



# Enhancing IoT Project Success through Agile Best Practices

**WOUTER MOEDT VAN BOLHUIS**, University of Nebraska - Omaha, College of Information Science and Technology, USA and Management Center Innsbruck, Management, Communication and IT, Austria

**REINHARD BERNSTEINER**, Management Center Innsbruck, Management, Communication and IT, Austria

**MARGERET HALL**, Wirtschaftsuniversität Wien, Institute for Cognition and Behavior, Austria

**ANN FRUHLING**, University of Nebraska - Omaha, College of Information Science and Technology, USA

Worldwide spending on Internet of Things (IoT) applications is forecasted to surpass \$1 trillion by 2022. To stay competitive in this growing technological industry segment, lowering costs while increasing productivity and shortening time-to-market will become increasingly important. Adopting Agile Software Development practices for IoT projects may provide this competitive advantage, as it enables organizations to respond to change, while being dynamic and innovative. Applying a mixed-methods approach, agile IoT practitioners around the world and from diverse industries were surveyed and interviewed. Our study recommends that Agile Software Development team makeup, practices, and methods should be tailored to the specific industry, culture, people, and IT application of an organization. People play an important role in the success of agile projects; therefore, our research focuses on identifying the critical attributes of agile teams to maximize success. Our study identified the five critical agile practices: Collective Code Ownership, Continuous Integration, Single Team, Dedicated Customer, and Sprint Planning and found that both technical and soft skills are essential for successful IoT development.

CCS Concepts: • **Software and its engineering** → **Agile Software Development; Programming teams;** • **Human-centered computing** → **Ubiquitous and mobile computing theory, concepts and paradigms;** • **General and reference** → *Surveys and overviews*;

Additional Key Words and Phrases: Agile practices, agile team skills, Internet of Things, IoT development, IoT team skills

## ACM Reference format:

Wouter Moedt van Bolhuis, Reinhard Bernsteiner, Margeret Hall, and Ann Fruhling. 2023. Enhancing IoT Project Success through Agile Best Practices. *ACM Trans. Internet Things* 4, 1, Article 5 (February 2023), 31 pages.

<https://doi.org/10.1145/3568170>

Wouter Moedt van Bolhuis, Reinhard Bernsteiner, Margeret Hall, and Ann Fruhling contributed equally to this research. Authors' addresses: W. Moedt van Bolhuis, MCI Management Center Innsbruck, Universitätsstraße 15, 6020 Innsbruck, Austria; emails: [wmoedt@unomaha.edu](mailto:wmoedt@unomaha.edu), [w.moedt@mci4me.at](mailto:w.moedt@mci4me.at); R. Bernsteiner, Management Center Innsbruck, c/o Reinhard Bernsteiner, Universitätsstrasse 15, 6020 Innsbruck, Austria; email: [reinhard.bernteiner@mci.edu](mailto:reinhard.bernteiner@mci.edu); M. Hall, Wirtschaftsuniversität Wien, Vienna University of Economics and Business, Welthandelsplatz 1, A-1020 Vienna, Austria; emails: [mahall@unomaha.edu](mailto:mahall@unomaha.edu), [margeret.hall@wu.ac.at](mailto:margeret.hall@wu.ac.at); A. Fruhling, University of Nebraska Omaha, 1110 S. 67th Street, PKI 280A, Omaha, Nebraska 68182, USA; email: [afruhling@unomaha.edu](mailto:afruhling@unomaha.edu).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored.

Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2023 Association for Computing Machinery.

2577-6207/2023/02-ART5 \$15.00

<https://doi.org/10.1145/3568170>

ACM Transactions on Internet of Things, Vol. 4, No. 1, Article 5. Publication date: February 2023.

## 1 INTRODUCTION

Decreasing development costs, improving productivity, reducing time-to-market, and boosting customer satisfaction are often advantages that can be attributed to Agile Software Development [44, 51]. Determining which factors influence the success of software development has been a major research focus for years [8, 11, 34]. To stay competitive, software development practitioners must balance the need to lower costs while increasing productivity and shortening time-to-market [13]. Early results from adopting Agile Software Development practices are supportive of these somewhat-competing goals. They report productivity improvements ranging from 15% to 23%, while reductions in time-to-market ranged from 25% to 50% [48]. New and upcoming technologies, such as **Internet of Things (IoT)**, create a demand for new research in this area. Best practices on how to implement and fine-tune Agile Software Development methods can be a driving force providing a competitive advantage and deserve significant attention especially in the still maturing IoT industry.

IoT development companies form a part of a growing technological industry segment, which has great economic potential, with worldwide spending on IoT forecasted to surpass \$1 trillion in 2022 [27]. Even considering the worldwide economic impact from the COVID-19 pandemic, remarkable growth rates occurred for global IoT spending in 2021 [28]. Early indicators projected the global number of IoT devices active endpoints to be 12.3 billion at the end of 2021 equating to a 9% growth rate [60]. Moreover, worldwide IoT spending is expected to continue double-digit growth from 2021 to 2024, while IoT is also expected to play an important role in the “next normal” after organizations have recovered [35] from the pandemic upheaval. Industry leaders anticipate there will likely be over 27 billion IoT connections by 2025 [60]. The potential benefits from adopting Agile Software practices for IoT development companies is poised to be immense.

### 1.1 Motivation for This Research

Agile Software Development is a popular approach that focuses on creating small working pieces of software in iterations [3]. One of its major benefits is that it allows customers to provide feedback and change requirements during the development phase [51]. This can lead to outcomes that better match their specifications, which can help save time and money when compared to traditional approaches, such as plan-driven development. In today’s fast-paced world, the ability to cope with continuous changes has become vital for organizations to survive, which is one of the reasons Agile Software Development has gained momentum [11, 32]. Multiple agile methodologies exist, each consisting of shared and unique practices that can be thought of as different ways of working. Organizations are not restricted to choosing just one agile methodology or adopting all of its practices; rather, they can combine practices from different methodologies [42]. This allows organizations to select those practices that suit their specific needs.

With the tremendous IoT market growth and agile software development popularity, it is important to further understand how agile practices can be leveraged in IoT development. This is important, because IoT development has certain characteristics that differ from other software development applications. For instance, IoT combines both hardware and software, whereas non-IoT applications focus mainly on software. In addition, IoT solutions are cloud-centric and often have additional security and privacy challenges. These unique requirements impact the teams developing IoT solutions.

Moreover, each organization is different. Agile Software Development team makeup, practices, and methods should be tailored to the specific industry, culture, people, and IT application of an organization [30, 42]. Whereas software development companies have widely and quickly adopted a range of Agile Software Development practices, many hardware development

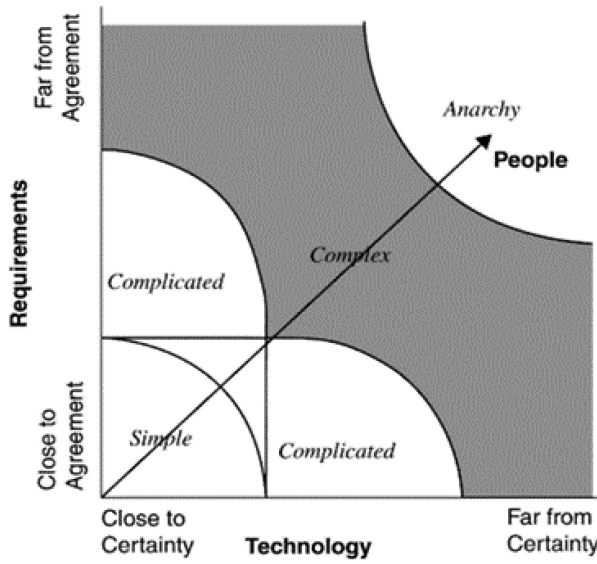


Fig. 1. Stacey matrix [54].

companies continue to work with traditional development approaches [25]. IoT systems form a category of their own, as they consist of at least a physical component (hardware), intelligent component (software), and a connectivity/communication component. The situational aspects of Agile Software Development combined with the multi-faceted technology stack of IoT present our research gap. Due to the novelty of IoT technologies and a continuous changing technology environment, there is limited research on the usage of agile methodologies for IoT development.

## 1.2 Agile Software Development

Agile Software Development is considered to be most applicable to dynamic environments as it makes use of practices that enable constant customer feedback [10]. The IoT industry is new, still maturing, and therefore, can be considered a suitable environment. The complexity matrix, also commonly referred to as the Stacey matrix, is a theoretical model supporting this premise. The Stacey matrix assesses the complexity of tasks by measuring their levels of technological certainty and agreement level on the technology's requirements, which helps select the appropriate management or leadership approach for a specific scenario [63]. Based on the assessment of these dimensions, a task fits into one of the categories in the matrix, which are *simple*, *complicated*, *complex*, and *anarchy*. The Stacey matrix, adapted for software development, includes *requirements*, *technology*, and *people* as dimensions [54], as shown in Figure 1. We foresee that a large portion of IoT development will fall into the *complex* category of the Stacey matrix.

Within a software development project, the requirements describe what has to be done, and the technology describes how and the people by whom the project is performed. Requirements can be close to agreement or far from agreement, when they are unclear or changing. The choice of technology can be close to certainty or far from certainty, when a selection has to be made from a wider range of different technologies. Projects may include only a few team members in a stable setting, but could also consist of a large and continuously changing group of people, which makes them less predictable and puts these teams into the anarchy category. In projects that are considered to be simple projects, both the requirements and the choice of technology are clear,

which makes them easier to be planned. Such projects are often repeatable and have a predictive nature, which makes traditional methodologies such as waterfall a better fit in comparison to agile methodologies [22]. Complicated projects are characterized by having more known than unknown aspects, but require expert knowledge in order to make decisions about the uncertainties. Depending on the scenario, both traditional and agile methodologies could fit, although Kanban or Scrum may be the preferred options [21]. Complex projects, on the other hand, face many more uncertainties, where both the requirements and the choice of technology are not yet clear. For example, customer requirements may not yet be fully defined, while developers may not yet know how a solution that meets the customer needs can and will be built, and the kind of challenges that may arise may be unknown. Agile methodologies are generally suitable for complex projects, but as different approaches and technologies could help to reach a successful outcome for these projects, extensive expertise and experience is required in order to make an educated decision.

Projects that fall in the anarchy category have many uncertainties and many different approaches are possible, for instance because only a rough idea exists [20, 37]. Requirements of IoT projects are likely not very stable as the market is new and continuously changing, while multiple technologies are required for the development of IoT solutions. Altogether, this places IoT development projects somewhat in the middle of the Stacey matrix, depending on the exact scenario. Since agile methodologies are suitable and even highly recommended for complex projects, this speaks in favor of using agile methodologies in IoT development. Exceptions for when agile methodologies may not be suitable include the development of life-critical systems [6] as well as scenarios where requirements are very strict and unlikely to change [36].

Since people play an important role in the success of agile projects, our research focuses on the attributes of agile teams and their team members. Agile teams within IoT development may require different skills, competencies, and agile approaches than teams that focus solely on hardware or software. Without best practices that provide IoT development teams with guidance on how to leverage Agile Software Development, they may not be as successful as they could be. This study addresses the following research questions: (1) Which agile practices, agile team skills, and IoT development team skills are perceived as most important for agile IoT development? (2) Which agile practices, agile team skills, and IoT development team skills most positively impact project success?

## 2 BACKGROUND

This section presents the theoretical background and related works relevant for the empirical part of this study. First, the specific characteristics of Agile Software Development and the required team competencies are presented. Next, the specific characteristics of IoT systems and their development are outlined, followed by indicators for measuring the success of these projects. A related works discussion concludes this section.

