I Want My IoT

End-User Developers and the Internet of Things

A thesis

submitted in fulfilment

of the requirements for the degree

of

Doctor of Philosophy

in

Computer Science

at

The University of Waikato

by

Tomás García Ferrari

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Colophon

This thesis is typeset in **Montserrat** for headings and **Piazzolla** for the main body.

Montserrat was originally created by Julieta Ulanovsky, inspired by the old posters and signs in the traditional Montserrat neighbourhood of Buenos Aires.

Piazzolla, designed by Juan Pablo del Peral, is a type system intended for optimising space. It performs well and maintains readability in small point sizes and long texts, both for screens and in print.

Using these typefaces—created by colleagues in Argentina, from a community that was my first group of belonging and related to my *Alma Mater,* the *FADU — Facultad de Arquitectura, Diseño y Urbanismo,* at the University of Buenos Aires—serves as my homage to my design colleagues and my formative years in Argentina.

Abstract

The rise of the Internet of Things (IoT) has enabled seamless data capturing, processing, and analysis from the physical world and remote control of interconnected devices. However, due to complex development processes, its potential remains largely untapped by non-expert users. This thesis investigates technical, conceptual, and practical barriers that limit non-expert end-user developers from creating bespoke IoT projects.

Based on the premise that simplifying IoT development and providing targeted support can democratise access, this research explores strategies to help end users create and manage IoT projects integrated into their daily lives. To address this, the thesis explores four research areas: the barriers end-user developers encounter when creating custom IoT solutions, the effectiveness of ideation tools in supporting their development process, the challenges involved in assembling and programming IoT systems, and the strategies that enhance effective troubleshooting.

A review of the literature reveals that, while substantial research has focused on the technical dimensions of IoT, relatively little attention has been given to making these technologies accessible to non-experts. To bridge this gap, the thesis employs a mixed-methods approach, including surveys, a longitudinal case study, and three user studies to explore the experiences of non-expert users with IoT technologies. The findings highlight that the lack of intuitive tools and resources is a significant barrier to user engagement and project completion.

The results underscore the importance of user-centred design in developing IoT support tools and demonstrate that non-experts can effectively create and manage IoT projects with appropriate guidance and resources. However, challenges remain, such as enhancing troubleshooting support and developing intuitive interfaces.

This thesis argues that the future of IoT lies in designing accessible tools and systems for a broader audience, with significant implications for inclusivity and ease of use. Future research should focus on refining these tools, understanding the cognitive processes of end-user development, and exploring new methods for integrating IoT into everyday life. This work contributes to developing a more inclusive IoT ecosystem accessible to all users, regardless of their technical expertise.

Acknowledgements

I could not have reached this point without the inexhaustible support of my family. My partner in life, Carolina, and our children, Luca and Lena, have stood by me throughout this journey. My parents, Julio and Raquel, who taught me essential values in life and the importance of education. Though we live on opposite sides of the world, my siblings—Francisca and her partner, Eli, and Julito, his wife, Ivana, and my nephew, Mateo—have always been with me in spirit.

I am deeply grateful to Assoc. Prof. Judy Bowen, Prof. David Bainbridge, and Prof. Annika Hinze, whose guidance and expertise were invaluable throughout my research. I also extend my thanks to all my colleagues at the University of Waikato for their support, as well as to those beyond the University who have offered their encouragement. In particular, I am deeply grateful to those who took the time to read and provide feedback on various parts of my work as I was writing.

I would like to acknowledge the participants in the user studies, particularly one individual who dedicated months to a project and provided substantial diary entries.

Finally, completing a doctoral thesis can be likened to the Hero's Journey. It has been a long and arduous path, but I have emerged transformed and ready for new challenges. I thank you all.

No podría haber llegado hasta acá sin el enorme soporte de mi familia. Mi compañera de la vida, Carolina, nuestros hijos, Luca y Lena, siempre junto a mi. Mis padres, Julio y Raquel, que me enseñaron los valores esenciales en la vida y la importancia de la educación. Aunque vivamos en lados opuestos del mundo, mis hermanos—Francisca y su compañera, Eli, Julito y su esposa, Ivana, y mi sobrino, Mateo—siempre me han acompañado en todo.

Estoy agradecido con los Profesores Judy Bowen, David Bainbridge y Annika Hinze, cuya guía y conocimiento fueron invalorables durante mi investigación. También extiendo mi agradecimiento a todos mis colegas de la Universidad de Waikato por su apoyo, así como a aquellos más allá de la Universidad que han ofrecido su aliento. En particular, estoy profundamente agradecido a quienes se tomaron el tiempo de leer y proporcionar comentarios sobre varias partes de mi trabajo mientras lo escribía.

Quiero reconocer a quienes participaron en los estudios de usuarios, particularmente a una persona que dedicó meses al desarrollo de un proyecto, creando una bitácora sustancial.

Finalmente, completar una tesis de doctorado puede ser comparado con el Viaje del Héroe. Ha sido un camino largo y arduo, pero salgo transformado y preparado para nuevos desafíos. Les agradezco a todos.

