A Multi-Method Study of Internet of Things Systems Testing in Industry

Jean Baptiste Minani [1], Fatima Sabir [2], Naouel Moha [3], and Yann-Gaël Guéhéneuc [4]

[1,4]Concordia University, Montreal, QC, CAN
Email: {jeanbaptiste.minani, yann-gael.gueheneuc}@concordia.ca
[2]University of the Punjab, Lahore, PK
Email: fatima.sabir@pucit.edu.pk
[3]École de Technologie Supérieure, Montreal, QC, CAN
Email: Naouel.Moha@etsmtl.ca

***Abstract*—As the Internet of Things (IoT) grows, its failures may have dramatic consequences on the lives of people who depend on it. Yet, it is hard to test IoT systems before they are deployed. Several researchers have provided state-of-the-art approaches for testing IoT systems. However, many of those approaches are based on academia rather than industry. Therefore, we conducted a multi-method study of IoT systems testing in the industry with IoT practitioners. We used three methods: ❶ an industry survey, ❷ practitioners interviews, and ❸ analysis of Eclipse IoT surveys. This study focuses on testing IoT systems by industry practitioners. The findings show that ❶ testing focuses more on the device, network, and application layers. IoT testing gives more importance to integration testing than acceptance testing. Test coverage is the most important metric, but metrics may vary depending on the project. ❷ IoT system testing mainly uses the model-based approach and is often manual or semi-automated, with low adoption of white box testing. Node-RED is commonly used in testing IoT systems, while Amazon AWS IoT is popular for cloud platform testing of IoT devices. ❸ Log analysis is the main approach to analyzing the root cause of bugs. ❹ The main challenges in IoT testing include the lack of standards, security, connectivity, and reference architecture. Generating test cases and establishing a standard test approach are recommended for further research. This study's findings can help IoT practitioners and researchers to identify and tackle challenges in IoT system testing, leading to future research opportunities.**

***Index Terms*—IoT System Testing, IoT Testing Challenges, IoT Protocols, IoT Platforms.**

I. Introduction

THE Internet of Things (IoT) is a network of interconnected physical devices exchanging data over the Internet [1]. The number of Internet-connected devices is increasing exponentially, and according to the Cisco report, it is expected to reach 500 billion devices by 2030 [2]. With this number of heterogeneous and distributed devices, which have the computing power and Internet-connected, IoT is shifting the computing capability from traditional devices, such as server or laptop computers towards ubiquitous computing.

With IoT evolving day after day, IoT is going to affect our personal lives and public safety directly [3]. As the number of IoT systems increases, the probability of failure becomes higher because of connectivity and scalability challenges. Therefore, ensuring that IoT systems are tested is of paramount importance because many IoT systems are safety-critical [1].

As with traditional software systems, IoT systems must be tested systematically to ensure their reliability. The acceptance criteria of the quality of service for any IoT system are required for each layer of the IoT architecture. A bug that can occur at any layer, from the device layer to the application layer, can cause the loss of millions of dollars and sometimes the loss of lives [4].

However, testing IoT systems is difficult [5, 6, 7, 8, 9, 10, 11] due to the heterogeneity and distributed nature of IoT systems. Lack of reference architecture for IoT systems, lack of standards, diverse protocols involved, security and privacy issues, coupled with insufficient test automation makes them difficult to test [12]. IoT systems have their own characteristics, such as connectivity, scalability, complexity, and heterogeneous architecture, which makes them difficult to test [13] as compared to the traditional systems that have multiple testing methods and tools to choose [14]. The testing requirements of IoT systems are different from those of traditional systems, and therefore existing approaches for traditional software testing may not be applied to IoT systems testing. Testing of IoT systems requires customized testing tools and approaches [15].

Deploying any system without conducting proper testing can have many consequences. Some of the known examples include the crash of the Bloomberg terminal in London in April 2015, due to a software defect that affected more than 300,000 traders on financial markets and caused about 3 billion pounds of debt for the government [16]. Another known example is the case of Nissan, when the software in the airbag sensor detectors failed, causing two accidents that prompted Nissan to recall over 1 million cars from the market [17]. Another example is the software fault that caused F-35 to incorrectly detect targets in formation in 2012 due to sensors failure [18].

Ariane 5 launch is widely acknowledged as one of the most expensive software failures, causing the loss of approximately 370 million dollars [19]. The cause of this failure was an undiscovered bug in the rocket's inertial reference system. The extensive reviews and tests carried out during the Ariane 5 development, did not include adequate analysis and testing of the inertial reference system, or the complete flight control system, which could have detected the potential failure.

To minimize the probability of IoT systems failure, exhaustive testing must be conducted.

Despite several studies carried out to provide state-of-the-art approaches for IoT systems testing, academic literature does not provide enough evidence from the perspective of the industry. While researchers focus more on the theoretical foundation of IoT systems testing, practitioners bring hands-on expertise. Their different perspectives complement each other and help bridge the gap between theory and practice [20]. We identified only a few studies from practitioners that focused either on IoT systems testing challenges or on IoT testing methods. Some of those studies were conducted some years ago. However, the field of IoT is dynamic. The technology, methodologies, and challenges associated with IoT systems testing are evolving. Therefore, older studies may not capture the latest advancements, emerging trends, and current practices in the field. The main challenges of IoT practitioners when testing IoT systems can provide invaluable orientation for new research. This study fills the gap that exists in the academic literature by exploring the perspectives of IoT practitioners on the challenges, tools, approaches, and artifacts related to IoT systems testing in real-world settings. By gathering this information, we provide information on the actual needs and experiences of industry practitioners.

Consequently, we study the approaches and tools to test IoT systems in the industry. We focus on testing approaches, tested quality attributes, tools that practitioners use to test IoT systems, and the main challenges faced by practitioners when testing IoT systems. We conduct a multi-method study of IoT systems testing in the industry. To achieve our objective, we use three methods to obtain facts from IoT practitioners; (1) an industry survey with 49 practitioners about testing IoT systems, (2) interviews with 9 practitioners about testing decisions,(3) data analysis of four surveys conducted by EclipseIoT[1] with IoT systems developers.

We define research questions (RQs) for each method.

- **RQ$_s$**: Primary survey related RQs.
- **RQ$_i$**: Interviews related RQs.
- **RQ$_e$**: EclipseIoT surveys related RQs.

To ensure the quality of our RQs, we follow **SPIDER** [21], a framework for formulating qualitative or mixed methods research questions (RQs) focusing on *samples*. We also followed best practices, in particular those provided by [22, 23].

[1]Open source community for IoT. https://iot.eclipse.org/

In the survey with IoT practitioners, we answer the following RQs:

1) RQ$_s$1: What are the tools and approaches used by practitioners for IoT systems testing in the industry?
2) RQ$_s$2: What are the quality attributes considered when testing IoT systems by practitioners?
3) RQ$_s$3: What are the artifacts recommended by practitioners for IoT test automation?
4) RQ$_s$4: What are the challenges faced by IoT practitioners when testing IoT systems?

We also conduct interviews with IoT practitioners, and we answer three RQs:

1) RQ$_i$1: How do testers choose the right testing approaches, levels, coverage, and metrics?
2) RQ$_i$2: How can the research community contribute to overcoming testing challenges in the IoT?
3) RQ$_i$3: How are testing artifacts automation prioritized in IoT systems?

We analyze data from surveys conducted by EclipseIoT to answer the following three RQs:

1) RQ$_e$1: What are the practitioners' top challenges when testing IoT systems?
2) RQ$_e$2: What are the top IoT communication protocols and technologies?
3) RQ$_e$3: What are the top IoT cloud platforms available to the testers?

By answering these RQs, we can recommend to both practitioners and academia, the available testing tools, testing approaches, testing metrics, test coverage, testing artifacts, IoT quality attributes, tested layers, and testing challenges for IoT systems. We present a summary of our findings from four different perspectives. ❶ Testing focuses more on the device, network, and application layer. Integration testing is the most considered testing level, whereas acceptance testing is the least considered. Test coverage is the top metric for IoT system testing, and the choice of metrics varies based on the project. ❷ Model-based approach is popular for IoT system testing. IoT system testing is still manual or semi-automated, whereas the adoption of white box testing is low. Node-RED is the most used tool in testing IoT systems, while AWS [2]IoT is a popular cloud platform for testing IoT devices. ❸ Log analysis is the main approach to analyze the root cause of bugs. ❹ Top challenges in IoT systems testing include lack of standards, security, connectivity, and lack of reference architecture. Test case generation and standard approach for IoT systems testing are the top-recommended research focus. The main contributions of this study are:

1) Identify the main challenges faced by IoT practitioners when testing IoT systems to guide future research.
2) Compile the top-quality attributes considered by practitioners for IoT systems.
3) Identify testing tools, testing approaches, testing metrics, test coverage, testing levels, and most tested layers in IoT systems by practitioners.

[2]https://aws.amazon.com

4) Provide top artifacts to consider for IoT test automation.
5) Summarize top IoT protocols, technologies, and IoT middleware.
6) Discussed lessons learned to guide future work.

The rest of this paper is organized as follows: Section II discusses related work. Section III describes the multi-method study. Section IV presents findings and answers to our RQs, while Section V reports our observations based on the analyzed data. Section VI presents possible threats that could affect the validation of our answers. Section VII concludes with future work. Finally, Section VIII summarizes the lessons learned.

II. Related Work

Several studies summarized the state-of-the-art of IoT systems testing. One of the studies [24] focused on the different testing methods, the development of testbeds, and the challenges faced while implementing IoT systems testing. The study highlighted the importance of testing IoT systems in different domains like healthcare, smart homes, and smart cities before deployment. IoT systems challenges are discussed by the researchers on many aspects of IoT systems development and testing [25, 26]. A study [27] discussed the potential use of automation and debugging tools for IoT systems. The researchers in [28] also suggested some solutions that may solve development and testing challenges.

Another study [29] surveyed different types of IoT systems testing and challenges from academia and industry. However, the above-mentioned studies discussed challenges based on the academic literature and did not consider these challenges from the perspective of practitioners. Some recent surveys [30, 31] focused on several key technologies and issues that need to be further studied in the development of IoT technologies, applications, standards, and security. They further showed that security and privacy challenges need to be studied exhaustively for IoT systems. The quality of the IoT systems is also affected by the lack of standardized reference architecture [14]. The main findings of the above studies concluded that IoT testing is lagging behind in adopting the best practices and lessons learned from the software engineering community in the past decades.

Other researchers focused on testing techniques and approach for interoperability and integration testing of IoT systems [32]. Further, they discussed different types of testing and provided the classification of available testing methods for IoT systems. Others studied continuous testing for distributed IoT systems, along with a comparison of 18 tools for the test environment [33]. Researchers in [4] summarized different quality assurance approaches for IoT systems from the literature. Their findings identified the gap in the existing techniques for IoT systems quality assurance for specific layers of IoT architecture. A study [34] summarized testing challenges for IoT systems and IoT systems quality evaluation. It concluded that the testing of IoT systems is difficult mainly because of their heterogeneity, distributed nature, and scalability. The above-mentioned studies also concluded that new approaches need to be explored for testing IoT systems.

Other studies focused on quality approaches in IoT from an academic perspective [35, 36, 37]. They focused on performance evaluation for different layers for the quality aspects like functionality, reliability, security, maintainability, and performance. They also identified the use of model-based approaches to assess at least one quality aspect of IoT systems and presented new insights and approaches for future research. They claimed that more studies are needed to explore the use of model-based approaches to assess the quality of IoT systems. Furthermore, researchers in [3] also focused on different methods and principle techniques for quality assurance in IoT systems.

The study [15] reported the main testing techniques and tools that have been considered for IoT-based systems. The study targeted the detailed comparison and analytical criticism between testing techniques and tools for the main application domains of IoT. It concluded that there is a need to identify exhaustively all testing types that have been applied to IoT, which needs to be addressed.

Another study [38] reported the lack of consolidated testing approaches. It proposed a semi-automated model-based generation of executable test cases for system-level acceptance testing of IoT systems. The study used UML models of the studied system along with additional artifacts as input and produces a test suite as output. Other studies [39, 40, 41, 42, 43, 44, 45] focused on anomaly detection of IoT systems using other concepts such as ML[3] and DL[4] techniques. They analyzed both normal and abnormal behavior of IoT components to identify anomalies or faults in IoT systems. Yet, another study [46] discussed different IoT technologies, protocols, applications, and related issues. Its aim was to provide the framework for researchers and practitioners, on how different IoT protocols work, some key issues of IoT, and the relationship between IoT and other technologies including big data and cloud computing.

There are some studies that focused on the type of testing, and testing artifacts for embedded systems [47]. Others focused on the use of UML models to generate the test cases for IoT systems [48]. They all highlighted that their findings can help industry practitioners in choosing the right testing techniques and approaches for IoT and embedded systems testing. Their findings can also help other researchers to have insights into the latest trends in IoT systems testing and identify the topics which need further investigation. Another study[49], focused on methods for test generation from input/output transition of MBT[5] of the SUT[6]. Another study [12] discussed the testing patterns in IoT and highlighted the importance of

[3]ML: Machine Learning
[4]DP: Deep Learning
[5]MBT: Model-Based Testing
[6]SUT: System Under Test

those patterns. The researchers discussed five test patterns named test periodic readings, test triggered readings, test actuators, test alerts, and test actions. They claimed that it is pertinent to implement the tool to test more patterns for IoT systems. The researchers also discussed the use of fault-tolerant technique at different layers *i.e.,* device, network, and cloud [50].

Despite the above-mentioned studies scattered across vertical silos, all of them are based on academia. As per our findings, none of them focused on IoT systems testing from industry practitioners. IoT practitioners may have different perspectives regarding current trends for IoT systems testing. The main concerns of IoT practitioners while testing IoT systems can provide invaluable orientation for future research.

In our study, we target IoT practitioners from the industry, because we did not identify any study focusing on understanding how IoT systems are tested from the perspective of practitioners.