### 2.1 Agile Development Practices

The intention of Agile Software Development is to enable organizations to respond to change, while being dynamic and innovative [3, 10]. Agile Software Development consists of multiple methodologies that share the same characteristics, aiming to facilitate quickly changing requirements while focusing on the customer [24]. In essence, methodologies that are considered to be agile methodologies are lightweight, make use of short iterative cycles, actively involve users, and rely on the team's tacit knowledge and concise documentation. To deal with change, and more specifically with changing requirements, agile methodologies are iterative and are used to develop software incrementally [6, 53]. Multiple agile methodologies exist, which all adhere to the agile

Table 1. Distribution of Most Commonly Used Agile Practices

Agile Practice	Percentage of use
Daily Standup	90%
Sprint Planning	88%
Retrospective	85%
Sprint Review	80%
Short Iterations	69%
Release Planning	67%
Kanban Board	65%
Planning Poker	65%
Dedicated Customer	63%
Single Team	52%
Frequent Releases	51%
Common Work Area	47%
Product Roadmap	46%
Story Mapping	44%
Portfolio Planning	35%
Requirements Engineering Practice	Percentage of Use
Unit Testing	75%
Coding Standards	64%
Continuous Integration	54%
Refactoring	45%
Continuous Deployment	37%
Pair Programming	36%
Test-Driven Development	35%
Automated Acceptance Testing	32%
Collective Code Ownership	31%

principles defined in the Agile Manifesto [3] and consist of numerous mutually supportive practices that embody agile ways of working [56].

VersionOne’s, which later became AgileAI, “Annual State of Agile Report” identifies a total of 24 agile practices that are well-known and commonly used; this set of 24 agile practices also proved to be constant over a time period of 3 years [15, 16]. Table 1 lists the various practices and their respective usage among agile teams. Scrum is identified as the most commonly used agile methodology, followed by Extreme Programming, Kanban, and various hybrid versions [14]. Practitioners can also choose to employ practices from different methodologies, commonly referred to as “hybrids,” which can complement one another and may better meet their specific needs or preferences. In our study, we focus on Agile Software Development as a whole, rather than on a specific agile methodology.

## 2.2 Agile Teams and Competencies

Agile teams are cross-functional, consisting of developers, testers, and analysts who all work toward the same goal. Through supporting each other, teams are more cohesive, can reach common goals faster [69], and information is exchanged faster due to the close proximity of team members [23]. In addition, agile development is characterized by face-to-face communication as much as possible, allowing agile teams to respond to change faster [23], and typically resulting in more concise documentation. Agile teams are self-organizing where team members decide together how

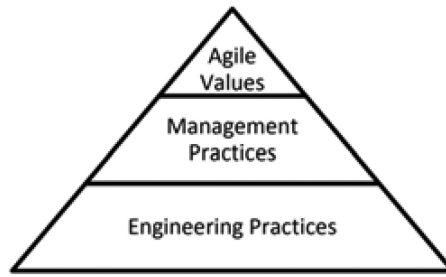


Fig. 2. Pyramid of agile competencies [32].

to distribute and manage the workload. This does not mean that agile teams are not managed; rather, management of agile teams is considered to be “less heavy.” Agile processes are designed to leverage individual strengths. Since each team is unique and has different competencies, there is no one-size-fits-all approach and agile processes and practices need to be tailored to individual teams [29, 53, 68].

The required competencies differ between teams that are using agile or traditional methodologies, and may also be between different agile methodologies [2]. Therefore, a team that is good at software development when using traditional methodologies may not necessarily be as successful in an agile environment, which makes use of iterations and incremental software delivery and focuses heavily on collaboration. When reviewing agile practices such as Test-driven Development or Collective Code Ownership, it becomes clear that team members require different skills in an agile environment [19, 32]. Creating test cases before starting development can be a shift in mindset when team members are used to testing with more information available. In addition, team members have to be comfortable with working on the same codebase with others when employing the Collective Code Ownership practice. Therefore, teams need a different mindset in order to succeed in an agile environment [64, 70].

Although software development sounds rather technical, it should be emphasized that in addition to technical skills, social and ethical skills are also essential in Agile Software Development [13]. To illustrate this, the required competencies for agile teams can be organized into (agile) engineering practices, (agile) management practices, and agile values. Engineering practices form a foundation that builds upon the software engineering competencies of an agile team, which mainly depend on the technical competencies of individual team members. This enables an agile team to develop high quality software [13]. Examples include Unit Testing, Coding Standards, Continuous Integration, Refactoring, Pair Programming, and Test-Driven Development. Other agile practices are aimed more at the management of team aspects and social skills. Examples include Daily Standup, Sprint Planning, Retrospective, Kanban Board, and Common Work Area. The agile values as defined in the Agile Manifesto are positioned at the top of the pyramid (Figure 2). Whereas individuals require hard and soft skills when it comes to the agile and engineering practices, this is not directly the case for the agile values. Thus, embracing these values may require a cultural shift for some individuals and may be more difficult to learn.

The composition of agile teams can be distinguished between four types of team members: non-experts, I-shaped experts, T-shaped experts, and C-shaped experts [49]. Non-experts do not have significant expertise in any field. People who are an expert in one field, but do not have significant expertise in any other field are called I-shaped experts. When I-shaped experts become knowledgeable about other fields too, they become T-shaped experts. T-shaped experts have advanced knowledge in a field and have more general knowledge in multiple other fields that are usually related. Finally, those with advanced knowledge in multiple fields are called C-shaped experts. For instance,



full-stack developers can be considered to be C-shaped experts, as they often have a lot of experience in multiple development fields. It is desirable that the majority of an agile team consists of T-shaped experts, because with multiple T-shaped experts the project requirements can be met, while they are also able to support each other but are less expensive to hire than C-shaped experts [49].

Since agile teams are usually relatively small and aim to leverage the strengths of individuals, it is important that the team members collectively possess all the skills that are required for the project. These skills include both hard skills and soft skills [46, 47, 57, 70]. Hard skills are required to perform tasks and consist of both technical knowledge and practical experience. Soft skills include personality traits as well as the ability to interact with others and are complementary to the required hard skills. In agile teams, soft skills are required to ensure quick and smooth communication, while hard skills are required to ensure products are delivered that meet the quality expectations of stakeholders [9]. Therefore, the composition of agile teams and the hard and soft skills they possess is of great importance to their success in IoT development.

Agile teams require proficiency in programming and sufficient knowledge of databases to ensure that a satisfactory software solution can be developed. Knowledge of agile methodologies is required to ensure that teams can deliver results quickly. The range of soft skills that team members should possess include communication skills to effectively transfer information among team members as well as facilitation skills to create a collaborative and transparent environment. Agile teams require interpersonal skills to maintain effective working relationships, people skills to work together in a positive manner, and teamwork skills to successfully accomplish common goals by combining skills and knowledge of individual members. Both analytical and thinking skills are required to understand and solve complex problems. Planning skills are required to determine which goals should be achieved and when and how resources should be utilized. Business and domain expertise of the three components of IoT systems are also important. Expertise on all three components and the knowledge on the ways to combine and integrate them is essential. This highlights the distinct difference between IoT projects and traditional hardware or software application development projects.

A challenge for agile teams can occur when experts from different domains with different mindsets need to interact, which makes soft skills such as cooperation important. Management skills are essential for organizing and motivating team members. Management leadership skills provide a sense of direction for the team [45]. Other critical soft skills include vision, negotiation, motivation, and decision-making [50]. Lastly, agile teams require knowledge of systems, software (and system) lifecycles, (system) architecture, testing, quality, security, and release management [22]. These knowledge attributes become more important as hardware development is usually executed in a different way than software application development.

The Agile Values component of the Pyramid of Agile Competencies foundation is based on the Agile Manifesto [10]. The Agile Manifesto contains the following values:

- “**Individuals and interactions** over processes and tools”
- “**Working software** over comprehensive documentation”
- “**Customer collaboration** over contract negotiation”
- “**Responding to change** over following a plan”

In their 2001 article, Cockburn and Highsmith [10] state that these four values do not imply that the items on the right are not important, but simply indicate that the items on the left are valued more. The first value therefore does not claim that processes and tools are not useful, but rather suggests that relying on individuals and interactions enables organizations to change processes and share information faster, which increases agility. Working software allows the customer to track actual progress instead of relying on plans and promises. The increased interaction between

people reduces the emphasis on documentation. Interaction between people is not limited to the team working on the development of new software. Instead, the team, customer, sponsor, and user should all collaborate. With the team, customer, and other stakeholders collaborating and working toward the same goal, outcomes will better match the specifications of the customer and user. This means that adjustments can be made over the course of a project, which would be less likely in a more bureaucratic environment where the focus is on contracts. Finally, although planning is useful, a continuously changing environment can cause plans to become outdated quickly. Therefore, deviating from a plan should be possible in order to respond to continuous changes.

### 2.3 IoT Characteristics and Competencies

IEEE Standards defines IoT as “Broadly speaking, the Internet of Things (IoT) is a system consisting of networks of sensors, actuators, and smart objects whose purpose is to interconnect “all” things, including everyday and industrial objects, in such a way as to make them intelligent, programmable, and more capable of interacting with humans and each other.” Taivalsaari and Mikkonen [26, 65] state that the majority of IoT solutions share the same underlying concept, regardless of the manufacturer. Fundamentally, all IoT solutions consist of hardware, gateways, and a cloud solution. Possible additions are applications that show visualizations, such as reports and algorithms that are used for analytics. The hardware of IoT solutions is used for data collection and for performing certain actions.

The hardware contains sensors to collect data about physical activities such as temperature or humidity, and actuators to perform the required actions. The gateways feature Wi-Fi or Bluetooth connectivity to securely transfer the collected data to the cloud. The cloud forms a fundamental part of an IoT solution and is primarily used to store the vast amount of collected sensor data. After storing the data, it can be further processed to make sense of the data. Depending on the use case, the data may be automatically analyzed in real time or afterwards. Finally, the cloud can be used as an actuator to enable remote access and the cloud solution may also take care of managing devices, users, and reports.

In order to further explain what sets IoT development apart from software application development, we assess what its unique characteristics and challenges are. A key difference is that IoT development consists of both physical hardware and software, whereas software development is solely focused on software. Since IoT includes a range of wireless communication technologies as well as different types of sensors, it is considered to be a multidisciplinary field [41, 52]. IoT systems utilize cloud technology for data collection and computations [65], which can be value-adding if the data has been filtered and analyzed [38]. IoT development can be very challenging for organizations because of requiring many different competencies. These competencies include knowledge of hardware and software, sensors and actuators, maintenance and upgrades, big data collection, and IoT business models. For smaller organizations it can be difficult to find enough skilled people [43]. Further, IoT development requires competencies in programming multiple interconnected devices, programming error handling across devices that are always switched on, as well as programming distributed software migrated to multiple devices. IoT teams require skills in embedded software development, cloud technologies, web client software and mobile software, as well as in data analytics and visualization [65, 66]. Lastly, knowledge on how to store data securely and the importance of privacy and security in IoT systems is critical [33].

### 2.4 Project Success Measurements

The design of a team can be linked to its performance. A team’s design determines the skills and knowledge needed by the team members [4]. High-performing teams achieve significant results, have high productivity and morale, and share the same values and goals. Team members are



flexible, support each other, and have open communication often leading to high group cohesion [5]. While skills and knowledge are important for teams to succeed, trust and communication have also been established as essential aspects of high-performing teams. In addition, high-performing teams are characterized by healthy rivalries among team members, clearly defined roles and responsibilities, as well as employing leadership that motivates the team [21]. Although it may seem intuitive that high-performing teams consist of some of the best and brightest members, research instead shows that the emphasis of high-performing teams is on the group dynamics and team behavior. To that extent, [40] found that not only agile practices and team skills determine a team's success, but also multiple other factors, including the team's physical setting.

In general, four dimensions used to measure success in projects are project efficiency, impact on the customer, business success, and preparing for the future [59]. Project efficiency measures if both the schedule and budget are met. Impact on the customer measures if performance, specifications, and needs are met to the customer's satisfaction. The commercial success and gained market share determine the business success. Preparing for the future assesses the future benefits such as entering a new market or creating a new product line. In line with this literature, [55] defines a set of questions that can be used to measure the success of projects. These questions include measuring if the project was on budget, on time, as well as if its scope and requirements were met. Additional questions focus on measuring the project success from the perspective of sponsors and stakeholders, the project team, client, and end users.

