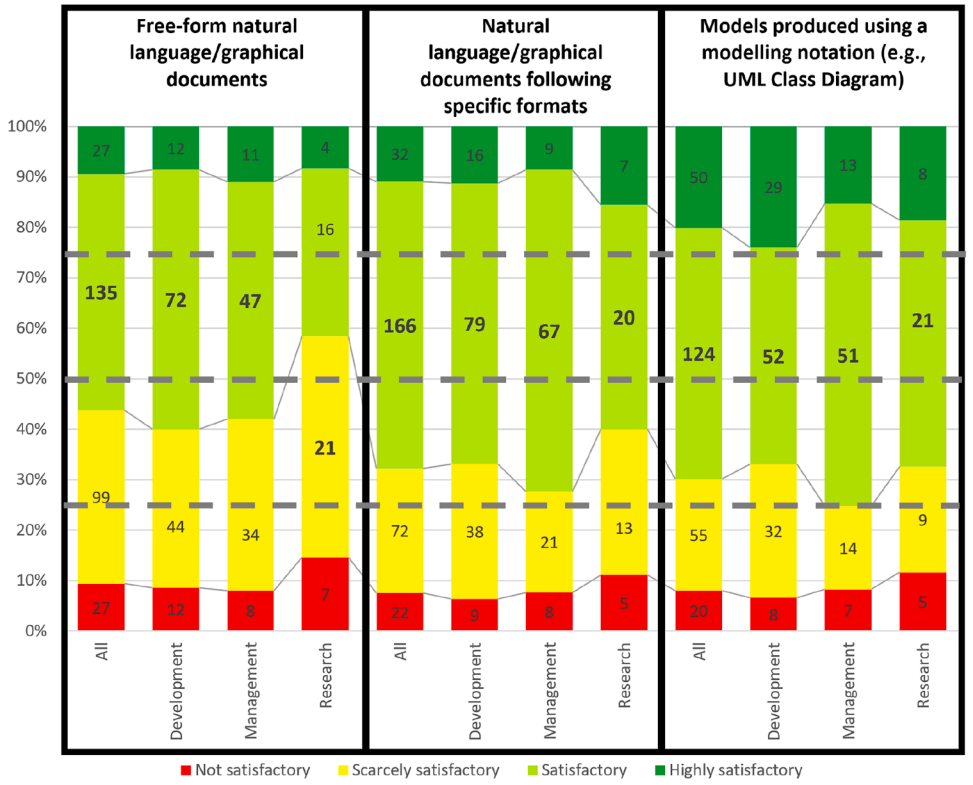

Fig. 12. Design documentation - user satisfaction for different styles.

Testing type	*ALL*	*Development*	*Management*	*Research*
Integration testing	**88.5%**	90.2%	93.9%	74.1%
Unit testing	**86.1%**	87.2%	90.4%	74.1%
Acceptance testing	**79.6%**	81.7%	86.1%	60.3%
Other	**20.6%**	21.3%	21.7%	15.5%

Fig. 13. Testing - usage of different types.

- descriptions of component interactions and system architecture, using system description languages like SysML or C4Model[16]
- focus on user interactions, using UX languages and use cases.

Testing

With question *Q12*, we investigated the popularity and appreciation of a few widespread testing types [20]. Figs. 13 and 14 show, respectively, the diffusion and satisfaction rates of different types of testing.

The percentage of testing usage is more pronounced than in other domains, possibly due to the greater complexity and higher risks in IoT development, and their ranking order is different as well. For example, a 2021 review by JetBrains[17] found out that 75 % of all respondents say that testing plays an integral role in their development, and the kinds of testing they have in their projects are Unit (67 %), Integration (48 %), End-to-End (33 %), Performance (31 %), Other (1 %). Analogously, in 2023, Gartner administered a more detailed questionnaire[18] to 248 IT and software engineering leaders proposing many kinds of automated testing and asking about their adoption. They found out that *API testing (56 %), integration testing (45 %) and performance testing (40 %) were among the most common types of automated software testing currently in use in respondents' organizations*, with Unit testing lagging well behind (24 %).

In our findings, usage rates have the same ranking across the different clusters. At the top, we find integration testing [21] (88.5 %). Indeed, IoT systems have to deal with a higher level of complexity than other software in designing interactions among the many different components, often across different protocols and networks.

[16] en.wikipedia.org/wiki/C4_model
[17] https://www.jetbrains.com/lp/devecosystem-2021/testing/#Testing_what-types-of-tests-do-you-have-in-your-projects
[18] https://www.gartner.com/peer-community/oneminuteinsights/omi-automated-software-testing-adoption-trends-7d6