Contents

List of Figures

List of Tables

“Once upon a time, non-experts were eager to remotely sense and manipulate the physical world according to their needs and desires. They recognised multiple operating scenarios, considering individual needs, small groups and even significant and collective social needs. Before moving forward, they scrutinised ancient texts and scriptures, seeking knowledge before it got lost. Then, they embarked on an initiatory mission, not knowing the hindrances they may encounter. All acknowledged their limitations. People could get better support. Envisioning ideas was not enough. They needed bridges to close the chasm between their talents and the real world. Multiple enterprises offered to help them, the sweet voices of sirens on the shore. Although tempting, they recognised the limitations of those gifts. Was it all in vain? Probably not. There was hope, and they wondered if they could only find it! Creators have not been paying much attention to those playing snakes and ladders. Maybe this is a call to them. Perhaps this is a desperate call to them. In the end, we all need each other. We all need each other.

— Tales of Times Forgotten

I Context

Chapter 1

Introduction

> *Computers began as cumbersome machines served by a technical elite and evolved into desktop tools that obeyed the individual. The next generation will collaborate actively with the user.*
>
> **— Lawrence G. Tesler**
> "Networked Computing in the 1990s" [167]

THE WIDESPREAD ADOPTION OF COMPUTERS in the 1970s and 1980s initiated a digital revolution, consolidated by the unprecedented expansion of global computer networks in the 1990s, triggered by the mass utilisation of the World Wide Web. In 1999, Kevin Ashton coined the term Internet of Things (IoT) to describe a world where computers gather data without human intervention [8]. Twenty-five years later, the IoT encompasses an increasing number of connected devices, many of which are not even recognizable as traditional computers. However, despite these advancements, developing IoT systems remains inaccessible for many end-user developers. Non-experts face significant barriers, such as the complexity of integrating hardware and software, which limits their ability to harness the potential of IoT technology fully. This thesis investigates what it entails for end-user developers—individuals who create or modify hardware and software for their use without formal training in programming, software engineering, or electronics—to build their own IoT projects. The study focuses on the challenges and opportunities presented by current technology and available support.

From a particular perspective, the IoT could have similarities with previous rapid developments, such as the personal computer revolution or Web 2.0 [141], where specific interventions unfolded the technology for end users. Given such a comparison, we need to know if the various components of IoT technologies are available for end users to create their own IoT systems for collecting, analysing and visualising data reasonably, allowing for the technology's widespread adoption.

This research aims to examine whether IoT technologies are usable by non-experts and identify any obstacles they may face in creating systems for data collection, sharing, mining, visualising, and understanding, as well as potential solutions to support their engagement. This process of empowering users could lead to what we refer to as *IoT 2.0*—a shift similar to what happened with Web 2.0. IoT 2.0 could involve more intuitive development platforms, better integration between hardware and software, and simplified interfaces that allow non-experts to create sophisticated systems without deep technical knowledge. Just as Web 2.0 enabled non-experts to actively contribute to and publish content on the web, IoT 2.0 would mark a transition where non-experts could harness IoT technologies to create solutions that serve their own needs and purposes with less reliance on specialised technical skills. The central question driving this research is why—when the technology is available, the cost affordable, and the systems accessible—developing simple ways of gathering, organising, and visualising data remains elusive for end users.

Despite the potential of IoT 2.0 to empower non-experts, with what is available today, developing even simple systems remains a complex task, as illustrated by the following examples.

- Developing a system that monitors an existing part of our environment (e.g., the water needed for a plant) and actuates in consequence (e.g., watering the plant as required) requires significant knowledge and technical skills. A person interested in developing such a system must know about basic electronics, computer programming, controlling services in the cloud, and more. And when off-the-shelf systems provide such functionalities, the data does not belong to the user. If the company providing the services decides to shut down, users are left with unusable devices.

- A homeowner already has surveillance cameras. Based on their work schedule, they decided to enhance this to get email messages of photos when movement is detected, but only for specific hours of certain days. They already own the cameras. They are already set up with email. Why is it that they need to throw away the current cameras they own and buy a new system that happens to have the desired functionality—which has been pre-programmed in by a tech expert—just because they did not start with the right product initially even though the equipment they already own is technically capable of this?

- When Billy is heading home on a bus from school, his backpack generates an email message to Mum and Dad. Sometimes, he manages to catch the bus that leaves 5 mins after school finishes, but other times, it is the bus that leaves 30 mins after that. Even adding a traceable digital tag to Billy's backpack, configuring a bespoke system that will fire an email when the tag moves beyond a geo-fenced area (e.g., the school), becomes a daunting task.

- Your garage door automatically opens up as you arrive home—the house lights turns on if it is the evening time (depending on the season), and the air-conditioning or heating turns off 15 minutes after a room has been left, based on the current temperature. Although the technology to enable such automation is available, configuring its behaviour remains a complex task for end-users.

Beyond the examples mentioned, IoT 2.0 could empower end users to create systems for home automation, personal health tracking, and environmental monitoring with minimal technical expertise. These applications demonstrate the transformative potential of IoT 2.0 in enabling non-experts to interact with and benefit from advanced technologies. However, a significant challenge lies in the limited interoperability of off-the-shelf products, which often fail to integrate seamlessly with devices from different vendors. As a result, end users are frequently compelled to adopt the role of developers to devise bespoke solutions for needs unaddressed by existing market offerings.

Research in this field has mainly centred on technical aspects, such as device connectivity and security, with comparatively less focus on enhancing usability for people navigating these dynamic technological environments. How to best support users in leveraging the potential of these device networks to accomplish their tasks remains unclear. [123].