## 2.5 Related Works

Due to the specific characteristics of IoT developments, agile methods must be carefully selected and adopted. A survey on different agile methods revealed that Scrum is the most often used method, although it has limitations. Using Scrum makes it difficult to work with more than one team on the same project. Furthermore, Scrum is not designed for large or highly distributed projects, which might span several organizations [39].

Based on a case study, the benefits of Scrum are mapped with the requirements of an IoT project in the field of home automation. Compared to non-agile process models, the agile approach allows for better customer engagement during the entire development process. This represents an advantage, since the project requirements are not fully known at the beginning of the project. The authors argue that the fast-changing nature of IoT and related projects require a flexible and easily adjustable project management approach. These requirements go in line with agile methods [18].

A similar situation is described in the field of factory automation. Factory automation includes elements from the Industrial IoT and services built on it. Agility plays a central role in the requirements engineering phase in those projects. It is in this phase that an iterative multi-step dialogue must include the opinions and requirements of all stakeholders. The technology analysis needs adjustments to clarify and incorporate new insights from changing technologies and market developments. Agile methods with their principles are eligible to fulfill these requirements [67].

In another study, three cases from the industry were analyzed to gain insights into the application of agile methods in embedded systems development [31]. The development of embedded systems is mainly about developing hardware components, which is usually left out of agile methods. In one of the investigated cases only hardware was developed. In the two other cases, software was also developed, but with software development playing a subordinate role. The study found that some of the agile principles and practices can directly be applied to embedded system development. Other practices must be adopted or reinterpreted according to the specific nature of hardware development [1, 17, 31].

In a different study examining agile methods in an embedded system project, the authors found that agile methods offer similar advantages to those that often occur in software-only development

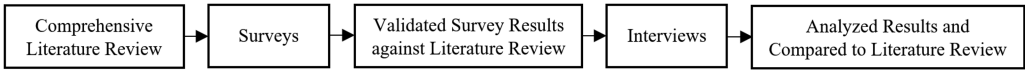


Fig. 3. Research methodology diagram.

projects. The study reported that teamwork and communication within the development team may be improved from following agile practices [31]. Two unique characteristics of embedded system project development were identified: (1) the slower nature of hardware development and (2) the different knowledge between developers. These characteristics must be addressed when agile methods are used in embedded system projects [31].

### 3 METHODOLOGY

#### 3.1 Research Approach

This study employed a mixed-methods approach using theory building, and quantitative (survey) and qualitative (interview) research methods [61]. We selected the triangulated mixed-methods approach because it often provides a more complete picture than qualitative or quantitative research can produce separately [12]. We followed nested research design, iterating between literature-based validation before and after the survey and interview phases [61]. We began our research by conducting a comprehensive literature review of the most common agile practices, IoT team expertise, and software developments skill sets used today. In addition, we examined literature on how to measure project success. This was used to inform the development of the survey mechanism.

The quantitative phase included surveying experts in agile and IoT development practices on what they perceive as the most important skill sets and expertise desired for an IoT development team. They were prompted to reflect on the skills based on the current agile/IoT project they were active in. Participants were asked to assess how successful the current IoT project was that was being developed by teams who possessed these attributes and skill sets. These results were compared to the findings from literature, and then used to create the interview questions. Interviews were conducted to provide deeper insights as to why certain agile practices, team skills, and development skills might lead to higher levels of project success. Although project success can be quantified to some degree, it was important to gain a deeper understanding of if and how agile practices might lead to success and to understand the rationale behind the choice and usage specific to agile practices. Creswell and Creswell [12] state interviews can help researchers gain a thorough understanding of a complex problem by gathering information directly from people in a certain context and from their point of view. Finally, we compare the results of the interviews to the “Annual State of Agile Report” practices and report on this in Section 4. In summary, Figure 3 illustrates the research methodology for this study.

#### 3.2 Survey Development and Design

**3.2.1 Variable Selection—Agile Practices, Team Skills, and Success Dimensions.** In order to measure the agile team skills, IoT development team skills, and project success factors, we first defined constructs for each category based on our literature review. The literature review identified 15 agile management practices and nine agile engineering practices that are most commonly used in line with [14–16]. This ensured that the list of agile practices in the survey were currently relevant practices. Based on the consolidated overview from the literature of the hard and soft skills that agile teams require, 10 agile team skills were selected for the survey. The required skills for IoT development came from the seven IoT development team skills [55]. Based on Serrador and Pinto [55], the six dimensions to measure success in projects were budget, time, scope and requirements, customer or stakeholder satisfaction, team satisfaction, and end user satisfaction.

**3.2.2 Survey Design.** Respondents received a brief explanation of each agile practice as well as brief descriptions of each agile team skill and IoT development skill. Respondents were asked to rate the importance of each agile practice. Next, respondents were asked to rate how competent they perceived their team to be in both agile team skills and IoT development team skills. Third, they were asked to rate their perception of the project's success according to the success dimensions. For all questions, participants were directed to base their responses on their most recently completed IoT project. All measures were recorded on a 5-point Likert scale (1 = lowest, 5 = highest).

For the analysis of the quantitative data, it was important to understand the background of respondents. Respondents were asked about their experience with agile practices and IoT development. In addition, respondents were asked in which industry they are active. Respondents could also select one or more Agile Software Development lifecycle phases they have experience in. To classify and distinguish between the characteristics of the projects and teams within different companies, team size, location of development, and the usage of tools were included as measurements. In addition, respondents were asked about the duration, complexity, and uncertainty of their most recently completed IoT project.

To summarize, the survey contained 24 agile practices, 10 agile team skills, 7 IoT development team skills, and 6 project success dimensions. The independent variables for this study consisted of agile practices, agile team skills, and IoT development team skills, and the dependent variables consisted of the project success dimensions. An electronically distributed survey was employed.

### 3.3 Expert Participants' Criteria

Our goal was to accurately assess the skill sets and the usage of each agile practice found in literature review within active IoT development teams. Therefore, participants who were knowledgeable on both IoT and agile development processes within their company were sought, as they were likely to have a solid understanding of agile practices that are being used and the skill sets among their team members. We defined a list of terms and skill sets related to both Agile Software Development and IoT development, which served as key criteria when searching for potential participants through professional online social networks such as LinkedIn. Participants that make use of agile practices for IoT development were recruited regardless of their country, industry, or type of organization. However, participants with experience in using agile practices for IoT development proved to be difficult to find, as IoT development is still maturing, and agile methodologies may not have been fully adopted. Therefore, additional participants were sought through snowball (chain) sampling by asking participants to refer additional eligible participants in their own networks.

### 3.4 Semi-structured Interview Design

Below, we present the questions that the interviewees were asked and how we recruited and prepared our interview participants.

**3.4.1 Interview Prompts.** The interview script consisted of eight questions (Table 2). The first question asked participants to explain their choice of agile practices that are important for their team. The second question asked participants how the agile practices have contributed to the success of their project. Subsequently, participants were asked what went well and what could be improved when reflecting on their most recently completed IoT project. Participants were also asked if their team had modified the agile practices, if they considered agile practices to be suitable for IoT development, and if they wanted to add anything to the interview themselves.

An additional dimension that was mentioned in the literature review focused on preparing for the future. This aspect was deemed to be more suitable to be discussed in the interviews, as it

Table 2. Interview Script

1.	In the survey, you have indicated which agile practices were most important for your team. From your perspective, what made these specific agile practices important for your team?
2.	When considering the last IoT project you have finished, how have these most important agile practices contributed to the following: <ul style="list-style-type: none"> <li>— Meeting project budget goals?</li> <li>— Meeting project time goals?</li> <li>— Meeting project scope and requirements?</li> <li>— The customer or stakeholder rating the success of the project?</li> <li>— The project team's satisfaction with the project?</li> <li>— The end users' satisfaction with the project's results?</li> <li>— Creating positive effects for the future, such as entering a new market or creating a new product/technology?</li> </ul>
3.	What went particularly well in the last IoT project you finished because of the agile practices you used, and why?
4.	What could have been improved in the last IoT project you finished in terms of making use of these agile practices, and why?
5.	Has your team made changes to any of the agile practices to make them more suitable for IoT development? If so, what changes have been made and why?
6.	Overall, do you feel agile practices are suitable for IoT development? Why or why not?
7.	Is there anything else you would like to add that you feel is important when using agile practices for IoT development?
8.	Do you have any connections working in the field of IoT who are also employing agile methodologies and may thus be eligible participants for this study?

may not apply to all projects and could therefore be difficult to include in the survey. Due to the innovative nature of IoT projects, this dimension is highly relevant in measuring their success. Even if a (pilot) project is not very successful, lessons learned may still contribute to improving future projects by helping to create repeatable processes.

**3.4.2 Interview Data Collection Approach.** The survey included a question at the end asking the participant if he or she would be willing to be interviewed. Those participants that agreed to be interviewed were contacted. Interview participants were sent an interview script prior to the interview which allowed them to familiarize themselves with the questions. The interview script contained a list of agile practices that participants had rated as “extremely important” or “very important” on the survey, to remind them of their selection.

## 4 RESULTS

We begin our analysis by providing a comprehensive discussion on the background of our respondents. Next, we present an analysis of the quantitative data followed by an analysis of the qualitative data from the interviews.

### 4.1 Respondent Demographics

**4.1.1 Survey Respondent Demographics.** From a total of 74 participants who started the survey, 45 participants completed all survey questions. Twenty participants dropped out without

Table 3. Survey Respondent Demographics (N = 45)

Gender	Frequency	Country of Residence	Frequency
Male	35 (77.8%)	United States of America	12 (26.7%)
Female	7 (15.6%)	Germany	10 (22.2%)
Other	0 (0.0%)	The Netherlands	6 (13.3%)
Prefer not to say	3 (6.7%)	Austria	4 (8.9%)
<b>Age</b>		India	3 (6.7%)
20–29 years	10 (22.2%)	France	2 (4.4%)
30–39 years	12 (26.7%)	Australia	1 (2.2%)
40–49 years	15 (33.3%)	Israel	1 (2.2%)
50–59 years	6 (13.3%)	Morocco	1 (2.2%)
60–69 years	2 (4.4%)	Poland	1 (2.2%)
<b>Level of Education</b>		Romania	1 (2.2%)
Less than high school degree	0 (0.0%)	Serbia	1 (2.2%)
High school graduate or equivalent	0 (0.0%)	Slovenia	1 (2.2%)
Some college but no degree	2 (4.4%)	United Kingdom	1 (2.2%)
Associate degree in college (2-year)	0 (0.0%)		
Bachelor's degree in college (3/4-year)	20 (44.4%)		
Master's degree or similar	21 (46.7%)		
Doctoral degree (PhD)	2 (4.4%)		
Professional degree (JD, MD)	0 (0.0%)		

answering any question, which may indicate they did not possess the required knowledge to answer the first questions related to Agile Software Development. Seven participants dropped out after answering the first questions about their experience with Agile Software Development and IoT development. Although some may have possessed the required knowledge, they may have found the survey too lengthy, which on average took almost 14 minutes to complete. The majority (77.8%) of respondents are male and many age groups are well-represented. Respondents were from 14 different countries; the majority of respondents are from the United States (26.7%), followed by Germany (22.2%), The Netherlands (13.3%), and Austria (8.9%). See Table 3 for additional demographics.

As shown in Table 4, over half (64.5%) of the respondents considered themselves to be at least very knowledgeable about Agile Software Development. A variety of industries were represented, including transportation, healthcare, residential/smart buildings, and agricultural and energy industries. The majority (51.1%) of respondents had between 1 and 3 years of experience in IoT development, while none of the respondents had more than 10 years of IoT experience, which is not surprising since IoT is a newer technology.

Most respondents worked in relatively small teams of under nine team members. This is common for agile teams [58, 62]. Teams mostly completed their work on-site or in a distributed manner, which corresponds with the agile principles of working closely together. The vast majority of IoT projects lasted under 12 months, although some took between 12 and 24 months. Finally, the majority of respondents considered their last IoT projects to be moderately complex and uncertain (Table 5).