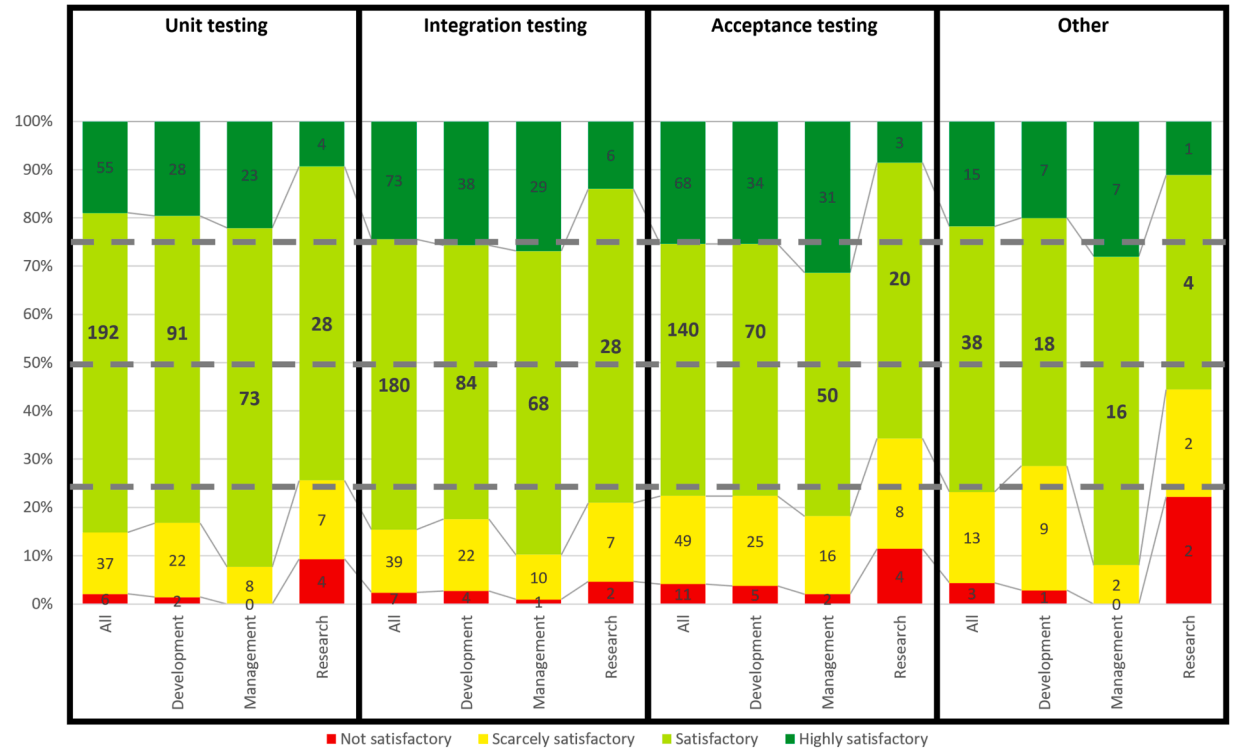

Fig. 14. Testing - user satisfaction for different types.

Then, we have unit testing (86.1 %), whose higher popularity might be related to the need to perfect components before installing them in an aggressively distributed environment, where recalling, correcting, and distributing the faulty parts again could be extremely expensive or unfeasible (in those cases where automatic online updates do not work because of low connectivity or anti-tempering mechanisms stopping them). Another reason that could contribute to the greater popularity of unit testing in IoT systems is that, given the relatively recent introduction of the domain, IoT systems are mostly new projects started from scratch when the culture of unit testing had already been established.

Finally, we find acceptance testing (79.6 % - still higher than the percentages of all testing kinds in the previously quoted JetBrains and Gartner surveys) whose perceived lesser relevance might be connected to the large number of IoT systems produced and sold on the market (or as a service) not commissioned by an external client.

We also provided an open-ended question for respondents selecting the *other* option to detail which different kinds of testing they performed. Only about forty people answered that question, and the answers were wide spread. A single respondent mentioned *formal verification*, suggesting that such a technique has a limited impact on IoT development. Overall, *non-functional* testing is the most popular category, with load and security testing preeminent among those answers, along with hardware testing, which is to be expected given the massive usage of nonstandard hardware components in IoT systems. This result agrees with the findings in [22] where the authors report the preeminence of testing of operational safety, conformance, interoperability, robustness, and comprehensiveness of IoT devices.

Fig. 14 shows that mode and all quartiles (including the median) correspond to *Satisfactory* for all testing types, suggesting that standard testing techniques are adequate for IoT systems. However, the third quartile for Integration and Acceptance testing is on the border with *Highly satisfactory*, highlighting the importance of testing at system level for IoT system, and the third quartile for Acceptance and Other-testing-kinds is very close to the *Scarcely satisfactory* range. The high percentage of scarcely satisfied (or totally dissatisfied) users for Acceptance testing goes hand-in-glove with the lower percentage of usage, possibly confirming the difficulties of application in a context where clients and final users are not a clear part of the development process.

If we compare the different clusters, we can see that Research is more critical about all testing types, especially the Other category. A possible reason is the minor percentage of professional testers in research departments, where innovation is the focus.

4.3. IoT System development challenges

The research question RQ3 addresses the perceived differences, challenges, and needs related to developing IoT systems using the data collected through the questions listed in the third section of Table 1. Question *Q14* is a simple boolean question asking whether the development of IoT systems is different. A large majority of the overall population (69.8 %) finds developing IoT systems different from other systems. Such specificity is perceived more strongly by Management (74.6 %) and Research (68.3 %) than by Development

Fig. 15. Main differences between IoT and other systems.

(66.5 %), possibly because more traditional subsystems are part of most IoT systems so that developers of those parts are not aware of differences, or, possibly, because developers tend to limit their focus to technical difficulties, which are somehow independent from the field of application. This difference is still strongly perceived by a survey described in [17], which reports that only "3 out of 127 respondents did not believe much noticeable distinctions in IoT development process challenges compared to the conventional software development".