As Anderson [5] explains, the IoT could be understood relative to the maker's community: *"the past ten years have been about discovering new ways to create, invent, and work together on the Web; the next ten will be about applying those lessons to the real world."* A simple inquiry: should it not be easier and more accessible? How can people with advanced knowledge of the technology provide tools for those without so many technical inclinations? The developments on the Web that Anderson mentions expanded the original idea of making it writable, not just readable. Many people contributed with a solid determination to make this process simpler [84]. It was how the Web was "born again," giving momentum to the blogosphere and what has been called "grassroots media" or "citizen journalism." There are now possibilities of applying similar concepts, extending from the idea of getting individual data (e.g., the notion of the quantified self [135]) to "citizen science", where members of the general public help to conduct scientific work either by collecting and providing data to scientists or even contributing to the analytical and distributive phases of a research project [39]. These ideas are already in place, given the spread of smartphones with the capabilities to record data, take images, and tag them with time and location. More possibilities are unfolding with the use of IoT technologies.

While IoT technology is accessible in many respects, crucial components—especially those designed for end-user accessibility—are still lacking. This presents a significant opportunity to adopt a user-centred approach, empowering individuals to acquire and control their own data. This thesis will investigate these missing components and propose strategies to simplify IoT development for end-user developers, ultimately contributing to the realization of IoT 2.0.

1.1 | Frame of Reference

In 1999, Kevin Ashton was the first to use the term Internet of Things (IoT) as the title of a presentation delivered at Procter & Gamble. He used it to describe a technological situation where computers gather data without human intervention [8]. It is generally agreed that in years to come, the IoT will be one of the significant disruptive technological forces. In 2014, ABI Research indicated that there would be 40.9 billion internet-enabled devices by 2020 [66], while Cisco had predicted in 2011 that there would be 50 billion devices connected by the same time [70]. Given this context, many companies have entered the market with different goals and objectives, from offering products and services to lowering technical barriers and easing the development of IoT projects.

It is unclear if this will be a process of emancipation or if it will create more obstacles and impediments to an already challenged population. We know that the technology behind the IoT offers an excellent chance of empowering end users, allowing them to take control of at least a portion of the massive amount of data that connected things will generate.

1.1.1 | Defining the Internet of Things

The IoT has been described as a concept in which the virtual world of information technology integrates seamlessly with the real world of things [171], the process of connecting machines, equipment, software and things through the use of the unique Internet protocol address that permits things to communicate without human intervention [156] or a world where the physical objects are seamlessly integrated into the information network and where the physical objects can become active participants in the business process [89].

Even some years after the introduction of the term, it seems that the Internet of Things is still in its infancy, similar to the state of the Internet before the World Wide Web made it indispensable for communication, business and entertainment. A similar revolution is needed to make the IoT useful [181].

In the context of this research, we understand the IoT as the scenarios where computing capability and network connectivity spread to everyday objects that are not usually considered computers. In those circumstances, objects with small power supplies, embedded sensors and addresses on the Internet can generate, exchange and consume data with minimum human intervention, operating through an information network [26, 92, 152].

Therefore, any project using a subset of these technologies without connection to a network is assumed to be outside the scope of this research. For instance, a digital pedometer that merely stores data in an internal memory card will not be considered an IoT project. Similarly, a motion sensor that triggers a local alarm without any form of data communication or remote access falls outside the scope of our definition.

In contrast, the *Good Night Lamp* is a representative example of an IoT object focused on ambient human connection: a networked lighting system where switching on a lamp in one location causes a paired lamp in another home to illuminate [52]. Another example might include an umbrella whose handle glows subtly when rain is forecast, reminding the user without requiring deliberate interaction. These examples illustrate how IoT can serve as ambient infrastructure for everyday life, offering utility, expression or reassurance without overt demands on attention.

As this research focuses on the challenges and opportunities for end-user development, we do not consider projects presented as closed systems (e.g., an internet-enabled fridge or smart handheld devices).

1.1.2 | Non-experts and End-users

Over time, professionals progress from novice to advanced beginner, competent, proficient, and, finally, expert [42, 65]. In this thesis, the term novice refers to an individual with training but little practical experience [65]. In contrast, the term non-expert defines an individual who does not possess specialised knowledge of digital technologies. A novice developer could be a recent Computer Science graduate, while a non-expert developer would be anyone undertaking a development project without formal training. In this context, non-experts are a group without training in a discipline focused on the creation of artificial things, as Simon [161] describes it: *"how to make artefacts that have desired properties and how to design."*

Non-experts face a more challenging path to expertise, not receiving initial rules and principles through formal training. Therefore, developing technology-driven solutions without specific training requires different types of assistance to realise their ideas. While a non-expert's level of expertise can increase, the time required for progress can vary significantly. Psychologists use learning curves to represent the relationship between practice and the associated behavioural change, where the curve's slope indicates the improvement speed. A steeper slope reflects faster learning, requiring less time [162]. Considering two novices on their journey to becoming experts, the one with formal training might experience a steep learning curve. At the same time, the self-taught individual is likelier to face a shallower one.