**4.1.2 Interview Respondent Demographics.** Five survey respondents offered to participate in a 30-minute follow-up semi-structured interview. Our interviewees had very strong IoT credentials. They were from different countries and industries and represented our larger group of survey respondents with different levels of agile and IoT experience (Table 6).

Table 4. Reported Experience and Background (N = 45)

<b>Agile Experience</b>	<b>Frequency</b>	<b>IoT Industry</b>	<b>Frequency</b>
<1 year	2 (4.4%)	Agriculture	3 (6.7%)
1–3 years	10 (22.2%)	Energy	2 (4.4%)
3–5 years	15 (33.3%)	Healthcare	3 (6.7%)
5–10 years	13 (28.9%)	Industrial	17 (37.8%)
>10 years	5 (11.1%)	Public Services	2 (4.4%)
<b>Agile Knowledge</b>		Residential/Smart Buildings	5 (11.1%)
Not very knowledgeable	0 (0.0%)	Retail	2 (4.4%)
Somewhat knowledgeable	5 (11.1%)	Telecommunication	2 (4.4%)
Moderately knowledgeable	11 (24.4%)	Transportation	6 (13.3%)
Very knowledgeable	22 (48.9%)	Other	3 (6.7%)
Extremely knowledgeable	7 (15.6%)		
<b>IoT Experience</b>		<b>Startup</b>	
<1 year	4 (8.9%)	Yes	17 (37.8%)
1–3 years	23 (51.1%)	No	28 (62.2%)
3–5 years	9 (20.0%)		
5–10 years	9 (20.0%)		
>10 years	0 (0.0%)		

Table 5. Reported Project and Team Characteristics (N = 45)

<b>Team Size</b>	<b>Frequency</b>	<b>Complexity</b>	<b>Frequency</b>
3–5 team members	13 (28.9%)	Not complex at all	2 (4.4%)
6–8 team members	15 (33.3%)	Somewhat complex	4 (8.9%)
9–12 team members	9 (20.0%)	Moderately complex	22 (48.9%)
>12 team members	8 (17.8%)	Very complex	13 (28.9%)
<b>Team Setting</b>		Extremely complex	4 (8.9%)
Completely on-site	11 (24.4%)	<b>Uncertainty</b>	
Mostly on-site	8 (17.8%)	Not uncertain at all	3 (6.7%)
Distributed	19 (42.2%)	Somewhat uncertain	7 (15.6%)
Mostly remote	3 (6.7%)	Moderately uncertain	20 (44.4%)
Completely remote	4 (8.9%)	Very uncertain	9 (20.0%)
<b>Project Duration</b>		Extremely uncertain	6 (13.3%)
Ongoing	2 (4.4%)		
3–6 months	17 (37.8%)		
7–12 months	15 (33.3%)		
13–24 months	9 (20.0%)		
>24 months	2 (4.4%)		

## 4.2 Quantitative Results

The aim of the survey was to find whether agile practices, agile team skills, and IoT development team skills influence the project success dimensions in IoT projects. We first describe the data with mean and standard deviation, where agile practices, agile team skills, and IoT team skills are completely listed. Since the data in this study was measured on a Likert scale, which is an ordinal scale, Spearman's correlation analysis was deemed most suitable to assess relationships between the agile practices, team skills, and project success dimensions for this study [7]. Additionally, we



Table 6. Interview Respondent Demographics

ID	Country	Role	IoT Industry	Agile Experience	IoT Experience	Education
A	United States	Co-founder of IoT start-up	Agriculture	1–3 years	1–3 years	Master’s in Computer Science
B	The Netherlands	Head of IoT	Industrial	>10 years	3–5 years	Master’s in Management
C	Belgium	IoT Developer	Healthcare	1–3 years	1–3 years	Bachelor’s in Computer Science
D	The Netherlands	IoT Product Owner	Public services	5–10 years	5–10 years	Bachelor’s in Business
E	Germany	IoT Developer	Industrial	3–5 years	1–3 years	Master’s in Informatics

applied a Kruskal-Wallis test to assess differences between the sets of agile practices, skills, IoT skills, and project success metrics given the level of uncertainty and complexity of the last-most completed project as reported by our survey participants with a 5-point Likert from “Not at all” to “Extremely High.” Considering only those who evaluated their previous project as complex or uncertain allows us to understand nuanced differences between simple, complicated, and complex agile projects in IoT according to the Stacey matrix.

**4.2.1 Agile Practices—Survey Results.** The agile practices were ranked according to their mean and are shown in Table 7. The five most important agile practices for IoT development reported are Collective Code Ownership, Continuous Integration, Single Team, Dedicated Customer, and Sprint Planning. The five least important agile practices were Planning Poker, Test-Driven Development, Pair Programming, Automated Acceptance Testing, and Kanban Board. Nevertheless, all agile practices have a mean value of at least 3.00 (“moderately important”), which shows that the set of 24 agile practices is commonly used and important for IoT development. Further, the agile practices with a mean of at least 4.00 (“very important”) have a relatively low standard deviation, which indicates that there was consensus among respondents. Many agile practices with a lower mean show a higher standard deviation, which suggests that there was less consensus among respondents, on the other hand.

Spearman’s analysis revealed that 5 out of 24 agile practices significantly correlate with project success dimensions. Table 8 shows those agile practices that significantly correlate with a project success dimension. Our study found five agile practices (Portfolio Planning, Release Planning, Frequent Releases, Kanban board, and Unit Testing) had significant correlation with at least one project success dimension. The remaining 19 agile practices did not have significant correlation and none of the agile practices significantly correlated with the “budget,” “scope and requirements,” nor with the “end user satisfaction” dimensions.

Additionally, the results of the Kruskal-Wallis test show there is a statistically significant difference in the importance level of the use of Product Roadmaps ( $p = 0.026$ ), Dedicated Customers ( $p = 0.036$ ), Frequent Releases ( $p = 0.035$ ), and Coding Standards ( $p = 0.023$ ) for participants who estimated their previous project to have high complexity as compared to medium complexity (Table 9).

**4.2.2 Agile Team Skills—Survey Results.** Table 10 shows how respondents rated the competence level of agile skills within their teams. Respondents rated the competence of 10 agile skills within their team on a 5-point Likert scale, with the option to mark skills as not applicable. Therefore, a score of 4.00 would translate to “very competent” and a score of 3.00 would translate to “moderately competent,” while a lower N value indicates one or more respondents have marked a skill as not applicable. These N/A values have been omitted from the statistical analysis and the “valid N”

Table 7. Reported Importance of Agile Practices (N = 45, Columns 1–3) Compared with Literature-Based Findings of Table 1 (Columns 4 and 5)

Agile Practice	Mean	Standard Deviation ( $\sigma$ )	Engineering or Management Practice	Rank in Table 1
Collective Code Ownership	4.47	0.842	Engineering	9 of 9
Continuous Integration	4.27	0.915	Engineering	3 of 9
Single Team	4.18	1.114	Management	10 of 15
Dedicated Customer	4.13	1.120	Management	9 of 15
Sprint Planning	4.09	0.996	Management	2 of 15
Product Roadmap	4.00	0.905	Management	13 of 15
Coding Standards	3.98	1.055	Engineering	2 of 9
Unit Testing	3.98	1.234	Engineering	1 of 9
Continuous Deployment	3.91	1.221	Engineering	5 of 9
Retrospective	3.89	1.318	Management	3 of 15
Daily Standup	3.89	1.402	Management	1 of 15
Refactoring	3.84	1.147	Engineering	4 of 9
Release Planning	3.80	1.100	Management	6 of 15
Common Work Area	3.78	1.347	Management	12 of 15
Short Iterations	3.71	1.036	Management	5 of 15
Sprint Review	3.71	1.236	Management	4 of 15
Portfolio Planning	3.69	1.164	Management	15 of 15
Story Mapping	3.69	1.535	Management	14 of 15
Frequent Releases	3.62	1.211	Management	11 of 15
Kanban Board	3.58	1.500	Management	7 of 15
Automated Acceptance Testing	3.47	1.531	Engineering	8 of 9
Pair Programming	3.40	1.587	Engineering	6 of 9
Test-Driven Development	3.31	1.459	Engineering	7 of 9
Planning Poker	3.22	1.550	Management	8 of 15

Table 8. Spearman's Correlations between Agile Practices and Project Success Dimensions

		Budget	Time	Scope and Requirements	Customer/Stakeholder Satisfaction	Team Satisfaction	End User Satisfaction
Portfolio Planning	$\sigma$		.323*				
	Sig		.030				
Release Planning	$\sigma$				.332*		
	Sig				.026		
Frequent Releases	$\sigma$		.326*				
	Sig		.029				
Kanban Board	$\sigma$					-.353*	
	Sig					.017	
Unit Testing	$\sigma$					.300*	
	Sig					.045	

\*Correlation is significant at the 0.05 level (two-tailed).

\*\*Correlation is significant at the 0.01 level (two-tailed).

value of 43 indicates that 43 survey responses did not have any missing values. This shows that most of the respondents considered all of these 10 skills applicable to some degree for their teams in IoT development.

As can be seen in Table 11, there are several correlations between agile team skills and the project success dimensions in comparison to the agile practices. Eight of the ten agile team skills correlated significantly with multiple project success dimensions. Release management and systems did not

Table 9. Kruskal-Wallis Test of Agile Practices Compared to Project Complexity

Independent-Samples Kruskal-Wallis Test Summary; N = 45							
Product Roadmap Project Complexity		Dedicated Customer Project Complexity		Frequent Releases Project Complexity		Coding Standards Project Complexity	
Test Statistic	11.093 <sup>a</sup>	Test Statistic	10.291 <sup>a</sup>	Test Statistic	10.324 <sup>a</sup>	Test Statistic	11.351 <sup>a</sup>
<i>df</i>	4	<i>df</i>	4	<i>df</i>	4	<i>df</i>	4
Asymptotic Sig.(two-sided test)	0.026	Asymptotic Sig.(two-sided test)	0.036	Asymptotic Sig.(two-sided test)	0.035	Asymptotic Sig.(two-sided test)	0.023

<sup>a</sup>The test statistic is adjusted for ties.  
Asymptotic significances are displayed. The significance level is .050.

Table 10. Reported Competence of Agile Team Skills

Agile Team Skill Competencies	Mean	Standard Deviation $\sigma$	N
Collaboration	4.09	0.973	45
Communication	4.07	1.009	45
Architecture	3.95	0.714	44
Systems	3.93	0.837	45
Software Engineering	3.82	0.896	44
Leadership	3.62	1.007	45
Quality	3.60	0.837	45
Release Management	3.53	0.882	43
Project Management	3.51	1.058	45
Testing	3.26	1.093	43
Valid N (listwise)	43		

significantly correlate with any of the six project success dimensions. In the context of this study, systems refers to the combination of databases, hardware and infrastructure, operating systems, and programming.

Leadership is a significant agile skill both for high complexity as well as high uncertainty projects according to the findings in the associated Kruskal-Wallis test. Table 12 details the results.

**4.2.3 IoT Development Team Skills—Survey Results.** Table 13 shows how respondents have rated the competence of their team regarding IoT development skills. In similar fashion to rating the agile team skills, the seven IoT development skills were rated on a 5-point Likert scale, with the option to mark skills as not applicable. Thirty-seven respondents rated all skills in this section of the survey, which suggests that several respondents considered some of these seven skills to be not relevant to their teams. “Embedded software development” was marked as not applicable most often. Possible explanations are that respondents did not need this skill for their specific project, did not understand its description in the survey, or outsourced this skill to an external supplier. Respondents rated the competence of their teams in “hardware” lower than other skills and considered their teams to be most competent in “cloud technologies,” followed by “web client software” and “wireless connections.”