When asking for more information about the specificities of IoT system development, we received 203 free text answers. We first split each answer into its components because most of the respondents proposed several reasons for the difference in their single answer. Then, we made the terminology uniform through an iterative process done by two of the authors, that led to classify the inputs by a common vocabulary. Then, the third author reviewed the results and we consolidated our clusters tagging them with 35 keywords. Fig. 15 proposes a tag cloud of those common terms, where text size is related to frequency, while the colors do not have meaning.

Then, we aggregated the keywords into categories to abstract away from the details and gain insights into the different realms. We identified the following categories (in decreasing order of interest for the overall population).

Development complexity includes all aspects of the development process. Design complexity is preeminent in this category, followed by the need for a broader development skill set that spans the entire development stack. The greater relevance of debugging, quality assessment, and customer interactions also belongs here.

Physical concerns consist of the differences caused by the greater relevance of physical aspects in IoT systems, such as the prominence of appropriate hardware components and the need for interaction with reality.

Highly distributed systems correspond to the very nature of most IoT systems, which are inherently distributed. Thus, selecting networks and protocols, choosing where to have computations and store data, and designing the underlying connection graph for system components are more relevant than in many traditional systems.

Quality requirement relevance is not a unique prerogative of IoT systems, of course. However, security, reliability, resilience, availability, performance in a real-time environment, and evolvability are crucial to the success of most IoT systems.

System complexity captures the complexity of the resulting IoT systems, their need for scalability, and the difficulty of mastering heterogeneity and making different components correctly interact.

Main differences in IoT development	ALL	Development	Management	Research
Development complexity	30.0%	31.9%	34.2%	17.9%
Physical concerns	26.1%	26.4%	21.9%	33.3%
Highly distributed systems	24.6%	30.8%	15.1%	28.2%
Quality requirement relevance	21.2%	26.4%	16.4%	17.9%
System complexity	19.2%	19.8%	24.7%	7.7%
Hardware limitations	15.3%	13.2%	15.1%	20.5%
Innovation	10.8%	7.7%	16.4%	7.7%
Data centrality	7.9%	7.7%	8.2%	7.7%
Unpredictability	4.4%	2.2%	4.1%	10.3%

Fig. 16. Main differences in IoT development by role.

Main SE challenges in IoT development	All	Development	Management	Research
Testing	48.6%	48.0%	52.4%	43.9%
Frameworks	47.3%	47.3%	40.8%	57.9%
Requirement capture and specification	44.1%	44.0%	49.5%	35.1%
Design	43.4%	48.0%	40.8%	36.8%
Development processes	39.9%	42.7%	35.9%	40.4%
Tools	37.9%	33.3%	40.8%	45.6%
IDEs	26.0%	27.3%	19.4%	35.1%
Languages	22.2%	24.7%	13.6%	31.6%

Fig. 17. Main software engineering challenges in IoT development.

Hardware limitations capture aspects such as device size and costs, resource limitations, and problematic power consumption.

Innovation groups the need for discovering new kinds of functionalities and interaction modes, to take full advantage from the innovative components developed daily.

Data centrality concerns aspects like data collection and analysis, as well as the difficulties posed by their huge volumes.

Unpredictability concerns the difficulties of long- and medium-term planning in an ever changing field, where costs, context, parts, and their evolution are often far away from the designers' initial expectations.

Fig. 16 presents the distribution of choices among categories. Each role has different perceptions of the differences that are most relevant. From the Development viewpoint, the topmost categories are *development complexity* (31.9 %), *highly distributed systems* (30.8 %), and *physical concerns* together with *quality requirement relevance* (26.4 % both). In other words, Development mostly perceives changes in the development process itself. It identifies distribution, hardware relevance, interaction with the physical world, and increased relevance of quality attributes as the primary differences that separate IoT from traditional development. Management also considers *development complexity* (34.2 %) the most relevant difference but reckons *system complexity* (27.4 %) the second factor. *Physical concerns* (21.9 %) rate third in Management ranking. Finally, Research finds that *physical concerns* (33.3 %), *highly distributed systems* (28.2 %), and *hardware limitations* (20.5 %) are the main causes of differences in IoT system development; they focus on the most technical aspects where cutting-edge technologies must be developed.

While question *Q15* helped us understand the areas where developing IoT systems differs more from standard software development, with question *Q16* we wanted to investigate where the IoT-specific challenges lie.

Fig. 17 shows the distribution of choices for the predefined options. The overall population considers testing (48.6 %) and frameworks (47.3 %) the most challenging aspects, followed by requirements (44.1 %) and design (43.4 %), and development processes (39.9 %) and tools (37.9 %). IDEs (26.0 %) and languages (22.2 %) are appropriate for three-quarters of respondents.

Development mimics the global ranking, more or less, except that they consider design (48.0 %) more challenging than requirement capture and specification (44.0 %).

Quality assurance is the first concern for Management as well (52.4 %), but their second choice is requirement management (49.5 %), with frameworks, design, and tools well behind in the ranking (40.8 %). Considering testing and requirements as the main challenges might be related to the lesser degree of Management involvement in the central parts of the development process, design and implementation.