More than 20 years ago, Cypher and Halbert [40] defined the end user as a *"user of an application program,"* describing someone who is not a computer programmer. Barricelli and Valtolina [12] further elaborated that such users *"use a computer as part of daily life or daily work, but are not interested in computers per se."*

This thesis aims to explore what supports end-user developers with different levels of knowledge, particularly those at the lower end of the expertise spectrum. It is beyond the scope of this thesis to discuss how they become experts.

1.1.3 | An IoT 2.0 Conceptual Framework

A successful IoT 2.0 approach needs a more straightforward pathway for end users to develop their IoT system. This includes ease of use of existing components and simplicity in assembling. However, the assembly does not need mere simplicity and easy use of components; it also needs to provide conceptual support to the end users turned developers. Putting together an IoT project should be as simple as mapping an idea, understanding the required components and setting them together. Like building with Lego blocks, modular and straightforward elements can be organised so that simple structures are easy to make and complex arrangements are also possible.

A few elements need to be developed to reach that level of simplicity.

We will briefly describe here our concept for how such help may be provided.

1. First, end users need tools that help them map their IoT 2.0 ideas. At this stage, a user may have a general goal in mind but not a clear concept of, for example, which data needs to be captured.
2. Secondly, users need to identify conceptually which components they wish to use.
3. Thirdly, they must be guided toward the technical parts (sensors, actuators, controller, etc.) combined to implement their concept from step 2.

We propose developing card sets and accompanying web resources that guide end users through those steps. While cards for ideating and developing IoT projects already exist, all of these target expert developers and programmers. Plug-and-play sensors, actuators and controllers have been created. This means there should be no parts with open electronic connections that need soldering. Finally, there needs to be guides for assembling the elements (e.g., in the form of instructional material) and to support end-user developers when things malfunction or stop working.

1.2 | Proposal

This section presents a hypothesis that serves as a guiding principle for this thesis, centring on end-user developers and their opportunities in creating bespoke IoT projects. Four scoping questions are then introduced to direct the research.

1.2.1 | Hypothesis

We identify the following hypothesis:

> With structured support, including facilitation and conceptual tools, non-expert end-user developers can successfully design and implement bespoke IoT solutions.

1.2.2 | Scoping Questions

The following scoping questions will be further considered and will assist in defining the scope of this research. As set out below, these questions are required to substantiate the previously proposed hypothesis.

A Comprehensive Analysis – Research Question 1 (RQ1)

To explore the implications of creating a bespoke IoT project, we require a comprehensive view of all the steps and components necessary for end-user developers. We must first understand their specific challenges and available resources.

RQ1 | What barriers do end-user developers face in creating bespoke IoT projects?

This investigation involves identifying the technical and logistical barriers and evaluating the accessibility of IoT components and technologies to non-experts. Our preliminary findings, discussed in "An IoT for Everyone: Fact or Fiction?" [78] at the 32nd International BCS Human Computer Interaction Conference (Belfast, July 2018), highlight that IoT accessibility remains contentious. In case_studychap case_study**??**case_studychap case_study

: **??**, we explore these issues through a long-term study to understand the development process, potential challenges, and time needed to create a functional IoT solution, addressing our primary research question.

End-Users Support – Research Question 2 (RQ2)

We must first identify the specific needs and challenges these users face to understand and develop different tools to assist end users in ideating and creating their own IoT projects. This involves examining current tools and determining the gaps hindering the creative process. By doing so, we can design and implement more effective solutions tailored to empower end users in their IoT endeavours.

RQ2 | In what ways can targeted support during the ideation phase improve end-user developers' ability to conceptualise unique technology-driven solutions?

To address RQ2, we develop tools designed to assist end users in creating their own IoT solutions. These tools, such as decks of conceptual cards, provide guidance and inspiration throughout the development process. An initial example of these tools is the deck of cards introduced in Chapter 5: Concept Refinement and Ideation.

Enhancing Development – Research Question 3 (RQ3)

To identify and implement essential support strategies that enhance end-user developers' success in building and programming IoT projects, we must first examine their specific challenges and obstacles. This involves analysing the existing support mechanisms and determining where they fall short in facilitating development. By understanding these gaps, we can develop targeted strategies that effectively empower end-user developers, ensuring their success in navigating the complexities of IoT projects.

RQ3 | What essential support strategies enhance end-user developers' success in building and programming IoT projects?

The subsequent chapters will comprehensively explore essential support strategies for end-user developers. Drawing insights from the long-term case study presented in case_studychap case_-study**??**case_studychap case_study

: **??**, faitapchap faitap**??**faitapchap faitap

: **??** then focuses on supporting the critical prototype development stage within the IoT process.

Together, these chapters offer valuable perspectives on how tailored support can significantly enhance the success of end-user IoT projects.

Problems – Research question 4 (RQ4)

To determine the forms of support that end-user developers require for efficiently identifying and resolving issues in their IoT projects, we must first explore the common challenges and troubleshooting needs that arise during the development process. This involves assessing the current resources available to these developers and identifying gaps where additional support may be necessary. By understanding these specific needs, we can develop targeted strategies and tools that empower end-user developers to overcome obstacles more effectively, ultimately enhancing their IoT projects' overall success and efficiency.

> **RQ4** | What forms of support do end-user developers require to efficiently identify and resolve issues in their IoT projects?