As displayed in Table 14, “data analytics and visualization” correlates with four of the project success dimensions, while “web client software” correlates with three project success dimensions and “cloud technologies” correlates with two project success dimensions. “Hardware” and “wireless connections” do not correlate with any of the project success dimensions. Moreover, the project success dimensions “time,” “scope and requirements,” as well as “end user satisfaction” all correlate with three of the skills. None of the skills correlate with “budget,” while “customer or stakeholder

Table 11. Spearman’s Correlations between Agile Team Skills and Project Success Dimensions

		Budget	Time	Scope and Requirements	Customer/ Stakeholder Satisfaction	Team Satisfaction	End User Satisfaction
Architecture	$\sigma$	.370	.374*	.413**	.509**	.371*	.480*
	Sig	.013	.012	.005	.000	.013	.001
Collaboration	$\sigma$	.389**	.354*	.402**	.410**	.586**	.387**
	Sig	.008	.017	.006	.005	.000	.009
Communication	$\sigma$	.460**	.361*	.312**	.446**	.563**	
	Sig	.001	.015	.037	.002	.000	
Leadership	$\sigma$		.418**	.390*	.471**	.454**	.405**
	Sig		.004	.008	.001	.002	.006
Project Management	$\sigma$	.334*		.302*		.409**	
	Sig	.025		.044		.005	
Quality	$\sigma$	.333*	.515**	.392**	.411**	.506**	.320*
	Sig	.025	.000	.008	.005	.000	.032
Release Management	$\sigma$						
	Sig						
Software Engineering	$\sigma$					.347*	
	Sig					.021	
Systems	$\sigma$						
	Sig						
Testing	$\sigma$		.435**	.404**	.412**	.362*	
	Sig		.004	.007	.006	.017	

\*Correlation is significant at the 0.05 level (two-tailed).

\*\*Correlation is significant at the 0.01 level (two-tailed).

Table 12. Kruskal-Wallis Test of Leadership Skills Compared to Project Complexity and Uncertainty Levels

Independent-Samples Kruskal-Wallis Test Summary; N = 45			
Leadership – Project Complexity		Leadership – Project Uncertainty	
Test Statistic	10.251 <sup>a</sup>	Test Statistic	9.669 <sup>a</sup>
<i>df</i>	4	<i>df</i>	4
Asymptotic Sig.(two-sided test)	0.036	Asymptotic Sig.(two-sided test)	0.046

<sup>a</sup>The test statistic is adjusted for ties.

Asymptotic significances are displayed. The significance level is .050.

Table 13. Reported Competence of IoT Development Team Skills (N = 45)

IoT Development Team Skill Competencies	Mean	Standard Deviation $\sigma$	N
Cloud Technologies	3.98	1.102	43
Web Client Software	3.79	0.976	42
Wireless Connections	3.76	0.916	41
Data Analytics and Visualization	3.58	0.982	43
Embedded Software Development	3.56	1.071	39
Distributed Systems	3.55	0.848	44
Hardware	3.19	1.075	43
Valid N (listwise)	37		

Table 14. Spearman’s Correlations between IoT Development Team Skills and Project Success Dimensions

		Budget	Time	Scope and Requirements	Customer/Stakeholder Satisfaction	Team Satisfaction	End User Satisfaction
Cloud Technologies	$\sigma$			.302*			.302*
	Sig			.049			.049
Data Analytics and Visualization	$\sigma$		.363	.371*		.530**	.370*
	Sig		.017	.014		.000	.015
Distributed Systems	$\sigma$		.308				
	Sig		.042				
Embedded SW Development	$\sigma$				.317*		
	Sig				.050		
Hardware	$\sigma$						
	Sig						
Web Client Software	$\sigma$		.419*	.313*			.467**
	Sig		.006	.044			.002
Wireless Connections	$\sigma$						
	Sig						

\*Correlation is significant at the 0.05 level (two-tailed).

\*\*Correlation is significant at the 0.01 level (two-tailed).

Table 15. Kruskal-Wallis Test of Project Uncertainty to Time and Customer/Stakeholder Satisfaction Levels

Independent-Samples Kruskal-Wallis Test Summary			
Independent-Samples Kruskal-Wallis Test Summary		Customer/Stakeholder Satisfaction - Project Uncertainty	
Test Statistic	13.638 <sup>a</sup>	Test Statistic	9.739 <sup>a</sup>
<i>df</i>	4	<i>df</i>	4
Asymptotic Sig.(two-sided test)	0.009	Asymptotic Sig.(two-sided test)	0.045

<sup>a</sup>The test statistic is adjusted for ties.

Asymptotic significances are displayed. The significance level is .050.

satisfaction” and “team satisfaction” each correlate with one of the skills. There were no significant findings in the Kruskal-Wallis tests.

**4.2.4 Project Success Metrics—Survey Results.** We also evaluated for significant differences in project success metrics (Table 15). Here, the Kruskal-Wallis tests show there is a statistically significant difference in the importance level of Time and Customer-Stakeholder Satisfaction for survey participants who rated the uncertainty level as “Extremely High.”

### 4.3 Interview Results

After exploring the quantitative results, the interviews aimed to provide additional insights on how agile practices contribute to project success linked to their responses on the survey. Five participants agreed to interviews in addition to completing the survey. A summary of interviewee demographics is found in Table 6.

**4.3.1 Suitability.** All interviewees could share positive examples of applying agile methods in IoT development, and as such, there was consensus among them that agile methods can be suitable for IoT development. According to participant A, practitioners usually adapt agile practices and processes to their own needs. Participant B also commented that agile practices are generally adjusted to specific projects, and that additional explanations are often required, as the broader scope of the IoT domain requires people to be knowledgeable about hardware and networking as well as software. Participant B found agile methods to be easier to apply to the software side than the

hardware side, because of lead times for hardware, especially when ordering from external parties. Once the hardware design has been created and the prototype is outsourced, there will be waiting times that can be unpredictable. In such cases, an agile approach can still be suitable because Sprints can be swapped in order to continue development even though the hardware isn't ready. Participant A stressed that IoT development brings an additional level of uncertainty because it is a newer field, and that agile methods can add a lot of value to IoT development by avoiding overinvestment in any one area (e.g., hardware, software, networking).

Participant D had several comments on the suitability of agile methods for IoT development. They commented that when building something new in a new market, agile methods are "perfect," allowing the development team to be efficient with time and money. Since the project determines if an agile approach is the right way forward, the scope, intake, context, and team should be carefully considered beforehand, especially in more conservative and conventional cultures. The team is an important factor, because when not everyone is convinced of the utility of an agile approach, some people may struggle or retreat, which may be detrimental for the outcomes of the project.

#### 4.3.2 Agile Practices.

*Planning and Roadmapping.* Participant A, co-founder of an IoT startup in the agriculture industry, commented that sprint planning and release planning reflect the importance of having a plan altogether. Planning may be even more important for IoT development in some aspects because IoT is still new and somewhat undefined, and therefore perhaps riskier. Even the process of creating a plan can be helpful, as it forces clarifying thinking about the required steps and dependencies, which is helpful for organizing where one needs and wants to go. Also, dedicated research ahead of time could help identify some of the problematic aspects of a solution that is being developed, while there is still time to go in a slightly different direction. At the same time, taking some risks is inherent to the IoT market, as trying to develop a differentiating product in the market can be the key to success, especially for startups.

Participant B, overseeing an IoT project in the industrial industry, found the product roadmap to be especially helpful when there are dependencies on external suppliers for hardware and related modules. When developing a new sensor that should be able to operate under extreme conditions, such as vibrations or very high temperatures, the project involves more than creating a performant sensor, because also an external party would have to be involved to complete the validation part. Thus, adding a step involving the certification of a new type of communication protocol or a new type of sensor. In addition, the product roadmap may need to require the customer receiving sales training in order to be prepared for technical questions and end users may need product training. Since these activities can take a couple of months, and product development cannot be pushed forward until the sensor is ready, it is important to have a good product roadmap for planning ahead of time. The usage of a product roadmap may be more important in larger organizations than in startups, where the salesperson, developer, and someone from operations are co-located, and all understand the product well enough to sell it. In particular, participant B noted "In larger organizations, innovative IoT teams should be more predictable, because more people are involved in the product launch and training." A product roadmap also makes it possible to describe which features the product will have in the first version, and which features will be added to later versions. This adds some flexibility to the exact design and implementation of the product, allowing more input later, and to react to unexpected situations that arise within the team or because of changes in technology.

A product roadmap can also be helpful in managing the scope of the project and the expectations of the customer, participant E explained. For instance, in heavy industry sectors a customer



may want to see the online status of all devices, even though this was not included in the contract. Therefore, it is important to be clear about definitions as well, to prevent situations where a customer may take a lot of features for granted under the name of “fleet management.” On the other hand, participant D started making less use of a product roadmap, because the future cannot always be predicted and therefore this is not a silver bullet. Nevertheless, the notion was that there is still value in making use of it to define how the envisioned results will be used, and to motivate stakeholders to work with these results. To come up with a useful product roadmap, good domain knowledge is required to anticipate what a customer or stakeholder may understand under certain IoT terminology.

*Commonality in Team and Structures.* Participants reported on different aspects of the importance of common baselines in team, spaces, and code. Participant E also pointed out that a single team working on an IoT project, rather than multiple teams, is extremely important to have. When working with microservices, it is often essential to have one very good developer in the backend, one very good developer in the frontend, perhaps an additional full-stack developer, and someone who does not develop but takes care of DevOps on the same team. They continue with “because trying to build a team in which everyone is an expert at everything will not work, it is important to find team members who each add some additional value to the overall team.”

Participant D found having a common work area to be an important practice, as it shows what is being worked on and what should be achieved. Although tools supporting working online can be very good, visualizing what is being worked on to make it more tangible and visible can have a lot of value.

Within a startup environment, it became apparent to participant A that the importance of collective code ownership is shown by not having to rely on any single individual to finish a task, because special knowledge is no longer kept in the brain of one individual. Participant C, active in a larger organization, shared that collective code ownership proves its value by ensuring that the code is peer-reviewed and mistakes can be corrected before releases.

*Continuous Integration and Continuous Deployment.* One to two people worked on participant A’s project simultaneously. They found using and refining development techniques provides value both during and after the project finished, which played an important role. Some of these techniques are included in the continuous integration practice. They found this practice helped build skills in a toolset that would carry over into future projects such as automated build and deployment processes. Participant D found continuous integration to especially help the environment when working in smaller teams, similarly to Participant A. Moreover, Participant A viewed the automation aspects of continuous integration and continuous deployment as a force multiplier that is helpful especially for smaller companies with limited resources, as the time spent on automation will pay benefits over and over again.

Participant E shared a practical example of how continuous integration adds value to IoT development involving Software-as-a-Service such as Microsoft Azure or similar services. Software-as-a-Service means some functionality is already available. In such a case, single unit tests are often insufficient because it could mean only a single function is tested that calls an individual microservice. Instead, continuous integration includes integration tests across the application. This helps testers understand the bigger picture of what has already been implemented, which can make delivery faster.

Participant C considered continuous deployment to be one of the most important agile practices because it enables constant visibility of how healthy the code base is. This can also create awareness of the system performance and possible performance gaps, so that the quality of the system

does not get jeopardized. A comprehensive software test suite within the agile code base can help to accomplish this and to deliver the product.

*Unit Testing.* Participant A believes in automated unit testing as part of the whole continuous integration process. In particular, with the thought in mind that the sooner a bug is caught, the less costly it is to fix it and the least impact it has. Participant B considered unit testing to be very important, even though a lot of time needs to be spent on it. For extremely complex project chains, unit testing, continuous delivery, deploying and designing infrastructure are essential.

*Aligning the Sprint to the Task.* Participant E valued sprint planning and pointed out that it is much more than working with tools that enable agile ways of working. People should be educated on how to do it appropriately, and it should focus on the sprint only. Participants C and E both agreed that it is important to look at the requirements and to align on what needs to be delivered after each sprint, since many people work on different parts of the project. Since sprints are iterative, the sprint reviews form a very important and continuous aspect. Participant D found short iterations to work well in IoT environments, both technically and functionally, and considered it to be the modern way of combining software development and product development.

*4.3.3 Advantages and Success Dimensions.* As would reasonably be expected from professionals practicing agile development in IoT, there was enthusiasm about the interactions between agile development and success dimensions. One respondent also mentioned that all agile practices are equally important for teams to be productive, while another respondent stated that the mindset and agile way of working in general contribute to reaching budget and time goals. Participant B mentioned that most of the agile practices are equally important because there will be more significant progress when all areas of agile practices are improved upon. Participant D mentioned that agile practices most significantly impacted the time dimension, because of quickly learning what adds value and what does not add value throughout the project; in turn this helps to deliver a result quickly.