III. Multi-Method Study Design

The objective of this study is to *analyze the current state of IoT systems testing from the viewpoint of industry practitioners and identify the key challenges affecting the testing process*. To achieve our objective, we target practitioners from the industry. We mainly target IoT systems testers or quality assurance engineers for IoT solutions. We equally target other professionals in the IoT industry, including IoT developers, IoT project managers, product owners, and maintenance engineers. We use the data from our own survey and data from the survey conducted by EclipseIoT on developers' concerns. Both the EclipseIoT survey and our own survey are important for this research. Conducted annually, the EclipseIoT survey enables us to gain a comprehensive understanding of persistent challenges since 2019. The EclipseIoT survey, with a large participant pool of approximately 600 developers, provides breadth, while our survey with 49 participants provides depth. While the EclipseIoT survey primarily focuses on developer concerns, our study takes a broader approach, encompassing testing tools, approaches, and quality attributes considered in IoT. By combining both surveys, we enhance the comprehensiveness of our research.

Fig. 1 shows the steps of our multi-method study.

We define RQs for each method based on the main objective. Table I summarizes the defined RQs.

We start with a primary survey to answer four RQs (RQ_s1, RQ_s2, RQ_s3, and RQ_s4). Based on survey answers, we conduct interviews with IoT practitioners, who willingly accepted to participate. We analyze the interview responses to find the answers to three RQs (RQ_i1, RQ_i2, and RQ_i3). We also analyze four surveys conducted by EclipseIoT from IoT practitioners between 2018 and 2022 to find answers to three RQs (RQ_e1, RQ_e2, and RQ_e3). We relate the answers from each category and draw a conclusion.

In the process of analyzing the data to find the answers to our RQs, we use the steps presented in Fig.2.

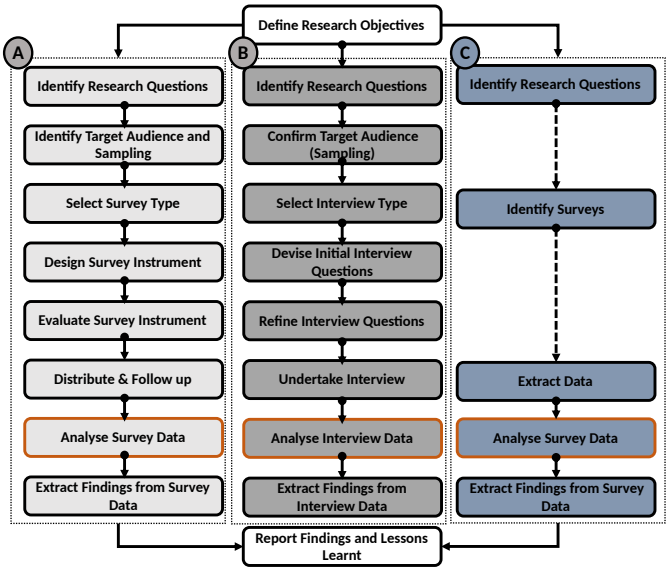


Fig. 1: Research Methodology

The rest of this section is organized as follows: Section III-A focuses on the primary survey. This section is organized as follows: Section III-A1 focuses on the target audience and sampling. Section III-A2 explains the type of survey we used. Section III-A3 describes survey design. Section III-A4 describes survey instrument evaluation. Section III-A5 focuses on how we collect and analyze answers from the survey. Section III-A6 focuses on RQ answers and conclusions.

Section III-B focuses on interviews. This section is organized as follows: Section III-B1 shows the practitioners who participated in our interviews. Section III-B2 explains the choice of the type of interview we used. Section III-B3 outlines how we devise the interview questions. Section III-B4 summarizes the refined interview questions. Section III-B5 provides details of how we conduct interviews. Section III-B6 describes how we analyze interview data. Section III-B7 focuses on the answers to interview-related RQs.

Section III-C focuses on EclipseIoT survey data analysis. This section is organized as follows: Section III-C1 shows the surveys we identified from the EclipseIoT community. Section III-C2 explains how we extract data from those surveys. Section III-C3 describes how we analyze the extracted data. Section III-C4 focuses on answers to our three RQs.

Section III-D relates the answers from primary surveys, interviews, and EclipseIoT surveys.

III-A Primary Survey

A survey is one of the empirical investigation methods which is used to collect data from a large population [51]. We aim to collect both quantitative and qualitative data from the industry to obtain evidence of the state of the art of IoT systems testing from the perspective of practitioners. We follow the steps defined in [51] as shown

TABLE I: Research Questions (RQs)

S/No	RQs	Rationale
RQ_s1	What are the tools and approaches used by practitioners for IoT systems testing in the industry?	We want to learn how IoT systems are being tested in the industry to complement what has been reported by academia.
RQ_s2	What are the quality attributes considered when testing IoT systems by practitioners?	With the lack of standards in the IoT industry, we seek more insights into quality attributes applicable to IoT systems and possible quality attributes overlooked when testing IoT systems.
RQ_s3	What are the artifacts recommended by practitioners for IoT test automation?	The recommendation from IoT practitioners will help the research community to focus on the most needed aspects of IoT test automation.
RQ_s4	What are the challenges faced by IoT practitioners when testing IoT systems?	The feedback from industry practitioners will help the research community search for solutions to real challenges.
RQ_i1	How do testers choose the right testing approaches, levels, coverage, and metrics?	We want to know how decisions are made when deciding on which approach to use, layers to test, levels to test, test coverage, and test metrics.
RQ_i2	How can the research community contribute to overcoming testing challenges in IoT?	We seek to hear from practitioners the research focuses to find solutions to their challenges.
RQ_i3	How is test artifacts' automation prioritized in IoT systems?	We want to know why practitioners recommended the automation of some test artifacts while others are overlooked.
RQ_e1	What are the practitioners' top challenges when testing IoT systems?	We want to know the top challenges of IoT practitioners for guiding the research community in understanding their needs and priorities.
RQ_e2	What are the top IoT communication protocols and technologies?	We seek more insights on the most popular communication protocols and technologies applicable to IoT systems, which could impact the testing process.
RQ_e3	What are the top IoT cloud platforms available for testers?	While conducting interviews, some practitioners mentioned that they test IoT solutions using IoT cloud platforms (also known as IoT middleware). We seek more insights on publicly available IoT middleware which could be used for testing IoT systems.

① RQ_s: Primary survey related Research Questions
② RQ_i: Interview Related Research Questions
③ RQ_e: EclipseIoT survey related Research Questions

in block A of Fig. 1. Surveys are an appropriate empirical strategy to gather data from IoT industry practitioners (e.g., about methods, tools, techniques, approaches, challenges) and to extract insights into the state of the art from the participants [52]. The scope of the survey includes various aspects of IoT systems testing from testing tools, approaches, quality attributes, testing levels, testing layers, test automation status, and IoT architecture tested. The rest of this section explains other steps of the survey process.

III-A1 Identify Target Audience and Sampling

We want to reach out to many practitioners as possible. We target people who are involved in IoT systems testing. Depending on the company organization, those professionals may be in different positions or job titles. We use judgment sampling to select the sample through the guidance of an expert [51], and we target the followings roles:

1) IoT systems developers;
2) IoT systems testers;
3) IoT systems quality assurance engineers;
4) IoT project managers;
5) IoT product owners.

We include in our form the option for the participant to write any other position or role not mentioned in the list.

III-A2 Select Survey Type

Several types of surveys exist based on deployment methods.

- **Online Surveys**: A survey method that uses the internet to collect data, typically using a web-based survey platform.
- **Paper or Mail Surveys**: A traditional survey method, where participants receive a printed questionnaire and return it by mail or in person.
- **Telephone Surveys**: Participants are interviewed by telephone, either by a live interviewer or using an automated system.
- **In-person Surveys**: Interviewers visit participants at their locations to collect data.

Each of these survey methods has its own strengths and weaknesses, and the best method for a particular study depends on various factors, including the RQs, the target population, and the resources available. An online survey is becoming more popular, especially when it is used together with social media. We want to use this type of survey to reach out to many professionals.

III-A3 Design Survey Instrument

Designing a survey instrument refers to the process of creating a survey questionnaire or form to collect quantitative and qualitative data from participants. Table II shows the questions we used in our survey. Our survey consists of

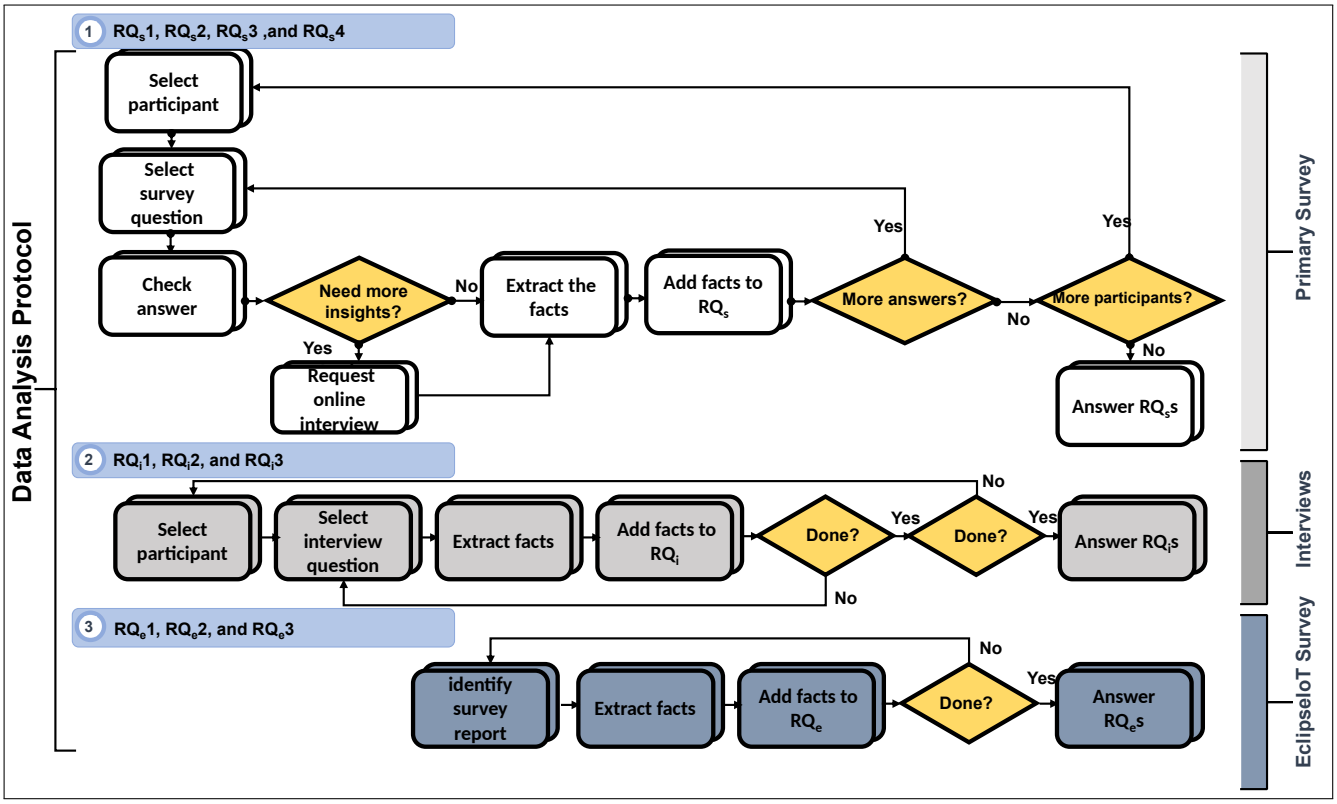


Fig. 2: Data Analysis Process

three main sections. The first two sections consist of close-ended questions, whereas the third section includes open-ended questions. The following are the three main sections of our survey questionnaire:

1) Practitioners' background information (Quantitative data).
2) IoT testing facts from the provided options (Quantitative data).
3) Understanding of practitioners in IoT systems (Qualitative data).

We want to collect information about the practitioners, including the address of affiliated organizations. We also capture job titles or positions, and years of experience in IoT. Regarding IoT systems testing facts, we ask about the tools used, top challenges, quality attributes tested, metrics used, etc. We provide different options for the practitioners to select from when answering those questions. The last section targets the understanding of practitioners in IoT systems testing.

The survey form can be found here.

III-A4 Evaluate Survey Instrument

After designing the survey instrument, we evaluate it to find if there are any flaws. To validate our questionnaire, a preliminary evaluation is conducted by all team members of this paper. The purpose is to check the completeness and understandability of the questionnaire and determine how to distribute the questionnaire. During survey pretesting, we share the survey questionnaire with 3 IoT experts to test it before sending it to other participants.

III-A5 Collect and Analyze Survey Data

We delete any data provided during survey pretesting and send the questionnaire to several professionals. We start data collection on July 1st, 2022. We finish data collection on September 30, 2022. We use Google Forms to design and generate the invitation link for the practitioners. We start with technology companies that have IoT projects, and we identify their practitioners and ask them to participate. We identify those companies through online websites including Facebook, LinkedIn, and Twitter. From those websites, we get the contacts of practitioners and the companies they work for. We contact the companies to get more practitioners to participate. We also consider different IoT research labs practitioners, and IoT practitioners from some alumni groups to participate. We follow the steps in the first part of Fig. 2 and Algorithm 1 to extract data from the responses provided by the participants. We record the extracted data in Excel. To statistically analyze the collected data, we use Excel to count different types of feedback (responses) in the survey, calculate percentages of the different responses, and generate tables or visualizations of the calculated results.

TABLE II: Survey Questions

No	Survey Question
A. Demographic Information	
1	What is your job position? [Choose the best position that fits you]
2	How many years have you been working in IoT Systems?
3	What is the address of the institution or company where you work?
4	What are the top two biggest challenges you are currently facing while testing IoT systems?
B. IoT Testing Facts	
5	On which IoT layers (device, network, cloud, or application) of your focus when testing your IoT systems
6	How do you test your IoT systems?
7	On which test level of your focus when testing an IoT system?
8	What is the architecture (based on a number of layers) of the IoT systems you tested?
9	What are the top two test artifacts that you recommend for automation for IoT system testing?
10	What are the types of test coverage (Line of code, branch, requirements, etc.) of your focus while testing an IoT system, and why?
11	What are the quality attributes do you test before deploying an IoT system?
C. Understanding IoT Testing	
12	What is your approach when you want to test a new IoT system, and why?
13	What are the tools that you use for testing IoT systems, and for which purpose?
14	Do you use any frameworks/tools (if any) to conduct End-to-End automated tests for IoT systems? If yes, give the name of the framework.
15	What are the metrics (if any) that you use while testing an IoT system, and why?
16	What are the open-source tools do you use to track the root cause of the problem when an IoT system has an error or a bug?
17	Has your testing activity been affected by the lack of standards, reference architecture for IoT systems, or the use of different communication protocols? If yes, what was the impact?
18	What are the tools previously used in testing the embedded system that are still used to test the IoT systems?
19	Did you ever perform white-box testing for an IoT System? If yes, how did you perform it, and on which layer?
20	Did you use a model-based approach while testing IoT systems? If yes, which tools did you use?