In Chapter 7: Guided Troubleshooting, we present the specific support that end-user developers require to efficiently identify and resolve issues within their IoT projects. Building on the insights gained from previous chapters, where we explored the skills, tools, and strategies necessary for successful project ideation and prototype development, this chapter focuses on troubleshooting challenges. We investigate how targeted support can alleviate the frustration caused by unpredictable technical malfunctions, ensuring that end-user developers are equipped to overcome obstacles and maintain momentum in their IoT projects.

1.3 | Methods

This section presents the research methods that will be used in the context of this PhD. The research will be based on Interaction Design (ID) and Human-Computer Interaction (HCI) methods, following models oriented to research how users, designers, and technical systems interact. The users will be understood as someone with experience with or through the technology [187]. The following approaches will be utilised:

- **User-centred design,** considering the situation where users, designers and technical practitioners work together to articulate the needs and limitations of the user and create a system that addresses these elements.
- **Value sensitive design (VSD),** as a method for building technology that accounts for the values of the people who use the technology directly, as well as those whom the technology affects, either directly or indirectly [76].
- **Speculative design and design fiction,** as a future-oriented practice concerning futurology, concerned with changing reality rather than merely describing or maintaining it [68].

The following research methods will be utilised:

- **Internet-based data collection.** The process of gathering available information on public repositories – digital libraries and commercial websites – is used to compare existing

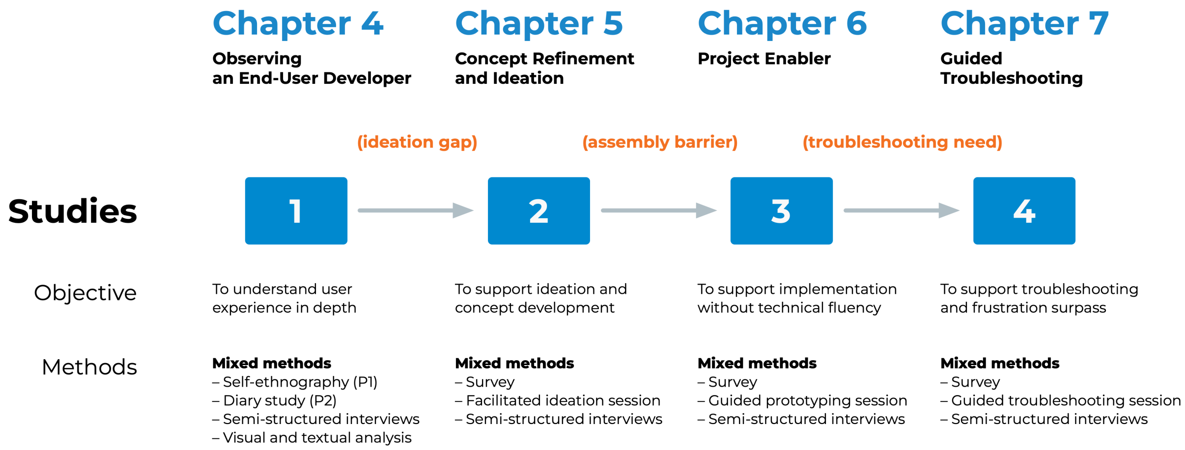


FIGURE 1.1: Studies and corresponding chapters.

technologies. This approach was used to write the paper *"An IoT for Everyone. Fact or Fiction?"* [78], where nine different technologies and platforms were evaluated against scenarios and requirements.

- **Surveys.** On-line surveys are conducted to understand how users interact with existing or fictional IoT services and devices.
- **Interviews.** Structured and unstructured interviews with users are conducted to gain insights into their needs and desires concerning IoT projects.
- **Conceptual models and cards.** Different decks of cards are created and tested with users, allowing them to develop conceptual models for Internet of Things projects. Conceptual models are tools for the understanding or teaching of systems. In constructing a system, the design should ideally be based on a conceptual model so that the image of that system seen by the user is consistent, cohesive and intelligible [139].
- **Gamification.** In recent years, we have experienced the rise of gamification. This concept indicates that game design elements in non-game contexts, products, and services can motivate desired behaviours [59]. In the context of users creating their own IoT systems, we explore different notions of gamification.
- **Experimentation.** Studying how end users with different levels of technological experience can construct simple IoT solutions. A first example of this – *Squeezy Mood*, is described in Chapter 4.

1.4 | Thesis Structure

This paragraph outlines the overall structure of the thesis, detailing how each chapter contributes to the overarching research objectives. Chapter 1: Introduction envisions the possibilities of end-user developers creating bespoke IoT projects. Chapter 2: IoT Across Scenarios presents various scenarios involving individuals, groups and significant and collective social needs. It also introduces a collection of requirements for bespoke IoT projects. Chapter 3: Supporting End-User Developers compares and contrasts the literature on what supports end-user developers aiming to create IoT projects. As presented in Figure 1.1, the findings from the long-term study in Chapter 4: Observing an End-User Developer inform the design and testing in Chapter 5: Concept Refinement and Ideation, where a method using a deck of cards helps end-user developers conceptualise IoT projects. This exploration is further expanded in Chapter 6: Project Enabler and Chapter 7: Guided Troubleshooting, which test methods to support end-user developers in assembling and troubleshooting their IoT systems. Chapter 8: Discussion discusses the insights obtained from this thesis, and, finally, Chapter 9: Conclusion presents the final remarks.

With the research context and objectives established, the thesis now transitions to Chapter 2, which examines the diverse scenarios of IoT applications. This next chapter offers valuable perspectives on the practical applications and technical requirements that influence end-user IoT development.