Though the agile practices work in concert, some agile practices have a larger impact on success dimensions than others, argues participant B. This is due to team productivity. The productivity of a team could drop very quickly without having a good product roadmap, not having a dedicated customer, or not conducting a good retrospective after a demo of the product. By choosing to do product development physically alongside the people, they know who they are working with and can give immediate feedback, which is also important for productivity.

Participant C, active in the healthcare industry, found agile practices related to software craftsmanship to have a larger impact on quality than management-related agile practices and agile ways of working, such as planning poker. For example, collective code ownership, pair programming, continuous integration, and continuous deployment should all have a positive impact on the quality of what will be delivered. The agile ways of working, such as daily standup, sprint reviews, and sprint planning, may positively impact team satisfaction, because it increases communication between one another. Collective code ownership requires more collaboration between people, while it can also make it more enjoyable for those involved when both people enjoy working in the same environment. This aligns with the results in Table 10 and Table 11 where communication and collaboration are both ranked high.

Participant E made a clear distinction between customer/stakeholder satisfaction, and end user satisfaction, referring to a project where the customer was not really satisfied while the end user found it to be a brilliant platform. Since the requirements are gathered by engineers but the platform is used by the end user, there can be a discrepancy between perspectives of project success, i.e., mechanical/technical functionality versus usability.

*4.3.4 Difficulties and Lessons Learned.* Though our interviewees were broadly positive about the use of agile methods in IoT development, they recognized and reported on challenges from their perspectives and experiences.

Participants E, B, and D noted the difficulties in aligning differences in norms across organizational cultures. Participant E remarked on their experience with long-established industrial companies with a waterfall mindset, and the individuals in those companies not being open to making use of agile methods. This can create a mismatch between the agile development team and the customer that leads to friction, especially when a customer does not understand the agile mindset, does not have a product owner, and does not respond to tickets in a timely manner. In such a case, the reality will become a de facto agile waterfall, where the first step is analyzing, the second step is implementing, and the third step is testing. Participant B from the heavy industry sector commented that production is often very traditional and used to waterfall planning. As this often means the team is not allowed to start building before the design has been finished completely, it is recommended to plan building hardware up front. It is also important that the stakeholders understand technical terms as it is often difficult to describe an IoT solution on paper, and stakeholders better understand a solution when they physically see how a sensor works and the data it generates. To allow for some hands-on experience, customers can also be empowered to work with a portion of the data in order to come up with some first ideas of what the solution should entail. Further, stakeholder involvement in demos can help by letting stakeholders understand the complexity of the project, which in turn can lead to collaboration, thinking along in finding alternatives, and setting realistic expectations. Participant D experienced difficulties applying an agile approach in the public services industry when working with legacy systems. This can lead to negative forces, especially if there is a mismatch in cultures when people are used to working with other approaches, such as waterfall methods. In such cases, it is essential that people on a team actually want to work using agile practices, in order not to create any friction within the team.

In line with the survey responses, participant D reiterated the importance of communication. Continuously discussing what has been achieved, should be achieved, and if there are any obstacles, is important, because otherwise topics may be parked and could be put on hold. Participant D experienced how working on a project where all operational people work in the field made it difficult to utilize the daily standup, as not everyone could participate from different locations. Therefore, the focus on the dedicated customer increased, because continuously hearing the voice of the customer and end user was valued even more. Losing momentum here can be a large threat if there is no financial or time urgency, because then older projects will not be resumed. Due to the rapid technological developments with the platforms, sensors, and analytics tools that are being used, short-term changes can lead to struggles with politics and budgets. To ensure that IoT projects do not “slowly come to an end,” after which they may end up being resumed in a similar way but with newer technology, communicating about the opportunities is essential. From these experiences we also see that having a dedicated customer is essential in order to successfully develop an IoT product within an agile system.

Participant B detailed another situation that can arise in the field of IoT: the introduction of newly available sensors that could be researched and tested out. Since techniques, platforms, and architectural principles advance very quickly in the field of IoT, flexibility in terms of the end product is required. If this is done as a side activity, it could be at the expense of completing items that were planned in a sprint. Rather than forbidding these kinds of activities, participant B recommends structurally freeing some time in a sprint for unplanned items. Delivering business value should not be forgotten, however. Although both the customer and engineers may already be somewhat satisfied when something works in the first place, teams should first ask themselves

what they are going to build, and how is it going to create revenue, or save costs. This is something that has to be engrained in the team, to ensure that they are working toward reaching a shared goal in the next half a year or year, rather than working on something that may be exciting and interesting but not value-adding. Participant C reiterated that some engineers have a tendency to spend too much time on technical features of a project, such as implementing total custom libraries where the actual added input is not as important. Although this may be seen as important from an engineering perspective, from a business point of view it is not, and so to keep everyone aligned, engineers and the business need to be on the same page regarding project priorities.

Regarding business partner and stakeholder involvement and project architecture, participant B advocated that even if only a small portion of the automated provisioning of a sensor can be completed, achieving a first success can help ask for flexibility on the customer and stakeholder side. It can be beneficial to inform customers up front that things can be very different in the short and medium terms to manage their expectations. Suppliers should also be made aware that certain functions that are the best option today may not even be available anymore in the cloud platform and will have to be migrated to something else. This in turn makes it difficult to define a detailed project architecture, especially if it needs to be approved by an architecture board that may not meet on a weekly basis. In larger organizations, where other topics such as replacing an ERP package or introducing new infrastructure may be on the agenda, updates in the architecture cannot be made quickly and so it was recommended to institute a generic architecture that allows the team to make modifications themselves.

Participants C and E both addressed the role that the physical size of a device plays. The size affects the team's ability to simulate and test them. For example, in the healthcare sector, working with a lot of smaller devices makes IoT harder to simulate: artifacts are not a web application, but instead are devices that work with different protocols. Therefore, customer involvement plays an essential role to understand and test if the devices act as intended by the customer. Within industrial IoT, the size of end products can also be a limiting factor, participant E noted. For devices that are the size of a car or even larger, there typically will not be a physical test device which means it cannot be properly evaluated.

Another insight participant E stressed is the importance of not neglecting the theory behind everything when focusing on applying an agile approach in a practical manner. For instance, communication protocols are crucial for IoT and there is a great research domain in communication protocols, how to communicate safely, how to communicate serving traffic, and database normalization.

Lastly, participant D observed that although planning poker is being used, it may become tiresome after a couple of months which then also makes it less effective. This aligns with the previous finding that planning poker was rated lower than other agile practices in Table 7, and the recommendation is to introduce a new estimation method to ensure the team stays inspired.

## 5 DISCUSSION

The aim of this study was to better understand which agile practices and team skills are critical for successful IoT development projects from the perspectives of current agile IoT practitioners. In this section, we discuss the study results and provide insights into why certain agile practices and team skills are considered to be critical for project success from the perspectives of agile IoT practitioners. Our results are indicative of necessary conditions of an agile IoT development team, though we do not claim that these results alone are sufficient. The expert knowledge and leadership of the team lead is the first requirement for building a team optimized for agile IoT. We position our contribution as a building block toward the creation of a high-functioning agile IoT team.

### 5.1 Important Agile Practices Insights

In response to RQ1, which addresses currently important skills and competencies for agile IoT, we can recommend a series of individual agile skills and practices along with IoT skills that should aid team leaders in configuring future agile IoT development teams for success based on current practices in the field.

The somewhat unintuitive first result is that Collective Code Ownership is reported to be the most important agile engineering practice based on our sample, followed by Continuous Integration (Table 7). Collective Code Ownership is the least-used engineering practice as reported by the general overview of agile development in VersionOne/AgileAI's "Annual State of Agile Report" [15, 16]. We posit that the differences here may be driven by the hardware-software-cloud nexus of IoT development compared with development projects that concentrate on only hardware or software, based on the results of the interviews (Section 4.3.2). This interpretation also supports the difference between the use of "Single Team" in our sample (third most important practice) and the State of Agile Report, where it is the 10th most common practice of 15 management practices. The top-five practices are rounded out by Dedicated Customer and Sprint Planning—two management practices. The importance of Dedicated Customers is twofold; when the complexity level of the project is high, and when the culture of the customer is static, having a dedicated customer in place is highly relevant to the success of the project.

Finally, Sprint Planning is notable for its relatively high ranking across both our sample and the original reports [15, 16]. Combining the insights from the qualitative results and the practices-success correlation matrix, what we can infer is that aspects of planning are important to keeping agile IoT projects running smoothly. The point that planning aspects are in some way supportive of each other was addressed explicitly by participant A; in Table 8, we can also see that Portfolio Planning is significantly related to the success metric Time, and Release Planning is significantly related to Customer/Stakeholder Satisfaction. Our management recommendation for agile IoT leadership becomes less tied to the individual planning practice and more to the importance of utilizing planning. Otherwise said, while some agile practices are seen as essential though they do not guarantee success, if these practices (e.g., planning management) are not present, then it is likely the project will not be successful.

### 5.2 Competence of Agile Team Skills Insights

Continuing with RQ1, team-level competency in specific agile skills is important for the success of agile IoT projects. Collaboration and Communication, which are both soft skills, were rated highest among respondents, which confirms the insights taken from the literature review that these skills are essential for agile teams and high-performing teams in general. The next three highest agile team skills were hard skills, namely, Architecture, Systems, and Software Engineering. These skills are closely interrelated and make up the three most technical skills from the bank of agile-specific skills queried in the survey. Compared to other skills, survey respondents did not consider their teams to be very competent at "testing," as the rating of this skill was the lowest of all the agile team skills. This may be an artifact of the maturity of testing at the time of the interviews or a reflection of the difficulty of creating test environments that are suitable analogs to real environments. Future iterations of this work may find that technological developments shrink the gap, i.e., in the intervening time frame, digital twins have become prominent. Finally, leadership is critical for projects in which there is high complexity and/or high uncertainty.

As the literature review revealed, it often is not feasible to form a team consisting of C-shaped experts. Instead, it is desirable to have multiple T-shaped experts with different skills and expertise, who can complement the overall knowledge of the team [49]. Therefore, the skills of an entire team



are likely to be different from the skills of individual team members. This is in line with participant E, who stressed the importance of sourcing team members who each add some additional value to the overall team, because it is often not possible to build a team in which everyone is an expert at everything. Although not every team member needs to be an expert at every skill, it is desirable that the team as a whole is competent at each of these most important skills.

### 5.3 Competence of IoT Development Team Skills Insights

Respondents considered their teams to be most competent in Cloud Technologies, followed by Web Client Software and Wireless Connections. This aligns with the literature review, which described that the cloud forms an essential part of the IoT [65]. Knowledge of web client software is needed to let users interact with their IoT device, while competency in wireless connections is needed to turn objects into smart objects that can be connected to the internet. As such, both of these skills form a fundamental part of developing IoT systems. Respondents also considered their teams to be fairly competent in Data Analytics and Visualization. Although these skills are slightly less fundamental as not all IoT systems may require advanced data insights, they are important skills that can add significant value to the end user and provide a road map to future development of new products and innovation.

Embedded Software Development and Distributed Systems skills were rated just slightly lower; however, they are both essential for IoT systems to work properly. Although they are essential skills, the teams of the respondents may have been less experienced in them or may have found them to be more difficult and this may be why they were rated lower. Embedded Software Development was marked as not applicable most often. Possible explanations are that respondents did not understand its description in the survey, did not need this skill for their specific project, or have outsourced this skill to an external supplier as mentioned by participant B in their interview. Further, respondents rated the competence of their teams in Hardware significantly lower than other skills, which may suggest that not all teams of respondents required this skill, had the expertise, or have outsourced this skill.