① Group A and B offer practitioners a selection of options, along with the flexibility to incorporate their own answers.
② Group C consists of open-ended questions, allowing for unrestricted responses

III-A6 Extract Findings

We use the extracted facts to answer and draw the conclusion for the following RQs: RQ_s1, RQ_s2, RQ_s3, and RQ_s4. The answers to these RQs are presented in Section IV-A.

Algorithm 1: Data Analysis Algorithm

```
Data: K → Survey Answers; N → Participants;
Result: RQ_S Answers; Interview Participants;
N = 49, K = 20;
for p in N do
    for q in K do
        if q is not clear then
            Request online interview;
            if Interview is confirmed then
                Update Interview Participants;
            end
        end
        Extract Facts from q;
        Add Facts to Appropriate RQ_S;
    end
end
RQ_S Answers;
```

III-B Interviews

An interview is defined as an "*interchange in which one person attempts to elicit information or expressions of opinion or belief from another person or persons*" [53]. We conduct online interviews with practitioners who agreed to provide more details about their survey answers. The initial purpose of the interviews was to get clarification and possibly resolve contradictions among survey answers. However, based on the responses we received, we ended up defining three additional RQs. we use the interviews to get clarification on survey answers, but also to answer the following additional RQs: RQ_i1, RQ_i2, and RQ_i3. We follow basic steps derived from [54] as indicated in block B of Fig. 1. With the research objective and RQs explained, the rest of this section explains the remaining steps in an interview study.

III-B1 Sampling

Table III shows the details of the participants whom we interviewed. Nine practitioners among the 49 agreed to participate in an online interview.

TABLE III: Interview Participants

No	Profession	Years	Country
P1	IoT project manager	12	Lithuania
P2	IoT project manager	12	USA
P3	Senior IoT solution developer	8	UAE
P4	IoT project manager	7	South Korea
P5	IoT product owner	6	USA
P6	IoT solution developer	5	USA
P7	IoT solution developer	4	Italy
P8	IoT solution developer	4	German
P9	IoT solution developer	3	Pakistan

① **Years**: Years of experience in IoT

III-B2 Select Interview Type

Interviews come in various types, depending on the number of participants and the means to interview them. For example, one type of interview is a one-on-one, in-person interview while another is the panel, video interview [53]. We use a structured interview, which is based on a fixed set of pre-determined questions [54], in order to gather information. We define open-ended questions to give the respondent the opportunity to explore the questions from multiple perspectives, and this allows us to gather a variety of information about the research subject. We combine the use of one-on-one and video interviews.

1) **One-on-One Interview**: This is the most common type of interview, where the interviewee meets with the interviewer to answer questions and provide information.
2) **Video Interview**: A video interview is a type of interview that is conducted remotely, usually through a video conferencing platform such as Zoom, Skype, or Google Meet.

III-B3 Devise Initial Interview Questions

We design the interview questionnaire and test it with the authors of this paper to check the time it takes to complete. We also check the language suitability, and potential sources of bias to verify if it produces enough relevant data to answer the RQs. We use the feedback from this pilot interview, to improve our interview design.

III-B4 Refine Interviews

We use the feedback from the previous task to improve our interview design. Table IV shows the questions included in our interview design. Our interview design consists of eight open questions. All the questions are designed based on the answers received from the survey.

TABLE IV: Interview Questions

No	Interview Question
1	What can be done to overcome testing challenges?
2	Why some test artifacts must be automated ahead of others?
3	Why do you consider some test coverage and not others?
4	What levels, and layers, the quality attributes are tested?
5	What testing approach do you use? Why?
6	How are test metrics selected for user acceptance?
7	How are bugs identified in IoT systems?
8	What impact is caused by the lack of standards, reference architecture, and multiple communication protocols?

III-B5 Interview

We conduct an online interview with each participant. With their consent, we record the answers, and we later analyze them to find the answers to the concerned RQs.

III-B6 Analyze Interview Data

We follow the steps in the middle part of Fig. 2 and Algorithm 2 to analyze interview data and extract the facts for RQs.

Algorithm 2: Interview Data Analysis Algorithm

```
Data: K → Participants;
      Q → InterviewQuestions;
      RQ_I → ResearchQuestions;
Result: RQ_I Answers;
K = 9, Q = 8;
for i in K do
    for j in Q do
        Extract Facts from j;
        Add Facts to Appropriate RQ_I;
    end
end
RQ_I Answers;
```

III-B7 Extract Findings

We use the extracted facts to answer the following RQs: RQ_i1, RQ_i2, and RQ_i3. The answers to those RQs are presented in Section IV-B.

III-C EclipseIoT Surveys

During our interview sessions, we learn that the EclipseIoT Working group publishes IoT developers' surveys every year since 2018. The purpose of each survey is to provide essential insights into IoT and edge computing industry landscapes. Those surveys present several findings including IoT developers' concerns, IoT communication protocols, IoT security technologies, connectivity protocols, IoT middleware, IoT programming languages, edge computing artifacts for IoT, IoT development tools, open-source databases for IoT, IoT strategies, and key industry focus areas. We use the insights from those interviews to find answers to the following RQs: RQ_e1, RQ_e2, and RQ_e3. Both research objectives and RQs are already defined. We focus on the remaining steps of the C block in Fig. 1.

III-C1 Identify Surveys

We identify four different surveys published by the EclipseIoT Working group in 2019 [55], 2020 [56], 2021 [57], and 2022 [58], with ***1717, 1652, 662, and 910*** participants respectively.

III-C2 Extract Data

We use an Excel spreadsheet to extract data manually from each of those four surveys. We double-check the extracted data to ensure its correctness. Table XIV in the Appendix section shows the template used for data extraction.

III-C3 Data Analysis

We follow the steps in the last part of Fig. 2 and Algorithm 3 to extract facts from survey data to answer the following RQs: RQ_e1, RQ_e2, and RQ_e3.

Algorithm 3: EclipseIoT Survey Data Analysis

Data: **N** $\rightarrow$ *EclipseIoT Survey*;
RQ_E $\rightarrow$ Research Questions;
Result: *RQ_E Answers*;
$N = 4$, $RQ_E = 3$;
for *i* ***in*** *N* **do**
  Extract Facts **f** *from* **i**;
  Add **f** *to Appropriate* RQ_E;
end
RQ_E Answers;

III-C4 Extract Findings

We use the extracted facts to answer RQ_e1, RQ_e2, and RQ_e3 and draw some conclusions. The detailed answer for each of those research questions is presented in Section IV-C.

III-D Relate Answers

We relate the answers to our RQs from each method, and we draw some conclusions based on our findings. We also relate the findings of this multi-method study with the findings in the existing literature. The relationship between some of our answers and corresponding conclusions is presented in Section IV-D.

IV. Results

We analyze the extracted data to find insights to answer our research questions. In the next sub-sections, we present the answers to our research questions for each method. It is worth noting that we conducted a literature review study, which has not yet been published. However, the data we used can be accessed. online[7]. Data and templates we used for this paper can be accessed online[8].

IV-A Primary Survey

Fig. 3 shows the demographic information of the participants. The highest number of survey participants are IoT solution developers, with 21 participants (42.9%). Regarding the professionals who are dedicated to IoT testing, we have identified only one participant (2.0%) as an IoT systems QA engineer. However, 17 participants (34.69%) did not provide their roles. Based on this finding, we can conclude that the developers of IoT systems are also involved in testing those systems. North America has the most participants, with 19 professionals (38.8%). Asia comes next with 16 participants (32.7%), followed by Europe with 5 participants (10.2%). Africa and South America have 3 and 2 participants respectively. Four participants (8.0 %) did not specify the location of their affiliated companies, since they are independent consultants and we allowed them to be anonymous if they do not wish to disclose such information. Twenty-three participants (46.9%) have between 0 and 2 years of work experience in IoT. Sixteen participants (32.7%) have between 2 and 5 years of work experience in IoT. Seven participants (14.3%) have between 5 and 10 years of experience in IoT, whereas 3 participants (6.1%) have more than 10 years of work experience in IoT. At least 10 of our participants have more than 5 years of experience in IoT, and we believe that the answers to this survey are reliable. In the next section, we present the answers to our survey research questions.

IV-A1 How Are IoT Systems Tested (RQ_s1)?

IV-A1.1 Testing Tools for IoT Systems

Testing tools are software packages that help in testing and evaluating the quality of IoT systems. These tools are designed to automate the testing process. ISO/IEC/IEEE 29119-1 describes some of the areas covered by testing tools such as test case generation, test case execution, test data generation, static analysis, test environment implementation, and maintenance [59]. We focus on the tools that are used to evaluate the quality and functionality of IoT systems. We exclude test environment tools such as testbeds, emulators, or simulators. We analyze the data provided by the participants to extract different tools available for testing IoT systems. Node-RED, a programming tool for wiring together hardware devices, APIs, and online services [60], is the most popular among the participants with 35.5%. Other tools such as Wireshark, ThingSpeak, and Apache JMeter are also popular. Selenium, which is also popular for automating web applications for testing purposes [61], is used to test IoT systems and many other tools we reported as shown in Table V.

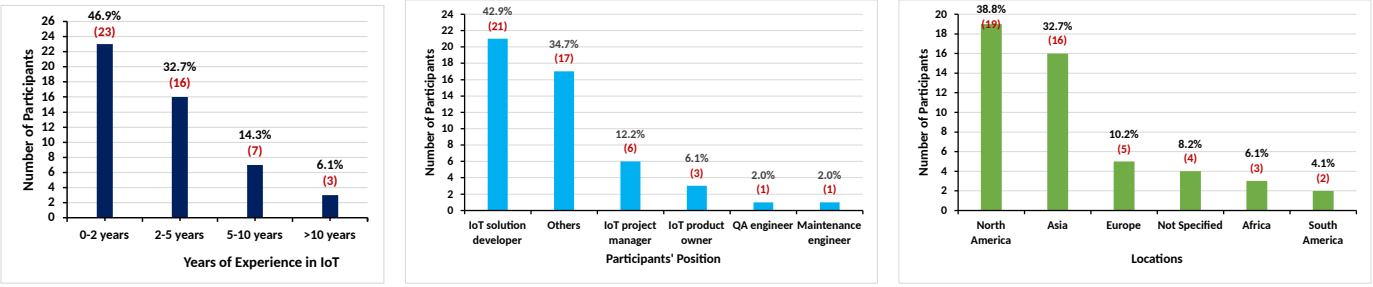
Nine participants (29.0%) indicated that they do not use any tool when testing IoT systems. They test IoT systems manually through navigation. i.e., users performing tests by entering information into a system under test and verifying the results [59].

We observe that Node-RED is popular among IoT practitioners to create and execute Node-red flow simulating IoT devices and testing the communication flow in the system infrastructure [62]. Practitioners mentioned other tools, such as JTAG Dongle, Cucumber, and Mocha, that are not identified in the literature.

IV-A1.2 Tools Applicable to Embedded Systems

Some tools used in embedded systems are also being used to test some aspects of IoT systems. Node-RED, Tessy, and Wireshark are good examples with 4.0%, 8.0%, and 4.0% respectively as highlighted in Table XV in Appendix.

[7] https://www.ptidej.net/downloads/replications/tse22b/
[8] https://www.ptidej.net/downloads/replications/iotj23a/



(a) Experience In IoT (b) Participants' Positions (c) Participants' Locations

Fig. 3: Survey Participants

TABLE V: IoT Systems Testing Tools

Tool Name	# P	% P
Node-RED	11	35.5%
Wireshark	5	16.1%
ThingSpeak	3	9.7%
Apache JMeter	2	6.5%
SmartThings	2	6.5%
Zetta	1	3.2%
Apache Kafka	1	3.2%
Selenium	1	3.2%
JTAG Dongle	1	3.2%
Gemoc	1	3.2%
Tricentis Tosca	1	3.2%
AccelQ	1	3.2%
ThingBoard	1	3.2%
Tcpdump	1	3.2%
Cucumber	1	3.2%
TestRigor	1	3.2%
JEst	1	3.2%
Mocha	1	3.2%

① **#P**: Number of participants; **%P**: Percentage of participants.

TABLE VI: IoT Testing Metrics

Metric	# P	% P
Test coverage	5	17.2%
Success rate (Pass/Fail)	4	13.8%
Response time	4	13.8%
Performance	3	10.3%
Functionality	2	6.9%
Code coverage	2	6.9%
Connectivity	2	6.9%
No of active clients	2	6.9%
Scalability	1	3.4%
Quality	1	3.4%
Reliability	1	3.4%
Packets loss	1	3.4%
Latency	1	3.4%
Periodic reading	1	3.4%
Data transmission frequency	1	3.4%
Effectiveness	1	3.4%
Efficiency	1	3.4%
Number of connected devices	1	3.4%
Customer feedback	1	3.4%
Security	1	3.4%
Accessibility	1	3.4%
No of faults	1	3.4%

① **#P**: Number of participants; **%P**: Percentage of participants.

22 participants did not provide their responses to this question, while 19 participants do not know any embedded system testing tool that can be also used for IoT systems.

We observe that Node-RED and Tessy could be used in embedded systems. Arduino IDE was not identified in the existing literature. However, the number of participants using those tools is small and we can not take any conclusion. Therefore, further studies are needed.

IV-A1.3 Metrics for IoT Systems

A metric is a standard unit of measurement that quantifies results. In the context of IoT, metrics used for evaluating or testing IoT systems are termed IoT metrics [63]. To effectively control the test processes, metrics can be used to monitor them [59]. We observed that test coverage (i.e., the number of requirements covered by executed tests [59]), is popular with 17.2%. The success rate of executed test cases (pass or failure), and response time come next with 13.8% each. Performance is the fourth with 10.3%. Functionality, code coverage, connectivity, and the number of active clients follow with 6.9% each. Many other metrics are also reported including latency, packet loss, number of faults, feedback from customers, number of connected devices, etc. as shown in Table VI.