Chapter 2

IoT Across Scenarios

From Personal Use to Citizen Science

> “*Anyone should be able to become an inventor.*
> *We want to enable people to do what used to take*
> *two weeks in two minutes, or was not possible at all.*
>
> — **Takehiro Hagiwara**
> as quoted in *The Seattle Times*,
> “Sony Kit Helps Tinkerers Invent Sensor Devices” [169]

IN THIS CHAPTER, WE EXAMINE THREE DISTINCT SCENARIOS that illustrate the various contexts in which potential end-user developers might engage in creating Internet of Things (IoT) projects. These scenarios reveal the developers' motivations, needs, and challenges. By exploring applications that range from individual to community-level initiatives, we aim to provide a comprehensive view of the IoT landscape from the perspective of end-user innovation. Following this, we will outline the essential requirements for empowering end-user developers to successfully create and implement IoT projects, contributing to the ongoing advancement of IoT technology.

2.1 | Scenarios

In this section, we describe different circumstances where individuals or groups of people could benefit from the possibility of gathering, storing, and visualising data. The scenarios discussed in this chapter were developed through multiple brainstorming sessions, where we explored various contexts, focusing on practical applications that address real-world challenges while also considering the data flow within the *Sensing*, *Processing*, and *Actuating* space (Figure 2.1). In the exploration, common themes were emerging, helping in the process of refining ideas, providing a broad but targeted exploration.

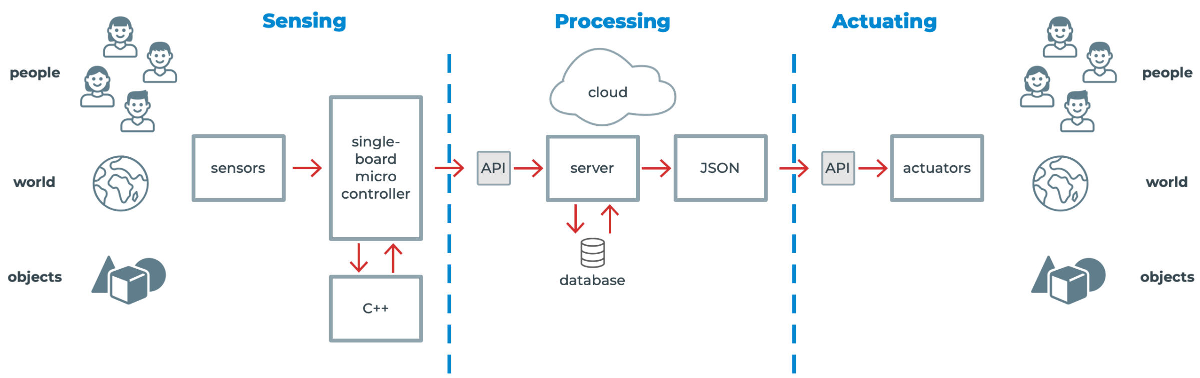


FIGURE 2.1: IoT data flow: sensors gather data from people, objects, and the environment, processed by microcontrollers and cloud servers, with APIs enabling communication and actuators responding to the data.

The scenarios presented on the following pages were selected for their common features and their feasibility for users to assemble. Following this process presented us with scenarios that were relevant and manageable in terms of end-user development, enabling an assessment that revealed the supporting needs. The three scenarios presented here, are considering diverse groups of potential users, from an individual or family-focused use to a collective and social need, encompassing various associated personas.

In the context of this research, we are not considering stand-alone and off-the-shelf digital products with specific hard-coded functionalities (e.g., an IoT kettle such as the fictional Polly—presented by Lindley et al. [112]—or a connected coffee machine like the Nespresso Prodigio[1]). In particular, we exclude products developed as closed systems with limited or no access to the raw data to the end user.

2.1.1 | Scenario A: Individual

Jasmine Kirstel is a digital designer who works from home. Over the years, she has steadily developed the capability of operating and delivering projects in collaboration with a globally distributed network of clients and collaborators. However, in her personal workspace, she works independently, relying on her judgement to manage her health and productivity. Recently, she started feeling the physical strain of long hours sitting at her desk. While she acquired an adjustable desk, she often forgets to change its height and move to a standing position when intensely focused on work.

Jasmine seeks to create an artefact that can track and monitor her desk usage, capturing data such as the height of the desk's tabletop at regular intervals. The device must be easily configurable via a web or mobile application, integrating with her existing network setup (a Wi-Fi router with Ethernet ports). Given that she works alone, the system must function unattended for long

[1] https://www.nespresso.com/prodigio/experience/desktop/#/en/uk

periods, even when her computer is off, allowing her to view captured data and receive alerts or reminders on her phone or computer. These requirements reflect the importance of seamless, low-maintenance solutions for individual users who cannot rely on others for technical support or reminders.

2.1.2 | Scenario B: Family-focused Data

The Taylors enjoy tracking their children's growth, previously marking their progress on a kitchen door frame. With a move to a new house, they now seek a more permanent and interactive way to collect and visualise this data. They envision a plug-and-play artefact that measures height, stores data, and allows easy mobile app or website access.