### 5.4 Correlation of Skills and Practices with Project Success

In response to RQ2 on the elements of project success, significant correlations were found among both agile and IoT development team skills and project success. As can be observed, both soft and hard skills appear to be of importance for high-performing agile teams in IoT development. Having identified multiple critical skills further corresponds with the Agile Manifesto, which states that balanced teams with a range of different skills are important. Although the importance of team skills depends on the context of the project and organization, as seen by the importance of extremely high project uncertainty on timeliness and Customer/Stakeholder Satisfaction, these results do provide clear guidance on successful team formation. This also blends in with a comment from one of the interview respondents, who analogized that “the agile methodology and mindset provides value by going fast and following the concept of a goat path, in which you try to go end to end as quickly as possible, improve each step along the way and do not invest too much in aspects that may end up being wasted efforts.”

Although the management of agile teams is considered to be less heavy than in traditional teams, there should be leadership and project management qualities within the team in order to be able to pursue clear goals that can be achieved. Agile team members in IoT development should be good communicators and collaborators in addition to strong developers. Team members can have different knowledge and specialties, yet, it is recommended that they together possess sufficient knowledge of “architecture,” “web client software,” and “cloud technologies.” These are essential skills that enable teams to deliver quality IoT solutions that meet the expectations of customers.



However, the average team of the respondents has considerable room for improvement when it comes to “testing,” which proved to be of importance for meeting “time” goals, “scope and requirements,” and improving the “customer or stakeholder satisfaction.” Since survey respondents considered their projects to be moderately successful, increasing competence in “testing” may benefit their project outcomes.

Further, the results of the Spearman’s analysis revealed that not all of the 24 agile practices identified in literature significantly correlate with the project success dimensions in our sample. A possible explanation may be that respondents require different combinations of agile practices, as they work on different projects, at different organizations and in different industries. Such a contextual dependency could make it more difficult to find which specific agile practices correlate with the project success dimensions. On the other hand, another possible explanation may be that the large number of agile practices do not uniquely contribute to the prediction, but rather contribute as a whole. This statement aligns with the literature, which states that choosing an agile approach depends on the type of project, organization, culture, and requirements [42], especially in an IoT setting. This suggests that each agile practice can bring different advantages to an IoT development team and that choosing a set of agile practices that best supports IoT development will be dependent on many different factors. Preferences and experiences within the team may also impact the best choice of agile practices.

## 6 CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

This is the first study to our knowledge that assessed leveraging agile practices for IoT development. The study involved real practitioners who are experienced in using agile methodologies in the field of IoT development and are active in diverse industries worldwide. This is one of the first global studies examining the optimal agile team design for IoT in a variety of industries.

This research is important to practitioners and academics alike. This study shows that agile practices are suitable for IoT development and can play an important role in achieving successful projects. It identified that soft skills are as important as technical skills to project success, which suggests that the team chemistry is as important or more important than having superstar individual contributors. Our study found that multiple agile practices play an important role in IoT development. Agile practices that appear to be vital for IoT development include Collective Code Ownership, Continuous Integration, Single Team, Dedicated Customer, and Sprint Planning. In addition to the importance of both hard and soft skills for successful IoT development, we find that the importance of individual aspects like leadership, customer satisfaction, or planning and release scheduling is higher when the complexity or uncertainty level of the project is also high. These results are supportive of the value of agile IoT teams adopting agile practices to ensure project success, and can help agile IoT projects that have inherent risk to be more stable and outcomes to be more predictable.

We recognize that the sample size for this study is relatively small and that the study is cross-sectional in nature. Therefore, the ability to generalize our findings beyond this study is limited. As with most studies, future research could aim at validating the results with a larger sample size. Nevertheless, this study is unique in that developers and IoT professionals active around the globe participated. Another limitation linked to the sampling strategy is that the survey was directed to those with expertise in both Agile Software Development and IoT development. Practitioners in only one or the other development environment are not currently represented in our sample, which may result in some bias in attitudes toward agile practices. Likewise, we can assume that those who use an agile approach also have a neutral or positive perspective of agile development. Future researchers could consider a design where all practitioner expertise types are included. This would likely require changing the nesting of the research design from survey-interview to

interview-survey due to a potential lack of experience [61]. Comparative research addressing the use of agile or other methods in IoT development and the relationships with suitability of approach and project success would be a fruitful future direction. Both a strength and a weakness are that some respondents were still developing ideas or were working on prototypes, whereas other respondents had completed multiple IoT projects. Respondents who were new to IoT development may have had less insight into the applications of agile practices in their teams, as they may not yet have been able to optimize their agile ways of working. Yet, their insights are still valuable to help set the future direction of the composition of agile IoT teams. Future work could include industry, company size, or technology maturity specific-case studies. Our goal was to map the state of the field now; however, as IoT continues to evolve there will be many opportunities to build on this foundational work.

## ACKNOWLEDGMENTS

We would like to express our appreciation for the research support and resources made available through the University of Nebraska Omaha, College of Information Science and Technology, Public Health Informatics Research Laboratory.

## REFERENCES

- [1] Albert Albers, Jonas Heimicke, Johannes Müller, and Markus Spadinger. 2019. Agility and its features in mechatronic system development: A systematic literature review. In *ISPM Conference Proceedings*. 1–13.
- [2] Mashal Alqudah and Rozilawati Razali. 2017. Key factors for selecting an Agile method: A systematic literature review. *International Journal on Advanced Science, Engineering and Information Technology* 7 (2017), 526–537. Retrieved September 9, 2019 from <https://doi.org/10.18517/ijaseit.7.2.1830>
- [3] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. 2001. Manifesto for Agile Software Development. Retrieved September 9, 2019 from <http://www.agilemanifesto.org/>.
- [4] Bradford S. Bell and Steve W. J. Kozlowski. 2002. A Typology of Virtual Teams: Implications for Effective Leadership. <https://doi.org/10.1177/1059601102027001003>.
- [5] Ken Blanchard, Don Carew, and Eunice Parisi-Carew. 1996. How to get your group to perform like a team. *Training and Development* 50, 9 (1996).
- [6] Barry Boehm and Richard Turner. 2005. Management challenges to implementing agile processes in traditional development organizations. *IEEE Software* 22, 5 (2005), 30–39. Retrieved September 9, 2019 from <https://doi.org/10.1109/MS.2005.129>
- [7] S. Boslaugh and P. A. Watters. 2008. *Statistics in a Nutshell*. O'Reilly Media, Inc., California.
- [8] Rafael Camara, Annellyethon Alves, Iury Monte, and Marcelo Marinho. 2020. Agile global software development. In *Proceedings of the 34th Brazilian Symposium on Software Engineering*, Everton Cavalcante, Francisco Dantas, and Thais Batista (Eds.). ACM, New York, NY, 31–40. Retrieved May 26, 2021 from <https://doi.org/10.1145/3422392.3422411>
- [9] Luiz Fernando Capretz and Faheem Ahmed. 2010. Making sense of software development and personality types. *IT Professional* 12, 1 (2010), 6–13. Retrieved September 9, 2019 from <https://doi.org/10.1109/MITP.2010.33>
- [10] A. Cockburn and J. Highsmith. 2001. Agile software development: The business of innovation. *Computer* 34, 9 (2001), 120–127.
- [11] Kieran Conboy and Brian Fitzgerald. 2010. Method and developer characteristics for effective agile method tailoring. *ACM Transactions on Software Engineering and Methodology* 20, 1 (2010), 1–30. Retrieved September 9, 2019 from <https://doi.org/10.1145/1767751.1767753>
- [12] John W. Creswell and J. David Creswell. 2018. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (5th ed.). SAGE, Los Angeles, London, and New Delhi.
- [13] Claudia O. De Melo, Daniela S. Cruzes, Fabio Kon, and Reidar Conradi. 2013. Interpretative case studies on agile team productivity and management. In *Information and Software Technology* 55, 2 (2013), 412–427. Retrieved September 9, 2019 from <https://doi.org/10.1016/j.infsof.2012.09.004>
- [14] digital.ai. 2018. The 12th Annual State of Agile Report. Retrieved September 9, 2019 from [https://www.academia.edu/41465012/StateOfAgile\\_COLLAB\\_NET\\_VERSIONONE\\_COM](https://www.academia.edu/41465012/StateOfAgile_COLLAB_NET_VERSIONONE_COM).
- [15] digital.ai. 2019. The 13th Annual State of Agile Report. Retrieved September 9, 2019 from [https://www.duxdiligens.com/wp-content/uploads/2019/09/13th-annual-state-of-agile-report\\_7\\_May\\_2019.pdf](https://www.duxdiligens.com/wp-content/uploads/2019/09/13th-annual-state-of-agile-report_7_May_2019.pdf).

- [16] digital.ai. 2020. The 14th Annual State of Agile Report. Retrieved May 26, 2021 from <https://www.qagile.pl/wp-content/uploads/2020/06/14th-annual-state-of-agile-report.pdf>.
- [17] Dorothee Feldmuller. 2018. Usage of agile practices in mechatronics system design potentials, challenges and actual surveys. In *2018 19th International Conference on Research and Education in Mechatronics (REM'18)*. IEEE, 30–35. Retrieved September 9, 2019 from <https://doi.org/10.1109/REM.2018.8421803>
- [18] Vlad-Valentin Fireteanu. 2020. Agile methodology advantages when delivering Internet of Things projects. In *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI'20)*. IEEE, 1–5. Retrieved May 26, 2021 from <https://doi.org/10.1109/ECAI50035.2020.9223172>
- [19] Mohammad Ghafari, Timm Gross, Davide Fucci, and Michael Felderer. 2020. Why research on test-driven development is inconclusive?. In *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM'20)*. ACM, New York, NY, 1–10. Retrieved May 26, 2021 from <https://doi.org/10.1145/3382494.3410687>
- [20] Joachim Goll and Daniel Hommel. 2015. *Mit Scrum zum gewünschten System*. Springer Vieweg Wiesbaden. Retrieved September 9, 2019 from <https://doi.org/10.1007/978-3-658-10721-5>
- [21] Mila Hakanen and Aki Soudunsaari. 2012. Building trust in high-performing teams. *Technology Innovation Management Review* (2012). Retrieved September 9, 2019 from <https://doi.org/10.22215/timreview567>
- [22] Aymeric Hemon, Barbara Lyonnet, Frantz Rowe, and Brian Fitzgerald. 2020. From agile to DevOps: Smart skills and collaborations. *Information Systems Frontiers* 22, 4 (2020), 927–945. Retrieved May 26, 2021 from <https://doi.org/10.1007/s10796-019-09905-1>
- [23] Alistair Cockburn and Jim Highsmith. 2001. Agile software development: The people factor. *Computer* 34 (2001), 131–133. <https://ieeexplore.ieee.org/document/963450>.
- [24] Rashina Hoda, Norsaremah Salleh, John Grundy, and Hui Mien Tee. 2017. Systematic literature reviews in agile software development: A tertiary study. *Information and Software Technology* 85 (2017), 60–70. Retrieved September 9, 2019 from <https://doi.org/10.1016/j.infsof.2017.01.007>
- [25] Philip M. Huang, Ann G. Darrin, and Andrew A. Knuth. 2012. Agile hardware and software system engineering for innovation. In *IEEE Aerospace Conference Proceedings*. Retrieved September 9, 2019 from <https://doi.org/10.1109/AERO.2012.6187425>
- [26] IEEE Standards Association. 2015. Internet of Things (IoT) Ecosystem Study. Retrieved September 9, 2019 from [https://iot.ieee.org/images/files/pdf/iot\\_ecosystem\\_exec\\_summary.pdf](https://iot.ieee.org/images/files/pdf/iot_ecosystem_exec_summary.pdf).
- [27] International Data Corporation. 2019. IDC Forecasts Worldwide Spending on the Internet of Things to Reach \$745 Billion in 2019. Retrieved September 9, 2019 from <https://www.businesswire.com/news/home/20190103005070/en/IDC-Forecasts-Worldwide-Spending-on-the-Internet-of-Things-to-Reach-745-Billion-in-2019-Led-by-the-Manufacturing-Consumer-Transportation-and-Utilities-Sectors>.
- [28] International Data Corporation. 2019. Worldwide Spending on the Internet of Things Will Slow in 2020 Then Return to Double-Digit Growth, According to a New IDC Spending Guide. Retrieved September 9, 2019 from <https://www.idc.com/getdoc.jsp?containerId=prUS46609320>.
- [29] Ivar Jacobson, Ian Spence, and Pan-Wei Ng. 2017. Is there a single method for the Internet of Things? *Communications of the ACM* 60, 11 (2017), 46–53. Retrieved September 9, 2019 from <https://doi.org/10.1145/3106637>
- [30] Ayesha Khalid, Shariq Aziz Butt, Tauseef Jamal, and Saikat Gochhait. 2020. Agile scrum issues at large-scale distributed projects. *International Journal of Software Innovation* 8, 2 (2020), 85–94. Retrieved May 26, 2021 from <https://doi.org/10.4018/IJSI.2020040106>
- [31] Kaisa Könölä, Samuli Suomi, Tuomas Mäkilä, Tero Jokela, Ville Rantala, and Teijo Lehtonen. 2016. Agile methods in embedded system development: Multiple-case study of three industrial cases. *Journal of Systems and Software* 118 (2016), 134–150. Retrieved September 9, 2019 from <https://doi.org/10.1016/j.jss.2016.05.001>
- [32] Martin Kropp, Andreas Meier, Craig Anslow, and Robert Biddle. 2018. Satisfaction, practices, and influences in agile software development. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, Austen Rainer, Stephen G. MacDonell, and Jacky Keung (Eds.). ACM, New York, NY, 112–121. Retrieved September 9, 2019 from <https://doi.org/10.1145/3210459.3210470>
- [33] In Lee and Kyoochun Lee. 2015. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons* 58, 4 (2015), 431–440. Retrieved September 9, 2019 from <https://doi.org/10.1016/j.bushor.2015.03.008>
- [34] Xuan Lu, Zhenpeng Chen, Xuanzhe Liu, Huoran Li, Tao Xie, and Qiaozhu Mei. 2017. PRADO: Predicting app adoption by learning the correlation between developer-controllable properties and user behaviors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), Article 79, 30 pages. Retrieved September 9, 2019 from <https://doi.org/10.1145/3130944>
- [35] Carry MacGillivray, Stacy Crook, Marcus Torchia, Krishna Chinta, Jonathan Leung, Gabriele Roberti, Heriberto Roman, Alexandra Rotaru, Yuta Torisu, and Nigel Wallis. 2021. Worldwide Internet of Things Forecast Update, 2020–2024. Retrieved May 26, 2021 from <https://www.idc.com/getdoc.jsp?containerId=US47434121>.