The Research ranking is quite different, with frameworks at the top (57.9 %), tools (45.6 %) and testing (43.9 %) well below. Moreover, the other values are distributed in a smaller range, with even the less challenging aspect, languages, scoring as high as 31.6 %. Having innovation as a mission, Research is actively engaged and in constant contact with the challenges of any task. The last option of *Q16* was an open-ended answer that about twenty participants selected. Most entries (thirteen) were about technological

Knowledge sources for IoT development	ALL	Development	Management	Research
Reading (web/printed) material provided by, e.g. companies producing IoT devices / software, blogs, tech survey sites	69.4%	74.4%	69.7%	57.1%
Attending IoT courses	41.2%	51.2%	36.9%	20.6%
Reading books	40.7%	43.6%	35.2%	44.4%
Scientific literature	36.8%	30.2%	27.9%	73.0%

Fig. 18. Knowledge sources for IoT development.

issues, all different. Three participants found the need to start any development from business concerns challenging. Three respondents saw security as a problem area that needs thorough investigation, while three others were concerned about analytics techniques.

4.4. Knowledge sources for IoT development

Our last research question concerns the sources and processes that help professionals learn about IoT system development. To answer RQ4, we used question *Q17*, reported in the last section of Table 1. Fig. 18 presents the predefined options and the percentages of votes they got. The most popular source by far is the material provided by IoT companies, such as product tutorials and introductory reports, and by technical leaders through surveys, blogs, and websites.

That answer was voted by 69.4 % of the overall population and is even more popular with Development (74.4 %). Attending IoT courses and reading books are close together in the second and third positions for the overall population (41.2 % and 40.7 %) and Management (36.9 % and 35.2 %). In comparison, Development differentiates more between them (51.2 % and 43.6 %). Scientific literature appeals to less than a third of Development (30.2 %) and Management (27.9 %), while is the first choice for Research with an astonishing 73.0 %.

Besides the proposed options, *Q17* also offered an open-ended answer that 91 participants selected. A handful of them merely rephrased one of the predefined options, in which case we counted their input as an extra vote for our formulation. The markedly different answers mostly fell into the *learn-on-the-job* category and reached noteworthy rates: 21.4 % for the overall population (18.6 % for Development, 32.8 % for Management, and 7.9 % for Research). Very few open-ended answers were for self-learning (1.4 %) and participation in IoT events (1.1 %).

4.5. Synthesis of results

We conclude this section with a brief discussion on what can be inferred about the state of practice. Our research questions primarily focus on comparisons among technical alternatives. Thus, our findings do not readily lend themselves to broader discussions. However, we can highlight the most interesting results of each research question.

RQ1 What software technologies are currently most used to develop IoT systems?

The analysis shows that MQTT and WiFi are the dominant protocols. Thus, a strong understanding of both is a must for practitioners. MQTT is particularly favored by Development (76.2 % vs. 68.0 % among all respondents) but not a focus of Research (47.6 % only). The most commonly used programming languages are Python and JavaScript, followed by Java (less used by Development, with 39.5 % vs 44.8 % for all respondents) and C (favored mainly by Development, with 52.9 %, vs 37.2 % for Management and 38.1 % for Research). Such a mix of high-level languages like Python and JavaScript (favored for data analysis and servers) and low-level languages like C (preferred for performance-critical device programming) gives a substantial advantage to programmers proficient in both.

For development environments, Eclipse is the most popular IDE, especially among Research (48.8 % vs. 32.0 % among Development, for instance), followed by Visual Studio and the Arduino IDE. The data also shows a strong preference for lightweight, extensible editors like Visual Studio Code (13.0 %) and other less specialized text-editors (11.6 %). Simpler tools usually provide less support for day-by-day tasks. Thus, developers should be more autonomous in their programming to fit in a working environment where lightweight development tools are extensively used.

Cloud platforms are widely adopted, with in-house solutions and AWS IoT Core being the most common and well-appreciated. The former is common mainly in research contexts (70.7 % vs. 65.7 % among all respondents), and the latter in industrial projects (66.7 % among Management and 57.1 % among Development, compared to 41.4 % among Research). Finally, services are ubiquitous (their usage ranging from 79.4 % among Research to 90.5 % among Development). Conversely, automatic code generation tools are not considered central to IoT development, with only 24.1 % of respondents using them.

RQ2 What approaches are used nowadays to develop IoT systems concerning development processes and selected specific software engineering tasks?

Iterative development processes, particularly Agile methods, are the most widely adopted (69.8 %), involving almost three-quarters of developers (75.0 %) and managers (74.2 %) but less than half of researchers (45.9 %). However, the Waterfall

model is still used by nearly 20 % of practitioners (19.6 % of Development and 23.3 % of Management), likely due to the stable design requirements of hardware components. This suggests a mismatch between currently preponderant agile practices and the hardware-software nature of IoT that should be addressed by proposing new hybrid methods.

For documenting requirements, Use Cases are the most popular and satisfactory technique (91.5 %), closely followed by Goals (88.1 %). Both are deemed satisfactory by at least three-quarters of the respondents. More than three-quarters of the practitioners are also familiar with requirements subsumed by other documents (80.4 % of Development and 82.8 % of Management), while the usage percentage among Research drops to 58.3 %.

For design, while documents in natural language are the most common (84.5 %), formal modeling notations like UML are less frequently used (72.1 %), but are more appreciated by those who do use them. On the one hand, researchers could focus on lowering the barrier to entry for formal modeling by proposing lightweight, IoT-specific modeling languages. On the other hand, practitioners should consider investing time to learn and apply these techniques over more common free-form documents as they can lead to higher-quality designs and greater professional satisfaction.