We observe that test coverage, response time, and code coverage are also identified in the existing literature [64, 65, 66, 67]. In the literature, we identified other metrics such as the number of defects detected, mutants killed, and requests per second that are not mentioned by practitioners in this study. We have come to the conclusion that there are a lot of metrics for testing IoT systems. This means that the must-have list of metrics for IoT systems needs to be studied in depth.

IV-A1.4 Test Coverage Considered for IoT Systems

Test coverage is an indication of the degree to which the test item has been reached or "covered" by the test cases, including both the breadth and depth. It is often expressed in terms of the percentage of code tested or the percentage of requirements tested [68]. Table XVI in appendix, shows the coverage sought by the participants. We received 45 responses for the metrics, and one participant is allowed to provide more than one coverage. In the answers we received from the participants, we identify three main

coverages: functional coverage, requirements coverage, and code coverage. Code coverage is popular at 82.2% because most IoT developers test the application layer as shown by the data received on IoT layers tested. Requirements coverage is popular at 60.0%, whereas functional coverage is mentioned at 55.6%.

We conclude that only three test coverages are popular among practitioners. Other techniques, such as path coverage, branch coverage, and statements coverage, are not mentioned by practitioners. Many practitioners mentioned code coverage, but this is only applicable to the application layer since they focus more on this layer. For user acceptance test, requirements coverage and functional coverage could be considered for any IoT system.

IV-A1.5 Testing Approaches for IoT Systems

The testing approach is defined as a high-level test implementation choice, typically made as part of the test strategy design activity. Typical choices made as test approaches are test level, test type, test technique, test practice, and the form of static testing to be used [59]. IoT practitioners reported different approaches for testing IoT systems. Many of them mentioned that they do not have any systematic approach to testing IoT systems. Component-based testing is popular among the reported approaches, with 22.9%. The verification and validation approach is the next with 11.4% followed by integration, interoperability, and security with 8.6% each. Model-based, Test driven development and simulation-based approaches are also popular with 5.7% while others, such as automation, end-to-end, and low-level coding are only mentioned by 2.9% of the participants as shown in Table VII.

TABLE VII: IoT Testing Approaches

Testing Approach	% P	# P
Component-Based Approach	22.9%	8
Verification & Validation Approach	11.4%	4
Integration Based Approach	8.6%	3
Interoperability Based Approach	8.6%	3
Security Based Approach	8.6%	3
Simulation-Based Approach	5.7%	2
Test Driven Development	5.7%	2
Model-Based Approach	5.7%	2
Low level Coding Approach	2.9%	1
Automation Based Approach	2.9%	1
End-to-End Approach	2.9%	1
Usability Based Approach	2.9%	1
Scalability Based Approach	2.9%	1
Did not mention	34.3%	12

① **#P**: Number of participants; **%P**: Percentage of participants.

Twelve participants (34.3%) reported that they do not use a specific approach when testing IoT systems. They navigate through the system features, to verify if the system works as expected.

We observe that component-based is the most popular approach among the participants. We also observe that participants confuse testing approaches with testing types or targets. verification, validation, security, interoperability, integration, end-to-end, or low-level coding, are not known as testing approaches, and thus show the need for IoT testing taxonomy. In the literature, we identified more approaches such as pattern-based, mutation-based, and fault injection approaches, not mentioned by the practitioners. We can conclude that many practitioners may use some testing approaches, but they do not know the names of those approaches.

IV-A1.6 Testing Levels for IoT Systems

Testing approaches can be on one or more levels, depending on the scope of the test and its objective. So, different test levels are defined, as follows [59]: **Unit Testing**: Testing of individual hardware or software units or groups of related units [68]. It consists of isolating each part of the system and showing that individual parts fit its requirements and functionalities. **Integration Testing**: Software and/or hardware components are combined and tested to check the interaction between them and how they perform together [68]. **System Testing**: Testing a complete, integrated system to check the system's compliance and behavior within the specified requirements [68]. **Acceptance Testing**: Formal testing conducted to determine whether a system satisfies its acceptance criteria and to enable a customer, a user, or other authorized entity to determine whether to accept the system [68] Table VIII shows testing levels considered for IoT systems. As for traditional software systems, unit testing, integration testing, system testing, and acceptance testing are adopted for IoT systems [69]. Among our participants, integration testing is the most popular testing level, followed by unit, and system testing. Acceptance testing is given less consideration among the four levels of testing by practitioners.

TABLE VIII: Testing Levels in IoT Systems

Testing level	# P	% P
Integration	32	65.3%
Unit	18	36.7%
System	18	36.7%
Acceptance	8	16.3%

① **#P**: Number of participants; **%P**: Percentage of participants.

We observe that integration testing is the most popular testing level, followed by unit and system testing among many practitioners.

IV-A1.7 Layers for IoT Systems Testing

We can define the IoT layer as a fundamental constituent of an IoT system. Basic IoT architecture consists of 3 layers: device (also known as edge or perception or sensing or thing) layer, network (also known as transport or gateway or communication) layer, and application layer [70]. Some architectures include processing as an additional layer and a business layer.

1) **The Device layer** of IoT architecture also known as the sensing layer or perception layer includes devices, sensors, and actuators that collect data from their

surroundings and control things at the edge [71]. For example, a temperature sensor takes temperature readings inside a refrigerator.
2) **The network layer** is responsible for connecting to other smart things, network devices, and servers. Its features are also used for transmitting and processing sensor data [72].
3) **The application layer** is responsible for delivering application specific services to the user [72].
4) **The processing layer** is also known as the middleware layer. It stores, analyses, and processes huge amounts of data that come from the transport layer. It can manage and provide a diverse set of services to the lower layers. It employs many technologies such as databases, cloud computing, and big data processing modules [72]. For example, the cloud stores and processes the incoming data to generate alerts in real-time and, when possible, reduce the total amount of data stored [71].
5) **The business layer** derives information and decision-making analysis from data.

Although some authors mentioned human or user layer [73], industry practitioners did not mention it. We analyze the extracted data to understand which layers are being considered for IoT system testing. Testing IoT systems involve testing different layers, as illustrated in Table XVII in the appendix. However, we observed that the application layer is the most tested layer with 63.8%, followed by the network layer, and device layer with 55.3% and 51.1% respectively. The business layer and cloud layer are less tested layers because many IoT systems may not have those layers.

We observe that the participants focused more on the application layer, followed by network and device layers.

IV-A1.8 Architecture of IoT Systems

IoT system architectures consist of 3, 4, or 5 layers [72, 74, 70]. Table XVIII in the appendix, shows various models of IoT architecture tested by the participants. The participants mainly indicated that the systems tested consisted of 3-layer and 4-layer architecture. 5-layer architecture is less adopted by the participants.

In the literature, 3-layer architecture is popular. However, we believe that a five-layer architecture can help both scholars and practitioners to better address the complexity of IoT systems.

IV-A1.9 IoT System Testing Automation

Test cases can either be run manually by a human test executor, or they can be executed by a test automation tool [59]. In automation testing, a test is executed by a test automation tool. We analyze the data to understand to what extent IoT testing is automated. Table XIX in the appendix, shows our findings. None of the participants performed fully automated testing for IoT systems. 59.2% of the participants performed semi-automated testing, while 40.8% performed manual testing only.

We conclude that the participants do not perform fully automated tests for IoT systems. In the existing literature, we did not identify any approach or tool for automated IoT systems testing. This could be one of the research areas that need further studies.

IV-A1.10 White-Box Testing in IoT

White-box testing, which is a method of testing internal structures of IoT systems, is not popular among participants. We did not focus on black or gray box testing because we want to understand to what extent the source code and internal structure of the device are tested. Only 14.8% of participants performed white-box testing at some layers (application and device layers). 85.2% of the participants did not perform white-box testing for IoT systems.

We observe that white-box testing is considered by some practitioners for application and device layers testing. We did not identify white-box testing information in our previous literature review study.

IV-A1.11 Model-Based Testing Adoption

Model-based testing (MBT) uses models to generate test cases systematically and automatically [59]. ISO/IEC/IEEE29119-1 states that "by using MBT tool support environments, test cases can be quickly generated from the model and automatically executed. Thus, MBT can support improved testing beyond that supported by natural languages and manual execution". In the existing literature, model-based testing is a popular testing approach for IoT systems. In this study, we want to understand practitioners' perspectives. Among 25 participants who responded to this question, 12 (48.0%) used a model-based approach when testing IoT systems.

We observed that model-based testing is a popular approach among IoT practitioners.

IV-A1.12 IoT Bugs Root Cause Analysis

TABLE IX: Bugs Root Cause Analysis Tools

Tool	# Participants	% Participants
Nagios	4	14.8%
Logs analysis	2	7.4%
Manual detection	1	3.7%
Splunk	1	3.7%
Arduino	1	3.7%
Azure DevOps	1	3.7%
Gemoc	1	3.7%

① **#P**: Number of participants; **%P**: Percentage of participants.

We observe that no specific tool is popular for the participants to identify the root cause of a bug in IoT systems. Among 27 participants, some rely on logs monitoring and analysis to understand the root cause of problems or manual detection by checking each component. Participants suggested some tools used including Nagios, Splunk, Arduino, Azure DevOps, and Gemoc, as indicated in Table IX.

We do not consider the answers from 16 participants (59.3%) for this question because they mentioned SpiraTeam tool which is not used for root cause analysis but for bug tracking.

We do not observe any specific tool that is popular among the participants for identifying the root cause of bugs in IoT systems. We conclude that some tools for analyzing the root cause of bugs in IoT systems may exist, but remain unknown to practitioners. This calls for a joint effort between scholars and industry practitioners to develop more tools.

IV-A1.13 Impact of Standards, Reference Architecture, and Protocols

Table XX in the appendix, shows that the lack of standards and reference architecture affected many participants. 75.0% of the participants reported a negative impact for not having standards, reference architecture, and the use of multiple protocols in IoT systems.

We conclude that lack of standards in IoT can have a negative impact on IoT systems testing. We observed the same in the existing literature.

Summarized Answer to RQ_s1

Semi-automated testing in IoT is popular. Node-RED is the most used tool with 35.5% of the participants. The application layer is the most tested layer with 63.8% of the participants, while integration testing is the most considered testing level with 65.3% of the participants. Component-based is the most popular approach with 22.9% of the participants. Requirements coverage is the top metric considered with 17.2%, while code, requirements, and functional coverage are the most used test coverage.

IV-A2 Quality Attributes (RQ_s2)

The product quality model defined in ISO/IEC 25010 comprises eight quality attributes, as shown in Fig. 4. Quality attributes, included in Fig.4, were proposed for the traditional software product. All of them, except functional suitability and maintainability, are considered for IoT systems by the practitioners. The practitioners mentioned other attributes such as connectivity and scalability, which are not included in ISO/IEC 25010 model. We observe that participants considered 14 quality attributes, as summarized in Table X. While the participants did not specifically mention functional suitability and maintainability, we believe these qualities are also relevant and important for IoT systems.

Performance is the most popular attribute among participants (63.6%), while other attributes such as connectivity (50.0%), interoperability (40.9%), security (40.9%), integration (38.6%), availability (36.4%), and functionality (31.8 %) are also mentioned. Reliability (29.5%), scalability (22.7%), usability (20.5%), and compatibility (20.5%) are considered, while deployability (13.6%), compliance (9.1%), and robustness (6.1%) are less popular attributes in IoT systems. *We observe that all quality attributes for traditional software systems are considered for IoT systems. IoT systems can have more quality attributes for measuring the quality of their specific characteristics due to their distributivity and dynamic nature. We conclude that there is a need to study exhaustively all quality attributes of IoT systems.*

TABLE X: Quality Attributes Considered for IoT Systems Testing

Quality Attribute	# Participants	% Participants
Performance	28	63.6%
Connectivity	22	50.0%
Interoperability	18	40.9%
Security	18	40.9%
Integration	17	38.6%
Availability	16	36.4%
Functionality	14	31.8%
Reliability	13	29.5%
Scalability	10	22.7%
Usability	9	20.5%
Compatibility	9	20.5%
Deployability	6	13.6%
Compliance	4	9.1%
Robustness	3	6.8%

Summarized Answer to RQ_s2

Fourteen quality attributes from traditional systems are also considered for IoT systems. Performance (63.6%), connectivity (50.0%), interoperability (40.9%), security (40.9%), and integration (38.6%) are the most reported quality attributes by practitioners.

IV-A3 Testing Challenges Faced by Practitioners (RQ_s3)

Testing challenges can be defined as issues or problems in IoT systems testing that calls for special effort or dedication[76]. Testing IoT systems have many challenges due to their complexity. we want to understand from practitioners' point of view the main challenges that are affecting the quality assurance of IoT systems. We observe that lack of standards, limited resources, communication protocols, and lack of reference architecture are the top four challenges. Test case generation is reported by 16.3% of the participants. Lack of testing automation tools, lack of testing environment, and adoption of model-based testing are among the top challenges reported. The participants mentioned many other challenges, as highlighted in Table XI.

Many challenges reported by practitioners are also found in the literature. In the existing literature, we identified other challenges such as test coverage analysis and lack of emulators not mentioned by the participants. However, lack of standards, lack of reference architecture, diverse protocols, test case automation, and testing environment are common challenges reported by both academia and practitioners.