- [36] Aniket Mahanti. 2006. Challenges in enterprise adoption of agile methods—A survey. *Journal of Computing and Information Technology* 14, 3 (2006). Retrieved September 9, 2019 from <https://doi.org/10.2498/cit.2006.03.03>
- [37] Dominik Maximini. 2015. *The Scrum Culture*. Springer.
- [38] Andrew McAfee and Erik Brynjolfsson. 2012. Spotlight on big data big data: The management revolution, 2012.. *Harvard Business Review* (2012). Retrieved September 9, 2019 from <https://hbr.org/2012/10/big-data-the-management-revolution>.
- [39] Soukaina Merzouk, Abdessamad Cherkaoui, Abdelaziz Marzak, and Sael Nawal. 2020. IoT methodologies: Comparative study. *Procedia Computer Science* 175 (2020), 585–590. Retrieved May 26, 2021 from <https://doi.org/10.1016/j.procs.2020.07.084>
- [40] Alessandro Montanari, Cecilia Mascolo, Kerstin Sailer, and Sarfraz Nawaz. 2017. Detecting emerging activity-based working traits through wearable technology. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), Article 86, 24 pages. Retrieved September 9, 2019 from <https://doi.org/10.1145/3130951>
- [41] D. Mourtzis, E. Vlachou, and N. Milas. 2016. Industrial big data as a result of IoT adoption in manufacturing. In *Procedia CIRP* 55 (2016), 290–295. Retrieved September 9, 2019 from <https://doi.org/10.1016/j.procir.2016.07.038>
- [42] Sridhar Nerur, Radhakanta Mahapatra, and George Mangalaraj. 2005. Challenges of Migrating to Agile Methodologies. *Communications of the ACM* 48, 5 (2005), 72–78. Retrieved September 9, 2019 from <https://doi.org/10.1145/1060710.1060712>
- [43] Stina Nylander, Anders Wallberg, and Pär Hansson. 2017. Challenges for SMEs entering the IoT world—Success is about so much more than technology. In *ACM International Conference Proceeding Series*. Retrieved September 9, 2019 from <https://doi.org/10.1145/3131542.3131547>
- [44] Pádraig O’Leary, Fergal McCaffery, Steffen Thiel, and Ita Richardson. 2012. An Agile process model for product derivation in software product line engineering. In *Journal of Software: Evolution and Process* 24, 5 (2012), 561–571. Retrieved September 9, 2019 from <https://doi.org/10.1002/smr.498>
- [45] Mazni Omar, Noor Liza Ahmad Khasasi, Sharifah Lailee Syed Abdullah, Nor Laily Hashim, Rohaida Romli, and Norliza Katuk. 2018. Defining skill sets requirements for Agile Scrum team formation. *Journal of Engineering and Applied Sciences* 13, 3 (2018), 784–789. Retrieved September 9, 2019 from <https://doi.org/10.3923/jeasci.2018.784.789>
- [46] Maria Paasivaara and Philippe Kruchten (Eds.). 2020. *Agile Processes in Software Engineering and Extreme Programming—Workshops*. Springer International Publishing, Cham. Retrieved May 26, 2021 from <https://doi.org/10.1007/978-3-030-58858-8>
- [47] Alexander Poth, Mario Kottke, and Andreas Riel. 2020. Evaluation of agile team work quality. In *Agile Processes in Software Engineering and Extreme Programming—Workshops*, Maria Paasivaara and Philippe Kruchten (Eds.). Lecture Notes in Business Information Processing, Vol. 396. Springer International Publishing, Cham, 101–110. Retrieved May 26, 2021 from [https://doi.org/10.1007/978-3-030-58858-8\delimiter"026E30F\\$textunderscore](https://doi.org/10.1007/978-3-030-58858-8\delimiter)
- [48] Donald J. Reifer. 2002. How good are agile methods? *IEEE Software* 19, 4 (2002), 16–18. Retrieved September 9, 2019 from <https://doi.org/10.1109/MS.2002.1020280>
- [49] Peyman Rostami and Mahmood Neshati. 2019. T-shaped grouping: Expert finding models to agile software teams retrieval. *Expert Systems with Applications* 118 (2019), 231–245. Retrieved September 9, 2019 from <https://doi.org/10.1016/j.eswa.2018.10.015>
- [50] Kenneth S. Rubin. 2012. Essential scrum. In *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley.
- [51] Daniel Russo. 2021. The agile success model. *ACM Transactions on Software Engineering and Methodology* 30, 4 (2021), 1–46. Retrieved May 26, 2021 from <https://doi.org/10.1145/3464938>
- [52] Farzad Samie, Lars Bauer, and Jorg Henkel. 2016. IoT technologies for embedded computing: A survey. In *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS’16)*. Retrieved September 9, 2019 from <https://doi.org/10.1145/2968456.2974004>
- [53] Pedro Santos, Marco Consolaro, and Alessandro Di Gioia. 2019. *Agile Technical Practices Distilled* (1st ed.). Packt Publishing and Safari, Erscheinungsort nicht ermittelbar and Boston, MA. Retrieved September 9, 2019 from <https://learning.oreilly.com/library/view/-/9781838980849/?ar>.
- [54] Ken Schwaber and Jeff Sutherland. 2012. *Software in 30 Days: How Agile Managers Beat the Odds, Delight their Customers, and Leave Competitors in the Dust* (online-ausg ed.). Wiley, Hoboken, NJ. Retrieved September 9, 2019 from <http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10546605>.
- [55] Pedro Serrador and Jeffrey K. Pinto. 2015. Does Agile work?—A quantitative analysis of agile project success. *International Journal of Project Management* 33, 5 (2015), 1040–1051. Retrieved September 9, 2019 from <https://doi.org/10.1016/j.jpproman.2015.01.006>
- [56] Helen Sharp, Hugh Robinson, and Marian Petre. 2009. The role of physical artefacts in agile software development: Two complementary perspectives. *Interacting with Computers* 21, 1–2 (2009), 108–116. Retrieved September 9, 2019 from <https://doi.org/10.1016/j.intcom.2008.10.006>

- [57] Jason Sharp and Sherry Ryan. 2011. Global agile team configuration. *Journal of Strategic Innovation and Sustainability* 7 (2011), 120–134.
- [58] Yogeshwar Shastri, Rashina Hoda, and Robert Amor. 2016. Does the “Project Manager” still exist in agile software development projects?. In *2016 23rd Asia-Pacific Software Engineering Conference (APSEC’16)*. IEEE, 57–64. Retrieved September 9, 2019 from <https://doi.org/10.1109/APSEC.2016.019>
- [59] Aaron J. Shenhar, Dov Dvir, Ofer Levy, and Alan C. Maltz. 2001. Project success: A multidimensional strategic concept. *Long Range Planning* 34, 6 (2001), 699–725. Retrieved September 9, 2019 from [https://doi.org/10.1016/S0024-6301\(01\)00097-8](https://doi.org/10.1016/S0024-6301(01)00097-8)
- [60] Satyajit Sinha. 2021. State of IoT 2021: Umber of Connected IoT Devices Growing 9% to 12.3 Billion Globally, Cellular IoT Now Surpassing 2 Billion. Retrieved May 26, 2021 from <https://iot-analytics.com/number-connected-iot-devices/>.
- [61] Mario Luis Small. 2011. How to conduct a mixed methods study: Recent trends in a rapidly growing literature. *Annual Review of Sociology* 37 (2011), 57–86.
- [62] Apoorva Srivastava, Sukriti Bhardwaj, and Shipra Saraswat. 2017. SCRUM model for agile methodology. In *2017 International Conference on Computing, Communication and Automation (ICCCA’17)*. IEEE, 864–869. Retrieved September 9, 2019 from <https://doi.org/10.1109/CCAA.2017.8229928>
- [63] R. D. Stacey. 1996. *Strategic Management & Organisational Dynamics*. Pitman, London, United Kingdom.
- [64] N. Steireif, M. Schirmer, M. Schnitzler, and S. Mutze-Niewohner. 2020. Towards an extended team model for agile development of complex products. In *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM’20)*. IEEE, 280–284. Retrieved May 26, 2021 from <https://doi.org/10.1109/IEEM45057.2020.9309811>
- [65] Antero Taivalsaari and Tommi Mikkonen. 2017. A roadmap to the programmable world: Software challenges in the IoT Era. *IEEE Software* 34, 1 (2017), 72–80. Retrieved September 9, 2019 from <https://doi.org/10.1109/MS.2017.26>
- [66] Antero Taivalsaari and Tommi Mikkonen. 2018. On the development of IoT systems. In *2018 3rd International Conference on Fog and Mobile Edge Computing (FMEC’18)*. Retrieved September 9, 2019 from <https://doi.org/10.1109/FMEC.2018.8364039>
- [67] Thomas Usländer and Thomas Batz. 2018. Agile service engineering in the industrial Internet of Things. *Future Internet* 10, 10 (2018), 100. Retrieved September 9, 2019 from <https://doi.org/10.3390/fi10100100>
- [68] M. Vijaya Bharathi and V. Spurthi. 2013. A survey on efficient agile development methods. *International Journal of Engineering Research and Technology* 2, 9 (2013).
- [69] Dave West. 2011. Water-scrum-fall is the reality of agile for most organizations today. *For Application Development & Delivery Professionals* (2011).
- [70] Dzulaiha Aryanee Putri Zainal, Rozilawati Razali, and Zulkefli Mansor. 2020. Team formation for agile software development: A review. *International Journal on Advanced Science, Engineering and Information Technology* 10, 2 (2020), 555. Retrieved May 26, 2021 from <https://doi.org/10.18517/ijaseit.10.2.10191>

Received 26 May 2021; revised 27 July 2022; accepted 29 September 2022