The most performed testing activities are integration (88.5 %) and unit testing (86.1 %). These figures are particularly notable compared to other domains; for instance, Madeja et al. [25] estimate that only 38.84 % of Java public projects have at least a significant test. The perceived relevance of testing in our study reflects the complexity of IoT systems. However, acceptance testing has a high rate of dissatisfaction, possibly because it is difficult to apply when there isn't a clear client or end-user, and researchers should address this issue by proposing testing techniques specific for environments where the *user* is not a clear entity.

RQ3 What are the challenges and needs related to developing IoT systems?

A large majority of experts (69.8 %) perceive the development of IoT systems as different from that of non-IoT systems. The primary difficulties identified are the overall complexity of development (which includes the need for broader developer skills), the challenges related to physical concerns (such as the more impactful relevance of hardware and the common need to interact with the real world), and the nature of highly distributed systems. Management should translate such needs into more generous timelines, budgets for acquiring broader skill sets, and a strategic focus on system architecture.

When asked about specific software engineering challenges, practitioners ranked testing (48.6 %) and the use of frameworks (47.3 %) as the most critical problem areas. New methods are needed for testing IoT systems, particularly at the system level, and for non-functional testing, such as load and security testing. The advance in research should be complemented by a shift in the management attitude, prioritizing and budgeting for comprehensive testing activities, with a special focus on integration testing and non-functional testing like security and load, which are critical in complex IoT systems. There is also a clear need for more robust, scalable, and secure IoT frameworks because the popularity of *in-house* cloud platforms further suggests that commercial offerings are not fully satisfactory, creating an opportunity for research into new platform architectures and open-source frameworks.

RQ4 How do people get their knowledge about IoT systems development?

While the research community considers scientific literature as a primary source of knowledge (73.0 % vs. 30.2 % for Development and 27.9 % for Management), professionals favor informal, non-academic material, such as tutorials and blogs provided by companies and tech leaders (74.4 % for Development, 69.7 % for Management, and 57.1 % for Research). Their second-ranked method of learning is attending IoT courses and reading books (51.2 % for Development, 36.9 % for Management, and 20.6 % for Research). Furthermore, learning on the job was a noteworthy write-in answer, particularly for participants in management roles (32.8 % for Management, 18.6 % for Development, and 7.9 % for Research).

Companies should acknowledge the preponderant role of experience as a skill-building source by creating mechanisms to capture and share lessons learned from projects, turning individual experiences into valuable institutional knowledge. In particular, managers should provide a working environment facilitating the sharing of experiences, best practices, and knowledge within each team and across different ones.

5. Related work

To the best of our knowledge, no other survey has been run with goals similar to ours. In the previous sections, we have cited several papers related to specific points investigated by our survey. Here, we report on empirical studies investigating IoT systems and their development that partly share our goals.

Among the several papers that have addressed the aspects that make IoT development more demanding, the survey presented in [23] states that security, privacy, interoperability, scalability, availability, reliability, communication standards, and ethical/legal issues are the most challenging aspects of IoT. In contrast, the authors of [24] think that the most relevant issues of IoT are powering and maintaining devices out of reach, guaranteeing high-speed and high-capacity Internet access, making IoT devices self-configuring, communication standards, security, and privacy.

Aguilar-Calderon et al. conducted a systematic mapping study on requirement engineering for IoT systems, examining various scientific literature repositories. They identified 24 primary studies and presented their findings in [19]. The study concludes that there is a lack of well-defined proposals for effectively developing IoT software systems with an incorporated requirement engineering

phase. Furthermore, the literature does not place sufficient emphasis on this topic. These results align with Fig. 17, which indicates that researchers do not consider the requirement issue for IoT systems a major challenge. Fig. 17 shows that both management and development still perceive requirements as a challenge in the context of IoT. This finding is also consistent with Fig. 9, which demonstrates the widespread use of currently available techniques for requirements specification, possibly because of the lack of an IoT-specific standard de facto for requirements.

Kowatsch et al. [26] investigate the privacy concerns of IoT systems in the European community and Akbar et al. [27] conducted a literature review to identify the risks related to IoT data privacy and security. They validated the review's findings by conducting an empirical study with IoT practitioners using a questionnaire.

Psychoula et al. [28] investigated the IoT system users' perception of privacy by a personal opinion survey (236 respondents). In general, users often lack a clear understanding of what data is being collected, how it is being collected, or why. Elderly individuals are more open to data sharing and less concerned about privacy compared to younger generations. The study also shows that regulations and robust security mechanisms play a crucial role in the acceptance of IoT technology.

Prasher in [29] presents the results of a small opinion survey (34 participants) on IoT project management. A question in that survey was about the adopted development process and, similarly to our results (see Fig. 8), the most used were agile and hybrid agile-waterfall, but the pure waterfall is confirmed to be still quite used.

Motta et al. present in [30] "14 significant concerns and seven facets that together represent the engineering challenges to be faced both by research and practice towards the advancement of IoT in practice" resulting from both examining the literature and interviewing the practitioners. Some concerns belong to software engineering: testing and requirements (first and third among the challenges revealed by our survey), whereas, [30] lists the issues related to design architecture, software qualities as scalability, and interoperability, that in our survey were subsumed by design (fourth challenge), see Fig. 17. [30] also reports the relevance of regulation, management and the preparation of professionals.