Drawing on D. Rose's *Animism future* [151], this artefact would mimic the comfort of a living relationship. Rather than merely tracking growth, it could "learn" the family's habits and respond intuitively, reminding them to take measurements at meaningful moments or "celebrating" milestones with custom visualisations. This approach fosters emotional engagement, making the device more than a tool—it becomes a companion in family life, transforming routine tasks into personal experiences.

2.1.3 | Scenario C: Citizen Science

The neighbours of Tamatea, in Napier (New Zealand), are unhappy about the noise levels around York Avenue. They aim to collect data to substantiate their concerns and demonstrate that the noise, particularly during night hours, has become disruptive. A simple IoT system would allow them to gather, process, and present this data to the local council.

They propose using plug-and-play artefacts distributed across the neighbourhood to measure noise levels. These devices should be easy to configure using a web or mobile application and set to collect data at regular intervals (e.g., every 15 minutes). They should operate unattended for extended periods (e.g., months) without requiring connection to the electricity grid or continuous monitoring via another device. The collected data should be accessible through a website or mobile application, utilising existing networks (e.g., 3G).

2.1.4 | Contextual Complexity of IoT Scenarios

Although the scenarios mentioned above share similar technological foundations, they differ in complexity due to their unique contexts. This is where technology intersects with human needs and desires, presenting opportunities and challenges in each case.

Jasmine has complete autonomy in integrating IoT solutions into her routine as an individual user. Unlike in a family or community setting, where tasks might be distributed, Jasmine must take sole responsibility for the design, configuration, maintenance, and interpretation of the data produced by the IoT system. Furthermore, given the specific nature of her needs, she is tasked with developing a bespoke IoT system that seamlessly fits into her workspace and routine. This process requires her to understand the technical specifications and how the system can be created to address her unique situation, reinforcing the importance of user-driven design in IoT development. This level of self-reliance aligns with the quantified self movement, as described by Lupton [119], where individuals use technology to optimise their health and productivity through personal tracking.

Drawing on D. Rose's concept of enchanted objects [151], this device would seamlessly embed computation and connectivity into her workspace, enhancing her daily routine. Like Rose's Glow-Cap—a smart pill bottle that glows to remind users to take their medication and improves adherence from 40-60% to over 90%—Jasmine's desk-monitoring system would not merely perform a task but enrich her interaction with her environment, promoting physical well-being through subtle, intuitive interactions. By offering personalised reminders, this *enchanted object* aligns with the quantified self's goal of self-optimisation through self-tracking and analysis.

Furthermore, as an individual user, Jasmine may face different challenges in engaging with this system compared to users in a group setting. Without shared accountability or collaborative troubleshooting, Jasmine's IoT device success depends solely on her motivation and ability to self-monitor. This highlights the importance of designing IoT tools for individual users that are intuitive, easy to maintain, and require minimal ongoing interaction—factors that ensure continued use without external support.

In the second scenario, a family project introduces a distinct context. The physical artefact should be easy to configure using a web or mobile application. It would allow measurements on request (e.g., with a button) and store photos alongside data. In this case, data collection and use are limited to a small group (e.g., 4 to 12 people). However, the dynamics of a family group could introduce complications. Even though all members may be non-expert developers, variations in roles naturally emerge: some will take on leadership in decision-making or technical aspects. In contrast, others will express differing opinions or preferences. These varying perspectives could slow progress or create conflicts, mainly if there are disagreements about how the artefact should be used or configured. The small size of the group can sometimes streamline discussions as fewer people are involved. Still, it also means that a single member's objections can significantly impact the project's direction. The psychological effects of these interactions—such as maintaining consensus or managing differing expectations—can hinder the smooth implementation of the project, adding an additional layer of complexity to what initially seemed like a straightforward task.

In the third proposed scenario, the project introduces a different context in which group dynamics play a more significant role. Given the broader interest of up to 500 people in this community, Participatory Design Fiction can be applied as a co-design approach to enhance the development of this IoT system [140]. By encouraging the exploration of unconventional ideas and engaging end users in the ideation phase, this method allows for creating innovative systems that better reflect the aspirations and needs of the community. However, coordinating such a large group introduces challenges in managing group dynamics. Despite sharing a common goal, variations in opinions, priorities, and levels of engagement among participants can slow down decision-making and lead to disagreements about design choices or the interpretation of the data. Unlike smaller groups, such as families, where discussions are more intimate and decisions can be made quickly, larger community groups must navigate a broader range of perspectives. The time required to reach consensus increases significantly, as does the risk of some participants feeling marginalised if their voices are not adequately heard. Additionally, the emergence of informal leaders within the group can influence the project's direction, which may or may not align with the broader community's intentions. These dynamics could hinder consensus-building and affect the community's ability to effectively deploy and maintain the IoT system, adding a layer of social complexity to the project's technical challenges.

Analysing the three scenarios reveals escalating complexity in group dynamics, which may hinder project completion. While the technological requirements—discussed in the following section—remain relatively consistent, aligning the goals and motivations of multiple participants, particularly with limited expertise, adds significant challenges.

2.2 | Requirements

Building on the scenarios discussed earlier, we identified several essential conditions that an IoT project must meet to effectively address the diverse needs of individual, family, and community users. These conditions ensure that the systems are adaptable to each scenario's unique context while maintaining usability, reliability, and engagement.