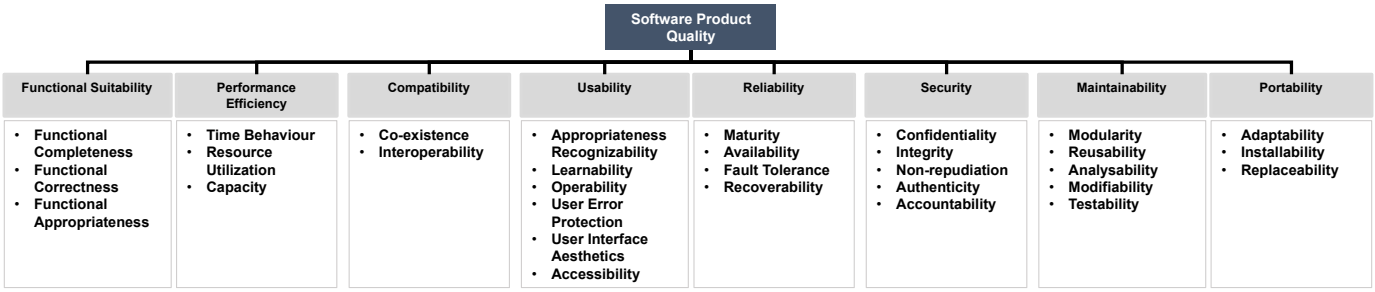


Fig. 4: ISO/IEC 25010 software product quality model [75]

TABLE XI: Challenges for IoT Systems Testing

Challenges	# P	% P
Lack of standards	14	28.6%
Limited resources	12	24.5%
Communication protocols	11	22.4%
Lack of reference architecture	8	16.3%
Test cases generation	8	16.3%
Big data	8	16.3%
Test automation tool	7	14.3%
Integration testing	7	14.3%
Testing environment	6	12.2%
Model-Based testing	6	12.2%
Connectivity issues	5	10.2%
Acceptance testing	5	10.2%
Interoperability	5	10.2%
Continuous testing	5	10.2%
Dynamic nature of IoT	5	10.2%
Hardware testing	5	10.2%
Test coverage	4	8.2%
Unit testing	4	8.2%
Testing Approach	3	6.1%
System testing	3	6.1%
Regression testing	3	6.1%
Performance testing	1	2.0%
Availability of sensors	1	2.0%
Compatibility testing	1	2.0%
Adequate semantic information	1	2.0%

① **P**: Participants.

Summarized Answer to RQ$_s$3

Twenty-five challenges are identified by the participants. Lack of standards is the most reported challenge, with 28.6% of the participants.

IV-A4 IoT System Testing artifacts Automation (RQ$_s$4)

Testing artifacts can be defined as items generated during the testing process. Good examples include test cases and test reports[77, 78, 79]. Testing artifact automation can be defined as the use of software tools or scripts to produce the test artifacts. The participants suggested different testing artifacts for automation, as summarized in Table XXI in the appendix. We observe that test cases and test data automation are the top-recommended artifacts for automation. Other artifacts mentioned by the participants include testing reports and testing strategies. In the existing literature, we also observed that test cases and test scripts are prevalent. In the literature and industry survey, both test scenarios, defect reports, and test plan automation are not mentioned.

In the existing literature review, we identified that test case generation is among the top challenges. It is also the most testing artifact discussed by many authors [80, 81, 38, 82, 83]. We believe that test case generation in IoT systems testing is still among the top challenges for many practitioners, and thus creates the need for an automated approach to generate test cases for IoT systems.

Summarized Answer to RQ$_s$4

Four testing artifacts are identified for IoT systems. Test case is the most recommended artifact for automation by many participants (90.9%).

IV-B Interviews

The questions we ask interviewees are based on the answers provided during the survey. We ask each interviewee the following questions:

- **Q1**: What can be done to overcome testing challenges?
- **Q2**: Why some test artifacts must be automated ahead of others?
- **Q3**: Why do you consider some test coverage and not others?
- **Q4**: At what levels and layers the quality attributes are tested?
- **Q5**: Which testing approach do you use? Why?
- **Q6**: How are test metrics selected for user acceptance?
- **Q7**: How are bugs identified in IoT systems?
- **Q8**: What impact has the lack of standards, reference architecture, and multiple communication protocols had?

Table XII shows the detailed facts extracted from interview data.

We used those facts to find answers to the three research questions (RQ_i1, RQ_i2, and RQ_i3).

TABLE XII: Interviews Outcomes.[1 : **Agree**. **0**: **Disagree**]

Groups	Practitioners' options	P1	P2	P3	P4	P5	P6	P7	P8	P9	Total	Percentage
Q1 Outcomes	Improve Security	0	1	1	1	1	0	1	0	0	5	**60.0%**
	Developing Standards	1	0	0	1	0	1	0	0	0	3	40.0%
	Test case automation	1	1	0	0	0	1	0	0	0	3	40.0%
	APIs	0	0	1	0	0	0	1	0	1	3	40.0%
	Testing approach	0	1	0	0	0	0	1	1	0	3	40.0%
	Heterogeneity testing	1	0	1	0	0	0	0	0	0	2	30.0%
	Testing framework	0	0	1	0	1	0	0	0	0	2	30.0%
	Testing Environment(Devices)	0	0	0	0	0	0	0	0	1	1	20.0%
Q2 Outcomes	Reduced time to test	1	1	0	1	1	1	1	1	1	8	**90.0%**
	Reduced errors	0	0	1	0	1	1	0	0	1	4	50.0%
	Improved robustness	0	1	1	0	0	0	0	0	0	2	30.0%
	Minimized repetitive efforts	0	0	0	1	0	0	0	0	0	1	20.0%
	Increased productivity	0	0	0	0	0	0	0	0	1	1	20.0%
Q3 Outcomes	Requirements check	0	1	1	1	1	0	1	1	0	6	**70.0%**
	Functionality check	0	0	1	0	0	1	1	1	1	5	**60.0%**
	Company policy	1	0	0	0	0	0	0	0	0	1	20.0%
	Project's evaluation	0	1	0	0	0	0	0	0	0	1	20.0%
	Customer expectations	0	0	0	0	1	0	0	0	0	1	20.0%
	Boost developers confidence	0	0	0	0	0	0	0	0	1	1	20.0%
Q4 Outcomes	Multiple layers and levels	1	0	1	1	1	1	0	1	0	6	**70.0%**
	System level	0	1	0	0	0	0	1	0	1	3	40.0%
Q5 Outcomes	System navigation	0	0	0	1	1	1	0	0	1	4	50.0%
	Simulation-based approach	1	1	0	1	0	0	0	0	0	3	40.0%
	Model-based approach	0	0	1	0	0	0	0	1	0	2	30.0%
	Static analysis approach	0	0	1	0	0	0	0	0	0	1	20.0%
	Adhoc approach	0	0	0	0	0	0	1	0	0	1	20.0%
Q6 Outcomes	Case by case	1	0	1	1	1	1	1	1	0	7	**80.0%**
	New metrics are needed	0	1	0	0	0	0	0	0	0	1	20.0%
	Number of defects detected	0	0	0	0	0	0	0	0	1	1	20.0%
Q7 Outcomes	Log analysis	0	0	0	1	1	1	1	1	0	5	**60.0%**
	Simulation	1	0	0	0	0	0	0	0	1	2	30.0%
	AWS IoT Device Management	0	1	1	0	0	0	0	0	0	2	30.0%
	Fault injection	1	0	0	0	0	0	0	0	0	1	20.0%
Q8 Outcomes	Communication issue	0	1	1	1	0	0	1	1	1	6	**70.0%**
	Security issue	0	0	0	1	0	0	1	1	1	4	50.0%
	No impact	0	0	0	0	1	1	0	0	0	2	30.0%
	Knowledge of all protocols	1	0	0	0	0	0	0	0	0	1	20.0%

P1: Participant 1;P2: Participant 2; etc.

IV-B1 Selection of testing approaches, levels, coverage, and metrics (RQ_i1)

IV-B1.1 Selection of Testing Approaches

During our survey data analysis, we learn that many participants responded that they do not follow any specific approach while testing IoT systems. This observation prompted us to ask those who accepted to be interviewed, to elaborate more on how they test IoT systems. 44.4% mentioned that they mainly navigate throughout the system, while 33.3% mentioned the use of simulation. They also use model-based approaches and static analysis. One participant mentioned ad-hoc testing to which there is no specific or known approach used. We realize that many practitioners may lack knowledge about different testing approaches, which could explain the absence of specific selection criteria in their responses.

Some IoT developers prefer to navigate through the real systems while testing, while others use a simulation approach. Model-based and static analysis are also preferred. However, the absence of specific selection criteria highlights the necessity to develop a comprehensive taxonomy encompassing all testing approaches, which can be widely adopted by IoT professionals.

IV-B1.2 IoT Layers and Levels for Quality Attributes Test

During the survey, we identified many quality attributes from the traditional systems that are being considered for IoT systems. We wanted to hear from the participants at what layer and levels those attributes are used. 66.6% of the participants mentioned that those attributes are used for many layers. Some examples include security, which is tested on the device, network, and application layers, whereas performance is mostly checked on the device and application layers. They also mentioned that those attributes can be tested at different levels, mainly unit testing and system testing. However, the decision to choose specific testing levels (unit, integration, system, or acceptance testing) is typically driven by project type and customer requirements, rather than specific selection criteria.

Some quality attributes such as security and performance are tested at multiple layers and at different levels of testing. Other quality attributes are checked at the system level when all components have been integrated. The choice of testing level is determined based on the nature of the project or customer requirements. We thus believe that there is a necessity to have proper guidance on quality attributes for IoT systems and how to test them.

IV-B1.3 Selection of Testing Metrics

We collected many metrics from the survey. Most of the participants mentioned two metrics only, and we wanted to know if those metrics are enough or not. 77.8% (7 participants) agree that the choice of metrics depends on the project. 11.1% (1 participant) used more than two metrics.

Requirements coverage, test case coverage, number of defects detected, performance, response time, and functional coverage are the top-recommended metrics for testing the entire IoT system. At the component level, code coverage is required for the application layer, while security is mostly needed for the device, application, and edge layers. We observe that many metrics can be preferred for one IoT system, but no specific metrics are common for every system. It is therefore necessary to have proper guidance for selecting the right metrics.

IV-B1.4 Selection of Test Coverage

From the survey data, most of the participants do not use common test coverages. We asked those who accepted to participate in our online interview to detail why they consider those test coverages and ignore the others. 66.6% (6 participants) mentioned that they want to ensure that all the requirements have been implemented and tested, while 55.5% (5 participants) highlighted the need to check the functionality of the system as the reason.

We realized that requirements verification and functionality check are the main reasons for IoT developers to decide which test coverage to use while testing IoT systems.

IV-B1.5 Root Cause Analysis

During the survey, many participants mentioned the use of SpiraTeam for managing bugs. We want to understand how developers identify the cause of the bugs. We ask the participants in our interviews to explain how they identify the cause of the bugs and the tools they use. 55.6% (5 participants) mentioned the use of log analysis, while 22.2% rely on simulation techniques. AWS IoT and fault injection are also mentioned. *The participant P6 commented that "To understand the root cause of bugs, we analyze the logs to understand the behavior of the system and identify any abnormal or unexpected events. In our logs, we record all system activities, capturing information such as errors, warnings, and other relevant data".*

Log analysis is the most frequently used approach to identify the cause of bugs in IoT systems. Simulation and fault injection techniques are also used, while other practitioners use the AWS platform for monitoring IoT systems. This observation highlights the need for more tools to help practitioners to track the root cause of bugs.

Summarized Answer to RQ_i1

Testing metrics, test coverage, testing levels, and layers to test are decided on a project basis. Most of the participants use log analysis to identify the root causes of bugs. System navigation and static analysis are approaches that are used by practitioners for testing IoT systems. However, no specific criteria for selecting those approaches.

IV-B2 Recommended Research Community Contribution (RQ_i2)

IV-B2.1 Lack of IoT Standards and Reference Architecture

In our survey, 24 participants mentioned having been impacted by the lack of standards, lack of reference architecture, and the use of many protocols. We asked the interviewed participants to explain what kind of impact they faced. 66.7% (6 participants) mentioned communication issues between different devices, especially with different message formats, while 44.4% (4 participants) also mentioned security issues because some of the IoT devices can be compromised.

The lack of standards introduces communication and security issues in IoT systems and has a negative impact on testing. This highlights the need for standards in IoT.

IV-B2.2 Actions to Overcome Testing Challenges

We ask all nine interviewees to propose research action. The interviewees are allowed to give more than one suggestion. 55% (5 interviewees) suggested security improvement in IoT, while 33% (3 interviewees) recommended focusing on the development of standards, test cases automation, API development for IoT devices, and testing approaches. Heterogeneity testing and testing framework both scored 22% (2 participants), while the testing environment got 11% (1 participant).

Security, standards, and test cases automation are among the top concerns IoT developers are still facing. APIs for smart devices and approaches for exhaustive testing are also mentioned among the concerns raised by IoT systems developers. We can conclude that security is still the main concern for many IoT developers, along with standards and test case automation.

Summarized Answer to RQ_i2

The participants proposed the research focus to overcome testing challenges. Improving security (55.0%) is the top recommendation. Standards (33.0%) and test case generation (33.0%) are also among the top recommendations from participants.

IV-B3 Testing artifacts Automation Prioritization (RQ_i3)

All survey participants proposed some testing artifacts to be automated. We requested the participants who accepted to have an interview with us, to explain their rationale when deciding on those testing artifacts. The participants were allowed to provide more than one reason. 88.8% (8 participants) mentioned a reduction of testing efforts (both time and money), while 44.4% specified the reduction of errors that can occur in manual processes.

Automation of testing artifacts such as test cases and test report results not only in reducing the efforts and resources required to test, but also reduces errors.

Summarized Answer to RQ_i3

Prioritization of testing artifacts' automation is based on expected benefits such as reducing errors in the final system, increasing test coverage, and reducing testing effort and resources.

IV-C EclipseIoT Survey

After our own survey, we also mine the insights from EclipseIoT surveys to understand the top testing challenges, communication protocols, connectivity technologies, and IoT middleware.

IV-C1 IoT Systems Testing Challenges (RQ_e1)

Fig.5a shows the developers' top challenges. Security has been the most popular concern from 2019 until 2021. Although the number of concerned developers decreases in 2022, security remains among the top three concerns. Moreover, the percentage of concerned developers on connectivity keeps growing over the years. Data collection and integration are also among the top concerns. Performance, privacy, and standards are also mentioned among other concerns [55, 56].

We believe that the continual increase in connectivity concerns highlights the challenge of finding the right technologies for IoT systems. The continual increase in integration concerns underscores the lack of APIs, and standards for many IoT devices.

Summarized Answer to RQ_e1

Connectivity and security are the top challenges in four different surveys [55, 56, 57, 58]. Data collection and integration challenges are new challenges not identified in the literature or in our primary survey.