Uddin et al. [31] and Mohab et al. [32] present the results of two surveys about questions asked on Stack Overflow concerning IoT and also Industry 4.0 in the second one. Both report that around 40 % of the queries concern software, whereas the others focus on network, hardware, and platform management. Furthermore, the questions related to software ask specific points about programming languages and frameworks. Software engineering issues are not reported in either of the two studies. The reason may be that, in general, software engineering issues are not the most popular on Stack Overflow.

Fahmideh et al. [17] investigate the development of IoT systems by first identifying the key tasks (27 covering analysis, design and implementation) using a literature survey and later validating the results by a personal opinion survey concerning 127 developers.

Hornos and Quinde present in [33] a (non-systematic) literature review concerning the available approaches for developing IoT-based systems, highlighting their benefits and limitations. The results show that despite the considerable efforts and advancements being made, there are still many significant challenges in this research area. The paper also highlights the importance of searching for a robust and well-established approach and that more research is needed to address the complex nature of IoT-based systems. Unfortunately, our survey does not consider the issue of an overall approach.

Many researchers investigated specific aspects of IoT-based systems using systematic literature reviews; for example, [34,35] focused on the authentication issue, whereas [36] investigated IoT systems for urban disaster management.

A survey on the future of IoT [37] run in 2023 by Arm (a company developing IoT platforms) engaging with more than 700 developers, managers, and executives in the IoT space, reports that the top challenge, cited by 60 % of respondents, is the lengthy and complex software and product development cycles. The other relevant challenges are porting software to newer and more performant hardware (55 %) and achieving the right level of performance (52 %). However, the survey did not include any questions to investigate the software engineering-related issues.

6. Conclusion and future work

We have administered a questionnaire to 361 experts from 53 countries to understand the status of IoT system development, the usage of software technologies and software engineering practices, and IoT-specific approaches, challenges, and needs, if any. Each closed answer in our questionnaire was complemented by an open question asking whether some relevant alternative was missing from the listed possible answers.

We focused on software technologies (RQ1) and software engineering practices (RQ2). For the former, we analyzed protocols, languages, IDEs, cloud platforms, services, and code automation tools to investigate the technologies of greater interest for IoT development. For the latter, we investigated development processes, notations used for requirement and design documentation, and testing types. In both cases, we presented the graded list of the most popular choices and the corresponding satisfaction levels of the users for each category investigated.

To better understand the challenges related to the development of IoT systems (RQ3), we asked our respondents whether they found such development different from that of more standard systems. A large majority answered positively. We followed up that question with an open-answer one, asking for the main differences. The answers were, as to be expected, various. However, their topmost categories were development complexity due to the many aspects involved, physical concerns, particularly the prominence of hardware issues, and the high degree of distribution inherent to IoT systems. Furthermore, we collected the IoT-specific challenges through an open-ended question. Almost half of the respondents chose testing and frameworks.

Finally, we investigated the most common sources of knowledge about IoT systems development (RQ4). A large majority of practitioners learned autonomously from non-academic material.

Our analysis provides a detailed, data-driven snapshot of the field, offering valuable insights into its state and future directions. To conclude our work, we summarize the most significant takeaways for researchers, developers, managers, and professional educators.

Our survey highlights several promising *research* directions. First of all, the popularity of waterfall approaches shows that there is a need for new hybrid development methods that bridge the gap between agile software practices and the rigid demands of hardware development. A second interesting research topic is working on lowering the barrier to entry for formal modeling by proposing lightweight, IoT-specific modeling languages that are more accessible to practitioners. Indeed, formal methods are adopted by a minority, but have a very high appreciation rate by those who use them, and this suggests that making their adoption easier would largely benefit the community. On a more technical side, the widespread use of in-house cloud platforms suggests that current commercial offerings are not fully meeting user needs. This creates a clear opportunity to research new, robust, and secure open-source frameworks and platform architectures. Finally, the high rate of dissatisfaction with acceptance testing in contexts without a clear end-user calls for new, specialized testing techniques. Furthermore, there is a dire need for new methods to address non-functional testing, such as load and security testing, which practitioners ranked as a top challenge.

Accordingly to our findings, on the technological side, a strong understanding of MQTT and WiFi protocols is essential for *developers*, as they are the most used in the field. Proficiency in both high-level languages like Python and JavaScript (favored for data analysis and servers) and low-level languages like C (preferred for performance-critical device programming) offers a marked advantage. Finally, to fit in with current trends, *developers* should be autonomous in their programming, as there is a strong preference for lightweight, extensible editors like Visual Studio Code over more comprehensive, full-fledged IDEs. From the point of view of processes and development methods, *developers* should consider investing time in learning formal modeling notations like UML, as they are more appreciated by those who use them compared to more common free-form documents. They should also be prepared to work within hybrid development processes, give the persistence of the Waterfall model alongside Agile methods.

Managers should account for the unique development complexity of IoT by providing more generous timelines and budgets for acquiring broader skill sets and a strategic focus on system architecture. They should also prioritize and budget for comprehensive testing activities, with a special focus on integration testing and non-functional testing like security and load, which are critical in complex IoT systems and identified as a top challenge by practitioners. Finally, given that learning on the job is a primary source of knowledge accordingly to practitioners, *managers* should create and facilitate mechanisms to capture and share lessons learned from projects, helping turn individual experiences into valuable institutional knowledge that can be shared within and across teams.