IV-C2 Top Communication Protocols and Connectivity Technologies (RQ_e2)

IV-C2.1 Communication Protocols

From the previous literature review study, we observed that connectivity issue is among the top challenges. We also observed the continual increase of connectivity concern in this survey since 2019. To deepen our understanding of connectivity issues, we extracted and collated data on communication protocol usage. Fig. 5c presents the most widely used IoT communication protocols. MQTT is the most widely used IoT communication protocol in 2021 and 2022. We observed a constant decrease in HTTP/HTTPS usage from 2020 to 2022. TCP/IP has seen a noticeable decrease since 2019 despite a slight increase of 3% from 2021 to 2022. Websocket is mentioned only in the 2019 survey [55], but its usage is low (26%) compared to HTTP (49%), MQTT (42%), and TCP/IP (54%). No data are collected on websocket since 2020.

We observe that MQTT is currently the leading protocol for IoT communications because it tolerates intermittent connections and reduces network bandwidth needs [84].

IV-C2.2 Top IoT Connectivity Technologies

We use the connectivity data for three years (2020 [56], 2021 [57], and 2022 [58]) to understand the most widely used connectivity technologies. We did not find connectivity data in the 2019 survey [55]. Fig. 5b shows the trend in IoT connectivity technologies. We observed that the top connectivity technologies used in IoT are WiFi and Ethernet. Cellular (LTE, 4G, 5G) and Bluetooth are also considered, despite their continual decline in their usage over three consecutive years. Based on identified challenges, we can observe that it is difficult for developers to determine the best connectivity technology to use. The results from the 2020 survey show that WiFi is the most used at 44% followed by Ethernet at 39%. However, the results of the 2021 survey show that Ethernet is the most used with 45%, followed by WiFi at 40%.

We observe that WiFi and Ethernet are the top connectivity technologies used in IoT.

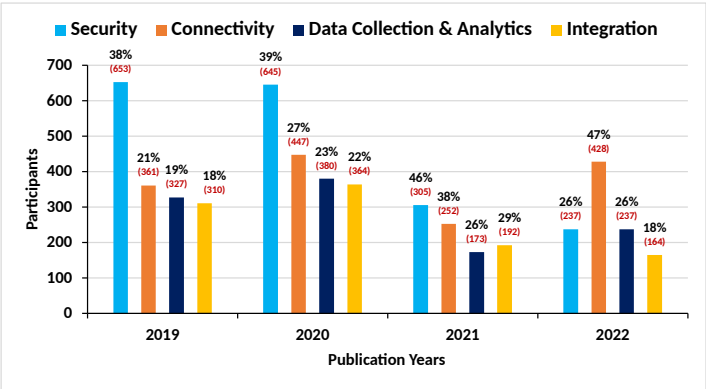
Summarized Answer to RQ_e2

MQTT is the most popular communication protocol in 2021 [57] and 2022 [58]. Since 2020 [56], HTTP and MQTT are the top two leading protocols. WiFi and Ethernet are the two top technologies used for connection in all three surveys [56, 57, 58].

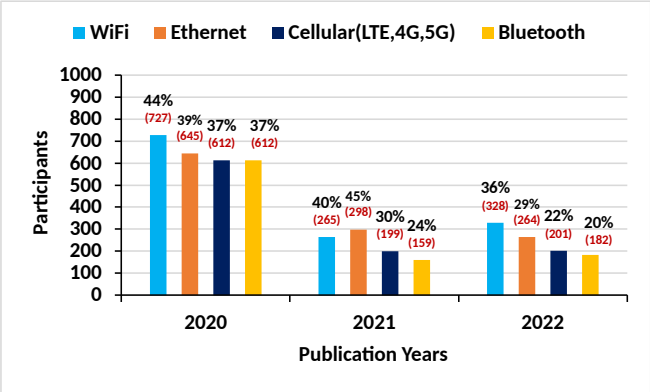
IV-C3 Top IoT Cloud Platforms (RQ_e3)

In our interviews, interviewees mentioned the use of IoT cloud platforms for testing, monitoring, and troubleshooting IoT systems. In EclipseIoT surveys, the developers provided data on the IoT platforms they used. We analyze the data to find the available IoT platforms. Fig. 5d shows the top three cloud IoT platforms. Amazon AWS IoT is the most used IoT platform, followed by Microsoft Azure IoT. Google Cloud IoT Platform is the least used among the three platforms. Other platforms such as Bosch IoT suite and IBM Watson IoT platform are also used, but their usage is low [58].

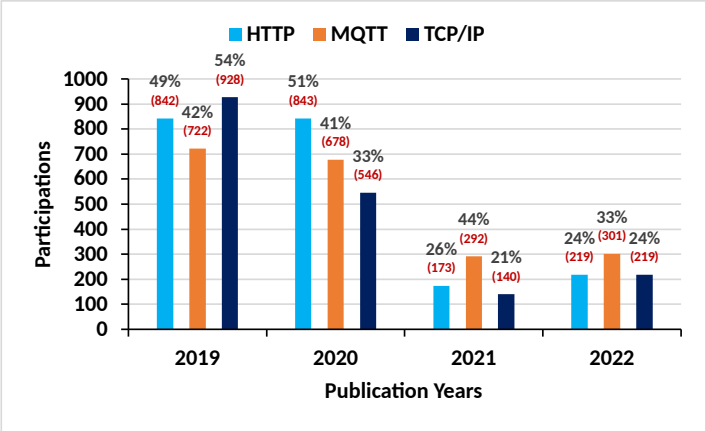
IoT platforms allow interactions among connected devices with cloud applications and other devices. They



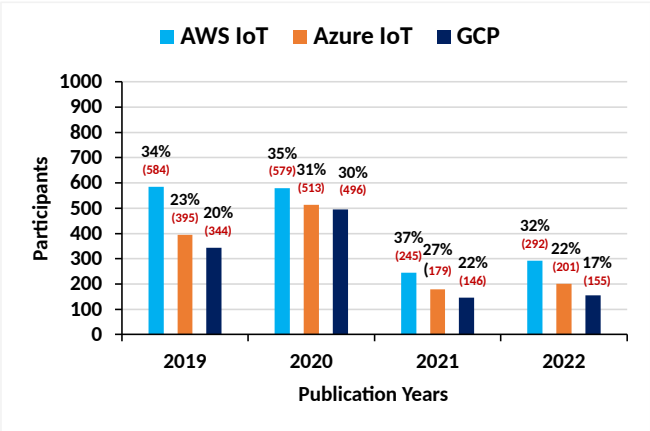
(a) Top Concerns



(b) Connectivity Protocols



(c) Communication Protocols



(d) Cloud Platforms

Fig. 5: EclipseIoT Survey Statistics

support HTTP, WebSockets, and MQTT protocols. For example, AWS IoT Device Simulator enables developers to create and simulate hundreds of virtual devices, without having to configure and manage physical devices [85].

Amazon AWS IoT is the most used IoT platform for testing, monitoring, and troubleshooting IoT devices.

Summarized Answer to RQ$_e$3

Three top IoT middleware (AWS IoT, Azure IoT, and GCP) are identified. Amazon Web Service (AWS) IoT is the most popular among the developers in four surveys [55, 56, 57, 58].

IV-D Relate Answers

We relate the answers to our research questions in this study. We also relate the answers with the existing literature. Table XIII shows the relationship between the answers within this study and the existing literature.

We conclude that the answers to research questions in this study identified common challenges, testing levels, testing layers, and testing approaches. We did not find any contradicting answers, but we observed the complementarity among those answers.

V. Discussions

We observe that there are many tools reported by academia in existing literature review that are not mentioned by practitioners such as Héctor, Izinto, InterOpT, F-interop, PatrIoT[9], Fit IoT-Lab[10], and CupCarbon[11]. Other tools are reported by the practitioners but not reported in the literature such as ThingSpeak[12], ThingBoard[13], and JEst[14]. We argue that collaboration between academia and industry can take full advantage of the tools available from both sides and hence work together to enhance them.

We also observe that the most important metrics considered for IoT systems testing such as response time, code coverage, requirements coverage, and the number of defects are the same for both practitioners and academia. We identify some metrics such as the number of mutants killed, number of requests per second, throughput, and number of defects, reported in the literature, but not mentioned by practitioners. Likewise, we identify other metrics

[9]https://patriot-framework.io/
[10]https://www.iot-lab.info/
[11]https://www.cupcarbon.com/
[12]https://thingspeak.com/
[13]https://thingsboard.io/
[14]https://www.jest.it/tag/iot/

TABLE XIII: Relationship Between the Answers

Primary Survey	**Interview**	**EclipseIoT Survey**	**Relate Answers in Current Study**	**Relate this Study with Literature review Study**
RQ_s1: Testing approaches	RQ_i1: Testing approaches selection	-	**Survey Findings**: • Component-based approach is more popular. • Model-based and test-driven development are also popular. **Interview Findings**: • System navigation and static analysis are two approaches not mentioned during the survey.	• Model-based approach is popular in literature and among practitioners. • System navigation and static analysis approaches are not mentioned in the literature review.
RQ_s2: Quality attributes	RQ_i1: Quality attributes testing	-	**Survey Findings:** • Fourteen quality attributes are mentioned for IoT systems. **Interview Findings:** • Quality attributes are tested at different layers (device layer, network layer, cloud layer, and application layer). • Quality attributes are tested at different levels (unit, integration, system, and acceptance testing).	• IoT layers to consider when testing IoT systems, different levels of testing, and quality attributes considered for IoT systems testing are identified in both literature and among practitioners. • This study shows that the quality attributes can be tested at different layers of IoT and at different levels of testing.
RQ_s3: Testing artefacts automation	RQ_i3: Automation prioritisation	-	**Survey Findings**: • Test case, test data, test report, and test strategy are testing artifacts identified. • Many practitioners recommended automation of test case generation. **Interview Findings**: • Reduction of time to test, reduction of errors, improved robustness, minimization of repetitive efforts, and increased productivity are top criteria for automation prioritization.	• Test case automation challenge is common in the literature and among practitioners. • We argue that test case generation for IoT systems deserves exhaustive study to address the needs of many practitioners.
RQ_s4: Testing challenges	RQ_i2: Overcoming challenges	RQ_e1: Practitioners' challenges	**Survey Findings**: • Lack of standards, limited resources, communication protocols, lack of reference architecture, and test case generation are top identified challenges. **Interview Findings**: • Improving security, development of standards, and test case generation are top recommended areas of focus for the research community. **EclipseIoT Surveys Findings**: • Security, connectivity, data collection, and integration are the top challenges reported by the developers. • Lack of standards can have a negative impact on security, connectivity, and integration.	• Lack of standards, lack of reference architecture, security testing, connectivity issues, testing approaches, and test case generation are common challenges in the literature and in this study. • Deployment, monitoring, debugging, heterogeneity test, and privacy testing are challenges identified in the literature but not mentioned in this study.

such as the number of active clients, number of connected devices, and packet loss, mentioned by practitioners but not mentioned in the literature.

We further notice that practitioners often mix up testing metrics with quality attributes. Performance, scalability, reliability, security, and accessibility are all mentioned among the metrics. However, the practitioners need the metric to assess those quality attributes instead of using them as metrics. Exhaustive studies on IoT testing metrics can help both industry and academia to choose the right metrics to consider when testing IoT systems.

Testing levels such as unit testing, integration testing, system, and acceptance testing are covered by both academia and industry. Integration and unit testing are reported as the most considered testing levels by both academia and industry. Based on our previous review of the literature [86, 87], system testing is the least considered testing level by scholars, whereas acceptance testing is the least considered among practitioners. We believe that acceptance testing in the industry could be the most important test to ensure that the system is checked against its requirement specifications across all the layers.

Both industry and academia suggested that the test case generation process is the most recommended task for automation. Test case automation can reduce the time required to test, reduce errors in the system, increase test coverage, improve robustness, and minimize repetitive efforts when testing. We believe that there is a need for exhaustive studies of test case automation in IoT systems for end-to-end testing.

From both academia and industry, challenges are reported while testing IoT systems. Many challenges such as deployment, debugging, monitoring, heterogeneity test, and reusability test are reported by academia but not mentioned by practitioners. New challenges such as *data collection* and *integration challenges* are identified in this study but are not identified in the existing literature.

Usability and availability are two quality attributes in

IoT systems reported by practitioners, but not identified in the existing literature addressing IoT systems testing. However, all other quality attributes mentioned in the literature [3], are also mentioned by the practitioners. We argue that the list of identified quality attributes is not exhaustive, and therefore, a further study on IoT quality attributes could provide better guidance to both practitioners and researchers.

Lastly, testing approaches, test-driven development, simulation, requirements, and model-based testing are commonly mentioned by both researchers and practitioners. Some approaches such as fault injection, fuzzy testing, fault tree analysis, pattern-based, mutation-based, and combinatorial testing are identified in the literature but are not mentioned by practitioners. Static analysis and system navigation approaches are mentioned by the practitioners, but not identified in the literature. We believe that an exhaustive study on IoT systems testing approaches is needed to find all possible testing approaches that can be used for testing IoT systems.

VI. Threats to validity

This section summarizes internal, external, and conclusion threats to the validity of this study.

Internal Validity: During the survey design, we defined twenty questions and grouped them into three sections. We designed the initial survey and conducted an internal review based on our research questions. We used internal review feedback to improve the survey design before sending it out to the practitioners. Despite conducting an internal review, it is possible that some options are omitted unwillingly from the list of possible answers to some questions. To minimize this threat, we provided an option for the practitioners to provide their own answers if not listed among the provided options.

On the distribution channels to reach out to the practitioners, we tried our best to reach as many practitioners as possible. Initially, we visited the websites of vendors of IoT solutions and device providers to find information about practitioners. We also reached out to IoT research labs and requested them to distribute our survey to the practitioners connected to those labs. However, this was limited only to a few research labs. To minimize the threat of missing some practitioners, we used alumni groups of some universities. In this case, we used four alumni groups namely: Carnegie Mellon University, USA; Concordia University, Canada; Vellore Institute of Technology, India; and COMSATS University Islamabad, Pakistan. We also used three social media websites: Facebook, Twitter, and LinkedIn to get more practitioners based on their profiles. This helped us to get more IoT practitioners that we could not identify by other means.

Regarding the position of participants, we were aware that some of the IoT companies may not have specific roles for IoT systems testing. To minimize this threat, we targeted more job titles that we believe at some point are involved in IoT systems testing.

We knew that participants may not have enough experience and have biased answers due to the lack of knowledge. We mitigated this threat in three ways: ❶ We did not provide any incentives for practitioners to participate, ❷ We allowed the participants to skip the question if they do not have enough knowledge to answer that question and, ❸ We conducted interviews with the participants who agreed to be contacted.

We analyzed the data manually using Excel for clear data validation. There is a possibility to have some errors in our analysis. Therefore, we conducted different review sessions among team members to check the accuracy of the analysis.

External Validity: We collected 49 responses from the survey participants, which might not be a good representative sample. Therefore, the results are not mineralizable. However, we believe that this is acceptable because getting answers from many IoT practitioners is not an easy task. Because, IoT is an emerging domain in its embryonic stage, some of those involved may not accept to participate in surveys and interviews. To the best of our knowledge, this is the most we could achieve in terms of the participants.

We observed more participants in the EclipseIoT survey, but the information provided could not be used for all of our research questions. We believe that our sample is still reliable that helps to provide results based on the answer of 10 practitioners having more than 5 years of experience in IoT systems. Moreover, we also have a big sample size for EclipseIoT surveys, with at least 600 participants per year.

Conclusion Validity: The information from our study may show some threats to the validity of our conclusion because either some facts are not provided in the participant's responses or the participants gave contradicting or ambiguous answers. However, this threat is acceptable because we use the interviews for the purpose of mitigating and discussing the answers of participants.

VII. Conclusion

Several research works proposed state-of-the-art approaches for IoT systems testing. However, there is still a need to study IoT systems testing from the perspective of industry practitioners. We presented in this paper, a multi-method study of IoT systems testing in industry with IoT practitioners. We used three methods: ❶ a survey with 49 practitioners about testing IoT systems, ❷ interviews with 9 practitioners about testing decisions, and ❸ an analysis of the data from the four surveys conducted by EclipseIoT with IoT systems developers.

We defined four research questions pertaining to IoT systems testing for the survey, three for the interviews, and three for EclipseIoT surveys. We analyzed the extracted data to find answers to the research questions. We provided the following contributions:

1) Identified the main challenges faced by IoT practitioners when testing IoT systems to guide future research.
2) Compiled the top quality attributes considered for IoT systems.
3) Identified testing tools available, testing approaches considered, testing metrics selected, test coverage used, testing levels considered, and the most tested layers in IoT systems.
4) Provided top artifacts to consider for IoT system testing automation.
5) We summarized top IoT protocols, technologies, and IoT middleware.
6) Discussed lessons learned to guide future work.

We reported that ❶ testing focuses more on the device, network, and application layer. Integration testing is the most considered testing level, whereas acceptance testing is the least considered. Test coverage is the top metric for IoT system testing, and the choice of metrics varies based on the project. ❷ Model-based approach is popular for IoT system testing. IoT system testing is still manual or semi-automated, whereas the adoption of white box testing is low. Node-RED is the most used tool in testing IoT systems, while Amazon AWS IoT is a popular cloud platform for testing IoT devices. ❸ Log analysis is the main approach to analyzing the root cause of bugs. Log analysis provides valuable insights into a system's behavior and allows for identifying abnormal or unexpected events. Logs are records of system activities, capturing information such as errors, warnings, and other data. By analyzing these logs, developers can trace the sequence of events leading to a bug, identify the specific conditions or interactions that triggered it, and pinpoint the underlying causes [88]. ❹ Top challenges in IoT systems testing include lack of standards, security, connectivity, and lack of reference architecture. Test case generation and standard approach for IoT systems testing are the top-recommended research focus.

Our work is beneficial for both IoT practitioners and the research community. It provides state-of-the-art of IoT systems testing by covering the testing tools, testing approaches, testing levels, quality attributes, testing metrics, test coverage, testing artifacts, IoT layers tested, IoT protocols, IoT middleware, and testing challenges. It also provides insights for future research.

In our future work, we will focus on a model-based approach for testing IoT systems. We will conduct experiments on semi-automated approaches for end-to-end IoT systems testing.

VIII. Lessons Learnt

Throughout the course of our study, we learned several lessons that can guide and inform future studies. ❶ Our analysis reveals a notable disconnect between the industry and academia. While various testing tools and approaches have been mentioned in the literature, it is surprising to note that many practitioners who participated in our survey were not familiar with them. This underscores the need for a good collaboration between industry and academia. Such collaboration can enable industry practitioners to adopt the most effective practices recommended by scholars, who can better address the needs of the industry. ❷ There are several developer tools proposed in the existing literature, such as testbeds, emulators, and simulators. However, the availability of testing tools for end-users remains scarce. Moreover, the existing tools for end-users address a few testing aspects, such as performance, connectivity, or interoperability. Our findings underscore the pressing need for more tools to test IoT systems, as well as an improvement for the existing tools. ❸ Despite the availability of various testing approaches, it is surprising to note that many practitioners do not adhere to any approach when testing IoT systems. This may indicate that the existing approaches do not entirely fulfill their requirements. We believe that the development of effective testing approaches is necessary to cater to the needs of industry practitioners. ❹ While scholars and practitioners proposed various quality attributes and metrics, our study has revealed some quality attributes that may be relevant to IoT systems not commonly known by practitioners. There is a need for a comprehensive study on quality attributes and metrics that can guide both practitioners and scholars in IoT systems testing. ❺ Practitioners and scholars may use testing concepts interchangeably, which can lead to confusion due to different interpretations. For instance, testing metrics, such as reliability or availability, are sometimes referred to as testing types by some scholars and practitioners. IoT testing taxonomy can facilitate the use of shared terminologies among both practitioners and scholars.

Acknowledgment

We thank the IoT professionals who participated in this study. We also thank the researchers whose research works contributed to this study. We equally thank Carnegie Mellon University (USA), Vellore Institute of Technology (India), and COMSATS University Islamabad (Pakistan) for their support to reach out to IoT practitioners. Finally, we thank the professors who accepted to distribute the survey to IoT professionals through their IoT research labs.

This work was funded by the Canada Research Chair program.

References

[1] M. Leotta, F. Ricca, D. Clerissi, D. Ancona, G. Delzanno, M. Ribaudo, and L. Franceschini, "Towards an acceptance testing approach for internet of things systems," in *ICWE Workshops*, pp. 125–138, Springer International Publishing, 2017.

[2] J. P. Dias, H. S. Ferreira, and T. B. Sousa, "Testing and deployment patterns for the internet-of-things," in *Proceedings of the 24th European Conference on Pattern Languages of Programs*, pp. 1–8, 2019.

[3] B. S. Ahmed, M. Bures, K. Frajtak, and T. Cerny, "Aspects of quality in internet of things (iot) solutions: A systematic mapping study," *IEEE Access*, vol. 7, pp. 13758–13780, 2019.

[4] G. White, V. Nallur, and S. Clarke, "Quality of service approaches in iot: A systematic mapping," *Journal of Systems and Software*, vol. 132, pp. 186–203, 2017.

[5] M. Bettayeb, O. Abu Waraga, M. Abu Talib, Q. Nasir, and O. Einea, "Iot testbed security: Smart socket and smart thermostat," in *2019 IEEE Conference On Application, Information And Network Security(AINS)*, pp. 18–23, IEEE; IEEE Comp Soc Malaysia, 2019. IEEE Conference on Application, Information and Network Security (AINS), Penang, Malaysia, NOV 19-21, 2019.

[6] N. Varghese and R. Sinha, "Can commercial testing automation tools work for iot? a case study of selenium and node-red," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4519–4524, IEEE, 2020.

[7] J. Beilharz, P. Wiesner, A. Boockmeyer, L. Pirl, D. Friedenberger, F. Brokhausen, I. Behnke, A. Polze, and L. Thamsen, "Continuously testing distributed iot systems: An overview of the state of the art," in *Service-Oriented Computing–ICSOC 2021 Workshops: AIOps, STRAPS, AI-PA and Satellite Events, Dubai, United Arab Emirates, November 22–25, 2021, Proceedings*, pp. 336–350, Springer, 2022.

[8] S. Bosmans, S. Mercelis, J. Denil, and P. Hellinckx, "Testing iot systems using a hybrid simulation based testing approach," *Computing*, vol. 101, pp. 857–872, JUL 2019.

[9] A. Savidis, Y. Valsamakis, and D. Linaritis, "Simulated iot runtime with virtual smart devices: Debugging and testing end-user automations," in *Proceedings Of The 17th International Conference On Web Information Systems And Technologies (WEBIST)* (F. Mayo, M. Marchiori, and J. Filipe, eds.), pp. 145–155, 2021.

[10] P. M. Pontes, B. Lima, and J. P. Faria, "Test patterns for iot," in *Proceedings of the 9th ACM SIGSOFT international workshop on automating TEST case design, selection, and evaluation*, pp. 63–66, 2018.

[11] H. Kim, A. Ahmad, J. Hwang, H. Baqa, F. Le Gall, M. A. R. Ortega, and J. Song, "Iot-taas: Towards a prospective iot testing framework," *IEEE ACCESS*, vol. 6, pp. 15480–15493, 2018.

[12] P. M. Pontes, B. Lima, and J. P. Faria, "Izinto: A pattern-based iot testing framework," in *Companion Proceedings for the ISSTA/ECOOP 2018 Workshops*, pp. 125–131, 2018.

[13] A. K. Gomez and S. Bajaj, "Challenges of testing complex internet of things (iot) devices and systems," in *2019 11th international conference on knowledge and systems engineering (KSE)*, pp. 1–4, IEEE, 2019.

[14] J. Dias, F. Couto, A. Paiva, and H. Ferreira, "A brief overview of existing tools for testing the internet-of-things," in *2018 IEEE international conference on software testing, verification and validation workshops (ICSTW)*, pp. 104–109, IEEE, 2018.

[15] N. Medhat, S. Moussa, N. Badr, and M. F. Tolba, "Testing techniques in iot-based systems," in *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 394–401, 2019.

[16] K. Julia and W. Graeme, "Bloomberg it meltdown leaves financial world in the dark." https://www.theguardian.com/business/2015/apr/17/uk-halts-bond-sale-bloomberg-terminals-crash-worldwide. Accessed: 2022-11-01.

[17] Y. Kageyama, "Toyota, nissan recall over 6 million cars due to airbag issues." https://www.ctvnews.ca/business/toyota-nissan-recall-over-6-million-cars-due-to-airbag-issues-1.2371169. Accessed: 2022-10-25.

[18] K. Osborn, "Software glitch causes f-35 to incorrectly detect targets in formation." https://www.military.com/defensetech/2015/03/24/software-glitch-causes-f-35-to-incorrectly-detect-targets-in-formation. Accessed: 2022-10-20.

[19] L. Jean-Marie and B. Alain, "Ariane 5 failure." https://www.esa.int/Newsroom/Press_Releases/Ariane_501_-_Presentation_of_Inquiry_Board_report. Accessed: 2022-12-26.

[20] V. Garousi, M. Felderer, M. Kuhrmann, and K. Herkiloğlu, "What industry wants from academia in software testing? hearing practitioners' opinions," in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, pp. 65–69, 2017.

[21] A. Cooke, D. Smith, and A. Booth, "Beyond pico: the spider tool for qualitative evidence synthesis," *Qualitative health research*, vol. 22, no. 10, pp. 1435–1443, 2012.

[22] "What makes a good qualitative research question?." https://www.bl.uk/business-and-ip-centre/articles/what-makes-a-good-qualitative-research-question. [Accessed 02-Jun-2023].

[23] S. McCombes, "Writing strong research questions." https://www.scribbr.com/research-process/research-questions/, 2022. [Accessed 02-Jun-2023].

[24] S. Zhu, S. Yang, X. Gou, Y. Xu, T. Zhang, and Y. Wan, "Survey of testing methods and testbed development concerning internet of things," *Wireless Personal Communications*, vol. 123, no. 1, pp. 165–194, 2022.

[25] A. K. Gomez and S. Bajaj, "Challenges of testing complex internet of things (iot) devices and systems," in *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 1–4, Oct 2019.

[26] M. Bures, T. Cerny, and B. S. Ahmed, "Internet of things: Current challenges in the quality assurance and testing methods," in *International conference on information science and applications*, pp. 625–634, Springer, 2018.

[27] F. Corno, L. De Russis, and J. P. Sáenz, "On the challenges novice programmers experience in developing iot systems: A survey," *Journal of Systems and Software*, vol. 157, p. 110389, 2019.

[28] E. J. Marinissen, Y. Zorian, M. Konijnenburg, C.-T. Huang, P.-H. Hsieh, P. Cockburn, J. Delvaux, V. Rožić, B. Yang, D. Singelée, I. Verbauwhede, C. Mayor, R. van Rijsinge, and C. Reyes, "Iot: Source of test challenges," in *2016 21th IEEE European Test Symposium (ETS)*, pp. 1–10, May 2016.

[29] J. A. Fadhil and Q. I. Sarhan, "A survey on internet of things (iot) testing," in *2022 International Conference on Computer Science and Software Engineering (CSASE)*, pp. 77–83, March 2022.

[30] A. S. Syed, D. Sierra-Sosa, A. Kumar, and A. Elmaghraby, "Iot in smart cities: a survey of technologies, practices and challenges," *Smart Cities*, vol. 4, no. 2, pp. 429–475, 2021.

[31] Y. Li, Y. Guo, and S. Chen, "A survey on the development and challenges of the internet of things (iot) in china," in *2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI)*, pp. 1–5, IEEE, 2018.

[32] M. Bures, M. Klima, V. Rechtberger, X. Bellekens, C. Tachtatzis, R. Atkinson, and B. S. Ahmed, "Interoperability and integration testing methods for iot systems: A systematic mapping study," in *International conference on software engineering and formal methods*, pp. 93–112, Springer, 2020.