Professional Educators can help improving IoT development under two respects. On the one hand, they can design core curricula with a sufficient coverage of dominant technologies, including MQTT, WiFi, Python, JavaScript, and C. Any curriculum should reflect the industry's reality by teaching not only Agile methods but also Waterfall and hybrid models, explaining the contexts in which each is applied. A prompt introduction of novel hybrid methods could amplify the impact of research and quicken the improvement pace greatly. Particular emphasis should be placed on robust testing methodologies, especially integration and non-functional testing, which are perceived as highly relevant and challenging in the industry. Students should be taught formal modeling notations and their practical benefits over informal documentation to better prepare them for developing high-quality designs and contributing to the collective knowledge of the team to further process improvements. On the other hand, educators could play a relevant role in lifelong learning initiatives aimed at upskilling the existing workforce by creating and disseminating content that bridges the gap between formal literature favored by researchers and the plethora of blogs and tutorials that practitioners overwhelmingly rely on currently.

CRediT authorship contribution statement

Maura Cerioli: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Data curation, Conceptualization; **Maurizio Leotta:** Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Methodology, Investigation, Data curation, Conceptualization; **Gianna Reggio:** Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Methodology, Investigation, Data curation, Conceptualization.

Data availability

Data will be made available on request.

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.

Acknowledgments

The authors are deeply grateful to the anonymous reviewers for their exceptionally thorough and constructive comments. Their detailed suggestions and thoughtful feedback went beyond standard review practices and were instrumental in enhancing the quality and clarity of this paper.

References

[1] IoT Analytics, State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally, 2024, (IoT Analytics Website). Iot-analytics.com/number-connected-iot-devices/.
[2] OECD, Measuring the Internet of Things, 2023, (OECD Website). http://www.oecd.org/en/publications/measuring-the-internet-of-things_021333b7-en.html.
[3] G. Reggio, M. Leotta, M. Cerioli, R. Spalazzese, F. Alkhabbas, What are IoT systems for real? An experts' survey on software engineering aspects, Internet Things 12 (2020) 100313. https://doi.org/10.1016/j.iot.2020.100313
[4] B.A. Kitchenham, S.L. Pfleeger, Personal opinion surveys, in: Guide to Advanced Empirical Software Engineering, Springer, 2008, pp. 63–92.
[5] G. Reggio, M. Leotta, F. Ricca, Who knows/uses what of the UML: a personal opinion survey, in: Proceedings of 17th International Conference on Model Driven Engineering Languages and Systems (MODELS 2014), 8767 of *LNCS*, Springer, 2014, pp. 149–165. https://doi.org/10.1007/978-3-319-11653-2_10
[6] V.R. Basili, G. Caldiera, H.D. Rombach, The goal question metric approach, Encycl. Softw. Eng. (1994) 528–532.
[7] M.Q. Patton, Qualitative Evaluation and Research Methods, SAGE Publications, inc, 1990.
[8] B. Kitchenham, S.L. Pfleeger, Principles of survey research (part 5): populations and samples, ACM SigSoft Softw. Eng. Notes 27 (5) (2002) 17–20.
[9] A. Jedlitschka, M. Ciolkowski, C. Denger, B. Freimut, A. Schlichting, Relevant information sources for successful technology transfer: a survey using inspections as an example, in: Proceedings of 1st International Symposium on Empirical Software Engineering and Measurement (ESEM 2007), IEEE, 2007, pp. 31–40. https://doi.org/10.1109/ESEM.2007.73
[10] M. Sayagh, N. Kerzazi, B. Adams, F. Petrillo, Software configuration engineering in practice interviews, survey, and systematic literature review, IEEE Trans. Softw. Eng. 46 (6) (2020) 646–673.
[11] P. Rodrigues, M. Souza, E. Figueiredo, Games and gamification in software engineering education: a survey with educators, in: Proceedings of IEEE Frontiers in Education Conference (FIE 2018), 2018, pp. 1–9.
[12] F.Q.B. da Silva, C. França, C.V.C. de Magalhães, R.E.S. Santos, Preliminary findings about the nature of work in software engineering: an exploratory survey, in: Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2016, ACM, 2016. https://doi.org/10.1145/2961111.2962625
[13] P. Chakraborty, R. Shahriyar, A. Iqbal, A. Bosu, Understanding the software development practices of blockchain projects: a survey, in: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2018, ACM, 2018. https://doi.org/10.1145/3239235.3240298
[14] M. Leotta, F. Ricca, M. Ribaudo, G. Reggio, E. Astesiano, T. Vernazza, An exploratory survey on SOA knowledge, adoption and trend in the Italian industry, in: Proceedings of 14th International Symposium on Web Systems Evolution (WSE 2012), IEEE, 2012, pp. 21–30. https://doi.org/10.1109/WSE.2012.6320528
[15] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering, Springer Science & Business Media, 2012.
[16] B. Mishra, A. Kertesz, The use of MQTT in M2M and IoT systems: a survey, IEEE Access 8 (2020) 201071–201086. https://doi.org/10.1109/ACCESS.2020.3035849
[17] M. Fahmideh, A. Ahmad, A. Behnaz, J. Grundy, W. Susilo, Software engineering for internet of things: the Practitioners' perspective, IEEE Trans. Software Eng. 48 (8) (2022) 2857–2878.
[18] V.S. Prasher, S. Onu, The Internet of Things (IoT) upheaval: overcoming management challenges, J. Mod. Project Manage. 8 (2) (2022). https://doi.org/10.19255/JMPM02402
[19] J.A. Aguilar-Calderan, C. Tripp-Barba, A. Zaldavar-Colado, P.A. Aguilar-Calderan, Requirements engineering for Internet of Things (IoT) software systems development: a systematic mapping study, Appl. Sci. 12 (15) (2022). https://doi.org/10.3390/app12157582
[20] T. Teik-Boon, C. Wai-Khuen, Software testing levels in internet of things (IoT) architecture, in: C. Chuan-Yu, L. Chien-Chou, L. Horng-Horng (Eds.), New Trends in Computer Technologies and Applications, Springer, 2019, pp. 385–390.
[21] M. Bures, M. Klima, V. Rechtberger, X. Bellekens, C. Tachtatzis, R. Atkinson, B.S. Ahmed, Interoperability and integration testing methods for IoT systems: a systematic mapping study, in: F. de Boer, A. Cerone (Eds.), Software Engineering and Formal Methods, Springer, 2020, pp. 93–112.
[22] S. Zhu, S. Yang, X. Gou, Y. Xu, T. Zhang, Y. Wan, Survey of testing methods and testbed development concerning Internet of Things, Wirel. Pers. Commun. 123 (2021). https://doi.org/10.1007/s11277-021-09124-5
[23] S. Kumar, P. Tiwari, M. Zymbler, Internet of Things is a revolutionary approach for future technology enhancement: a review, J. Big Data 6 (1) (2019). https://doi.org/10.1186/s40537-019-0268-2
[24] S. Balaji, K. Nathani, R. Santhakumar, IoT Technology, applications and challenges: a contemporary survey, Wirel. Pers. Commun. 108 (2019). https://doi.org/10.1007/s11277-019-06407-w
[25] M. Madeja, J. Porubän, S. Chodarev, M. Sulír, F. Gurbáľ, Empirical study of test case and test framework presence in public projects on GitHub, Appl. Sci. 11 (16) (2021) 7250. https://doi.org/10.3390/app11167250
[26] T. Kowatsch, W. Maass, IoT-I Deliverable - D2.4: Social Acceptance and Impact Evaluation, 2012, Available at https://www.alexandria.unisg.ch/entities/publication/356483cd-733f-451e-bd5b-3086130235c9.
[27] M.A. Akbar, T. Kamal, A.M. Baddour, Identification of privacy and security risks of internet of Things: an empirical investigation, Rev. Comput. Eng. Res. 6 (1) (2019) 35–44.
[28] I. Psychoula, D. Singh, L. Chen, F. Chen, A. Holzinger, H. Ning, Users' privacy concerns in IoT based applications, in: 2018 SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI, 2018, pp. 1887–1894. https://doi.org/10.1109/SmartWorld.2018.00317
[29] V.S. Prashe, Internet of Things (IoT) and Changing Face of Project Management, Master's thesis, Harrisburg University of Science and Technology, Harrisburg (PA) USA, 2018.
[30] R.C. Motta, K.M. de Oliveira, G.H. Travassos, On challenges in engineering IoT software systems, in: Proceedings of the XXXII Brazilian Symposium on Software Engineering (SBES '18), ACM, 2018. https://doi.org/10.1145/3266237.3266263
[31] G. Uddin, F. Sabir, Y.G. Gueheneuc, O. Alam, F. Khomh, An empirical study of IoT topics in IoT developer discussions on Stack Overflow, Empir. Softw. Eng. 26 (6) (2021). https://doi.org/10.1007/s10664-021-10021-5
[32] A. Mohab, K. Foutse, Y. Soumaya, What do practitioners discuss about IoT and industry 4.0 related technologies? Characterization and identification of IoT and industry 4.0 categories in Stack Overflow discussions, Internet Things 14 (2021). https://doi.org/10.1016/j.iot.2021.100364
[33] M.J. Hornos, M. Quinde, Development methodologies for IoT-based systems: challenges and research directions, J. Reliab. Intell. Environ. 14 (2024). https://doi.org/10.1007/s40860-024-00229-9
[34] S. Manasha, H.M. Ayaz, A systematic security assessment and review of Internet of Things in the context of authentication, Comput. Secur. 125 (2023) 103053. https://doi.org/10.1016/j.cose.2022.103053
[35] Z.A.A.M. Fneish, M. El-Hajj, K. Samrouth, Survey on IoT multi-factor authentication protocols: a systematic literature review, in: 2023 11th International Symposium on Digital Forensics and Security (ISDFS), 2023, pp. 1–7. https://doi.org/10.1109/ISDFS58141.2023.10131870
[36] F. Zeng, C. Pang, H. Tang, Sensors on the Internet of Things systems for urban disaster management: a systematic literature review, Sensors 23 (17) (2023). https://doi.org/10.3390/s23177475
[37] Arm, The Future of IoT: Survey Results, 2023, (Arm Website). Interactive.arm.com/story/iot-runs-on-arm/page/1.