[33] J. Beilharz, P. Wiesner, A. Boockmeyer, L. Pirl, D. Friedenberger, F. Brokhausen, I. Behnke, A. Polze, and L. Thamsen, "Continuously testing distributed iot systems: An overview of the state of the art," in *Service-Oriented Computing–ICSOC 2021 Workshops: AIOps, STRAPS, AI-PA and Satellite Events, Dubai, United Arab Emirates, November 22–25, 2021, Proceedings*, pp. 336–350, Springer, 2022.

[34] M. Cortés, R. Saraiva, M. Souza, P. Mello, and P. Soares, "Adoption of software testing in internet of things: A systematic literature mapping," in *Proceedings of the IV Brazilian Symposium on Systematic and Automated Software Testing*, pp. 3–11, 2019.

[35] G. White, V. Nallur, and S. Clarke, "Quality of service approaches in iot: A systematic mapping," *Journal of Systems and Software*, vol. 132, pp. 186–203, 2017.

[36] A. Zebouchi and Y. Aklouf, "A survey on the quality of service and metaheuristic based resolution methods for multi-cloud iot service selection," in *Innovations In Bio-Inspired Computing And Applications, IBICA 2021* (A. Abraham, A. Madureira, A. Kaklauskas, N. Gandhi, A. Bajaj, A. Muda, D. Kriksciuniene, and J. Ferreira, eds.), vol. 419 of *Lecture Notes in Networks and Systems*, pp. 412–424, 2022.

[37] T. Ismail and I. Hamarash, "Model-based quality assessment of internet of things software applications: A systematic mapping study," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, p. 128, 06 2020.

[38] D. Olianas, M. Leotta, and F. Ricca, "An approach and a prototype tool for generating executable iot system test cases," in *International Conference on the Quality of Information and Communications Technology*, pp. 383–398, Springer, 2020.

[39] A. Gaddam, T. Wilkin, and M. Angelova, "Anomaly detection models for detecting sensor faults and outliers in the iot - a survey," in *2019 13th International Conference on Sensing Technology (ICST)*, pp. 1–6, Dec 2019.

[40] A. Sgueglia, A. Di Sorbo, C. A. Visaggio, and G. Canfora, "A systematic literature review of iot time series anomaly detection solutions," *Future Generation Computer Systems*, vol. 134, pp. 170 – 186, 2022.

[41] M. Fahim and A. Sillitti, "Anomaly detection, analysis and prediction techniques in iot environment: A systematic literature review," *IEEE Access*, vol. 7, pp. 81664–81681, 2019.

[42] J. Manokaran and G. Vairavel, "Smart anomaly detection using data-driven techniques in iot edge: A survey," *Lecture Notes in Electrical Engineering*, vol. 844, pp. 685–702, 2022.

[43] M. Alsoufi, S. Razak, M. Siraj, A. Ali, M. Nasser, and S. Abdo, "Anomaly intrusion detection systems in iot using deep learning techniques: A survey," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 72, pp. 659–675, 2021.

[44] B. Sharma, L. Sharma, and C. Lal, "Anomaly detection techniques using deep learning in iot: a survey," in *2019 International conference on computational intelligence and knowledge economy (ICCIKE)*, pp. 146–149, IEEE, 2019.

[45] A. Gaddam, T. Wilkin, M. Angelova, and J. Gaddam, "Detecting sensor faults, anomalies and outliers in the internet of things: A survey on the challenges and solutions," *Electronics (Switzerland)*, vol. 9, no. 3, 2020.

[46] S. H. Shah and I. Yaqoob, "A survey: Internet of things (iot) technologies, applications and challenges," in *2016 IEEE Smart Energy Grid Engineering (SEGE)*, pp. 381–385, IEEE, 2016.

[47] V. Garousi, M. Felderer, Ç. M. Karapıçak, and U. Yılmaz, "Testing embedded software: A survey of the literature," *Information and Software Technology*, vol. 104, pp. 14–45, 2018.

[48] N. K. Bahrin and R. Mohamad, "A systematic literature review of test case generator for embedded real-time system," *International Journal of Software Engineering and Technology*, vol. 1, no. 1, 2014.

[49] S. L. D. C. Paiva and A. D. S. Simao, "A systematic mapping study on test generation from input/output transition systems," in *2015 41st Euromicro Conference on Software Engineering and Advanced Applications*, pp. 333–340, IEEE, 2015.

[50] M. T. Moghaddam and H. Muccini, "Fault-tolerant iot: A systematic mapping study," in *Software Engineering for Resilient Systems: 11th International*

Workshop, SERENE 2019, Naples, Italy, September 17, 2019, Proceedings 11, pp. 67–84, Springer, 2019.

[51] A. N. Ghazi, K. Petersen, S. S. V. R. Reddy, and H. Nekkanti, "Survey research in software engineering: Problems and mitigation strategies," *IEEE Access*, vol. 7, pp. 24703–24718, 2018.

[52] T. Punter, M. Ciolkowski, B. Freimut, and I. John, "Conducting on-line surveys in software engineering," in *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.*, pp. 80–88, 2003.

[53] E. E. Maccoby and N. Maccoby, "The interview: A tool of social science," *Handbook of social psychology*, vol. 1, no. 1, pp. 449–487, 1954.

[54] J. C. Young, D. C. Rose, H. S. Mumby, F. Benitez-Capistros, C. J. Derrick, T. Finch, C. Garcia, C. Home, E. Marwaha, C. Morgans, *et al.*, "A methodological guide to using and reporting on interviews in conservation science research," *Methods in Ecology and Evolution*, vol. 9, no. 1, pp. 10–19, 2018.

[55] E. IoT, "Iot and edge key findings 2019." https://iot.eclipse.org/community/resources/iot-surveys/assets/iot-developer-survey-2019.pdf. Accessed: 2022-11-10.

[56] E. IoT, "Iot and edge key findings 2020." https://iot.eclipse.org/community/resources/iot-surveys/assets/iot-developer-survey-2020.pdf. Accessed: 2022-11-10.

[57] E. IoT, "Iot and edge key findings 2021." https://outreach.eclipse.foundation/iot-edge-developer-2021. Accessed: 2022-11-10.

[58] E. IoT, "Iot and edge key findings 2022." https://outreach.eclipse.foundation/iot-edge-developer-survey-2022. Accessed: 2022-11-10.

[59] I. 29119-1:2022, "Software and systems engineering - software testing - part 1: General concepts." https://www.iso.org/standard/81291.html, 2022. [Accessed 10-Feb-2023].

[60] Node-RED, "Node-red." https://nodered.org/. [Accessed 10-Feb-2023].

[61] Selenium, "Selenium." https://www.selenium.dev/. [Accessed 10-Feb-2023].

[62] H. Smidt, M. Thornton, and R. Ghorbani, "Smart application development for iot asset management using graph database modeling and high-availability web services," *Hawaii*, 2018.

[63] P. Nirpal and K. Kale, "A brief overview of software testing metrics," *International Journal on Computer Science and Engineering*, vol. 3, 01 2011.

[64] L. Hu, W. E. Wong, D. R. Kuhn, R. N. Kacker, and S. Li, "Ct-iot: a combinatorial testing-based path selection framework for effective iot testing," *Empirical Software Engineering*, vol. 27, pp. 1–38, 2022.

[65] M. Bures, B. S. Ahmed, V. Rechtberger, M. Klima, M. Trnka, M. Jaros, X. Bellekens, D. Almog, and P. Herout, "Patriot: Iot automated interoperability and integration testing framework," in *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, pp. 454–459, IEEE, 2021.

[66] M. Leotta, D. Ancona, L. Franceschini, D. Olianas, M. Ribaudo, and F. Ricca, "Towards a runtime verification approach for internet of things systems," in *Current Trends in Web Engineering: ICWE 2018 International Workshops, MATWEP, EnWot, KD-WEB, WEOD, TourismKG, Cáceres, Spain, June 5, 2018, Revised Selected Papers 18*, pp. 83–96, Springer International Publishing, 2018.

[67] H. D. Q. Cunha, "Low-code solution for iot testing," *ThinkMind*, 2019.

[68] I. C. Society, "IEEE standard for software and system test documentation," *IEEE Std 829-2008*, pp. 1–150, 2008.

[69] T.-B. Tan and W.-K. Cheng, "Software testing levels in internet of things (iot) architecture," in *New Trends in Computer Technologies and Applications* (C.-Y. Chang, C.-C. Lin, and H.-H. Lin, eds.), (Singapore), pp. 385–390, Springer Singapore, 2019.

[70] A. Makhshari and A. Mesbah, "Iot bugs and development challenges," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pp. 460–472, IEEE, 2021.

[71] R. Heredia, "4 layers of iot architecture." https://www.zipitwireless.com/blog/4-layers-of-iot-architecture-explained. [Accessed 16-Feb-2023].

[72] P. Sethi and S. R. Sarangi, "Internet of things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, 2017.

[73] M. Fahmideh, A. Ahmad, A. Behnaz, J. Grundy, and W. Susilo, "Software engineering for internet of things: The practitioners' perspective," *IEEE Transactions on Software Engineering*, vol. 48, no. 8, pp. 2857–2878, 2022.

[74] D. Navani, S. Jain, and M. S. Nehra, "The internet of things (iot): A study of architectural elements," in *2017 13th International Conference on Signal-Image Technology And Internet-Based Systems (SITIS)*, pp. 473–478, 2017.

[75] I. 25010, "Software product quality." https://iso25000.com/index.php/en/iso-25000-standards/iso-25010. [Accessed 16-Feb-2023].

[76] Z. Hassan, H. Ali, and M. Badawy, "Internet of things (iot): Definitions, challenges, and recent research directions," *International Journal of Computer Applications*, vol. 128, pp. 975–8887, 10 2015.

[77] Q. Madness, "What is a test plan and how to write one?," Oct 2021.

[78] H. K. V. Tran, M. Unterkalmsteiner, J. Börstler, and N. bin Ali, "Assessing test artifact quality—a tertiary study," *Information and Software Technology*, vol. 139, p. 106620, 2021.

[79] K. Kuldeep, "https://artoftesting.com/test-artifacts-deliverables."

[80] L. G. Guşeilă, D.-V. Bratu, and S.-A. Moraru, "Continuous testing in the development of iot applications," in *2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI)*, pp. 1–6,

IEEE, 2019.

[81] N. Medhat, S. M. Moussa, N. L. Badr, and M. F. Tolba, "A framework for continuous regression and integration testing in iot systems based on deep learning and search-based techniques," *IEEE Access*, vol. 8, pp. 215716–215726, 2020.

[82] D. Olianas, M. Leotta, and F. Ricca, "Matter: A tool for generating end-to-end iot test scripts," *Software Quality Journal*, pp. 1–35, 2021.

[83] M. N. Zafar, W. Afzal, and E. Enoiu, "Towards a workflow for model-based testing of embedded systems," in *Proceedings of the 12th International Workshop on Automating TEST Case Design, Selection, and Evaluation*, pp. 33–40, 2021.

[84] P. Tracy, "Mqtt protocol minimizes network bandwidth for the internet of things." https://www.rcrwireless.com/20161129/featured/mqtt-internet-of-things-tag31-tag99. [Accessed 18-Feb-2023].

[85] I. A. W. Services., "Iot device simulator." https://aws.amazon.com/solutions/implementations/iot-device-simulator/, 2021. [Accessed 15-Feb-2023].

[86] L. G. Guşeilă, D.-V. Bratu, and S.-A. Moraru, "Continuous testing in the development of iot applications," in *2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI)*, pp. 1–6, IEEE, 2019.

[87] C.-S. Koong, C. Shih, P.-A. Hsiung, H.-J. Lai, C.-H. Chang, W. C. Chu, N.-L. Hsueh, and C.-T. Yang, "Automatic testing environment for multi-core embedded software—atemes," *Journal of systems and software*, vol. 85, no. 1, pp. 43–60, 2012.

[88] J. Cândido, M. Aniche, and A. Deursen, "Log-based software monitoring: a systematic mapping study," *PeerJ Computer Science*, vol. 7, p. e489, 05 2021.

Appendix

A Other Tables

TABLE XIV: Data Extraction Template

ID	Item	Description
D1	Year	Year the survey was conducted
D2	Participants	Number of the survey participants
D3	Challenges	Top challenges affecting development and testing
D4	Communication Protocols	Top Communication Protocols such as HTTP, CoAP, MQTT, etc.
D5	Connectivity Protocols	Top Communication technologies such as WiFi, Bluetooth, etc.
D6	Middleware	Software that acts as an intermediary between IoT devices and applications helps to integrate and manage IoT devices and services within a connected system.

TABLE XV: Embedded Systems Testing Tools

Tools	#P	%P
Tessy	2	8.0%
Arduino IDE	2	8.0%
Node-RED debug panel	1	4.0%
Wireshark	1	4.0%
TestPlant	1	4.0%
Serial Monitor	1	4.0%

① **#P**: Number of participants; **%P**: Percentage of participants.

TABLE XVI: Test Coverage for IoT Systems

Test Coverage	# P	% P
Code coverage	37	82.2%
Requirements coverage	27	60.0%
Functional coverage	25	55.6%

① **#P**: Number of participants; **%P**: Percentage of participants.

TABLE XVII: Layers for IoT Systems Testing

Layer	# P	% P
Application Layer	30	63.8%
Network Layer	26	55.3%
Device Layer	24	51.1%
Cloud Layer	12	25.5%
Business Layer	5	10.6%

① **#P**: Number of participants; **%P**: Percentage of participants.

TABLE XVIII: IoT Architectures Tested

Architecture	# P	% P
3-Layers	25	52.1%
4-Layers	24	50.0%
5-Layers	6	12.5%

① **#P**: Number of participants; **%P**: Percentage of participants.

TABLE XIX: IoT System Testing Automation Status

Automation status	# P	% P
Semi-automated	29	59.2%
Manual	20	40.8%
Fully automated	0	0.0%

① **#P**: Number of participants; **%P**: Percentage of participants.

TABLE XX: Impact of Standards, Reference Architecture, and Protocols

Category	# P	% P
Impact	24	75.0%
No Impact	6	18.8%
Somehow	2	6.3%

① **#P**: Number of participants; **%P**: Percentage of participants.

TABLE XXI: Artifacts for Testing Automation

Artifact	# Participants	% Participants
Test case	40	90.9%
Test data	24	54.5%
Test report	14	31.8%
Test strategy	8	18.2%