- **The things**
  A collection of plug-and-play artefacts used to measure different dimensions (e.g., distance, temperature, noise) and capture simple data points or more complex multimedia data (e.g., photographs, audio, video). In our scenarios, end users want to capture data without constructing a device from scratch.
- **Set-up and configuration**
  The ability to configure the data capture artefacts using the web or a mobile application (through a GUI). These are the systems that end users are already familiar with.
- **Interconnectivity**
  The possibility of connecting to other platforms and services (e.g., SMS). The end users we describe would like to receive alerts and notifications on systems they already use.
- **Operation**
  Having artefacts with the possibility of capturing data on request (e.g., now), or with the use of a timer (e.g., every fifteen minutes, hourly), or triggered by an event (e.g., someone passing a door). In the described scenarios, data capture instances can be triggered differently.
- **Power**
  Powered in a way that allows them to work unattended for long periods (e.g., months) without

being connected to the electricity grid, as in our scenarios, end users want to set up the system and leave it running. According to our scenarios, end-users want to use the systems without constantly considering how they are powered.

- **Independence**
  Work without the constant usage of a device that is not part of the system (e.g., a computer, a tablet or a smartphone). In our scenarios, end users want systems that do not depend on other existing devices, even if they have them.
- **Visualisation**
  Offer different ways of visualising the captured data using web and mobile interfaces. In our scenarios, end users want access to visualisations with the systems they are familiar with (e.g., web and mobile).
- **Connectivity**
  Work over commonly available networks (e.g., cellular and Wi-Fi). In our scenarios, end users want to use networks that are already available to them.

2.3 | Summary

This chapter examines Internet of Things (IoT) technology through the lens of distinct user scenarios, revealing its potential to enhance daily life in personal, domestic, and community environments. At its core, IoT offers a paradigm shift in how individuals, families, and communities interact with technology, bringing about efficiency and improved decision-making capabilities.

We began by exploring the individual use case through Jasmine Kirstel, a digital designer. Her story illustrates the need for IoT systems that adapt to personal work environments, focusing on how such technology can optimise physical conditions, like accommodating an adjustable desk, thereby enhancing productivity and comfort. While this scenario may seem simpler due to the absence of social dynamics, it places the burden of design, configuration, and troubleshooting solely on the individual, with no support network to rely on.

The family-focused scenario, represented by the Taylor family, highlights the application of IoT in a domestic setting. Here, the emphasis is on tracking and managing children's growth and development. The presence of multiple participants allows for shared responsibilities and collaborative decision-making. However, this also introduces the potential for disagreements and differing expectations, which can complicate the adoption and configuration of the technology.

In the citizen science application, we shifted our attention to a group of people in Napier, New Zealand. This scenario provided insight into how IoT can empower communities to tackle environmental concerns, such as noise pollution, through collective monitoring and data analysis. The presence of a large number of participants offers the advantage of shared effort and diverse perspectives. Yet, it also introduces challenges in coordination and reaching consensus, making the project more complex despite the increased support network.

The chapter further identified key requirements for IoT systems catering to these scenarios: ease of setup, interconnectivity, power efficiency, independence, visualisation capabilities, and robust connectivity. These requirements ensure the accessibility, reliability, and effectiveness of IoT in various real-world applications.

In conclusion, the diverse applications of IoT technology in this chapter underscore its vast potential to shape our interaction with the digital and physical world. However, the complexity of social dynamics—whether in family settings or larger community groups—highlights the need for IoT systems that are not only technically robust but also adaptable to the nuanced challenges of human interaction. From the simplicity yet self-reliance required in individual settings to the collaborative but sometimes complicated dynamics of group projects, IoT remains a pivotal technology with largely untapped possibilities for revolutionising everyday life.

In the next chapter, we will evaluate the literature and practical toolkits that support end-user developers, with a focus on existing frameworks, tools, and methods designed to empower non-experts in engaging with IoT technologies.

Chapter 3

Supporting End-User Developers

Literature Review and Practical Tools

> *The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.*
>
> **— Mark Weiser**
>
> *The Computer for the 21st Century* [183]

THIS CHAPTER EXAMINES both the existing literature and practical toolkits that provide support mechanisms for end-user developers involved in creating IoT projects. As IoT technologies become more accessible, individuals without formal training in computing or electronics are expected to develop IoT solutions. However, through a review of both academic foundations and practical resources, we argue that current mechanisms remain inadequate. The literature and the toolkits reveal significant challenges, making IoT development difficult and often impractical for non-experts.

The field of supporting IoT project development has a significant history, yet only in the past decade have researchers begun to focus explicitly on the needs of end-user developers. This chapter systematically examines how existing research addresses the full support required by end users, from the initial ideation phase to implementation, ongoing servicing, and long-term maintenance. The objective is to identify gaps in the current literature and available toolkits, suggesting areas for further research and development to better support and empower non-experts in IoT creation.

The following questions guide the review:

Question 1 | What kinds of support have been discussed and developed for end-user developers, generally and within the context of IoT?

Question 2 | What specific types of support have been created for IoT development, and how do these align with different stages of the creation process, such as ideation, prototyping, testing, and maintenance?

Question 3 | Where do the gaps exist in the current support systems for IoT development, particularly regarding their effectiveness for non-expert developers?

The chapter begins with Section 3.1, which presents the literature review and theoretical foundations shaping IoT development for non-experts. It then transitions to Section 3.2, where existing workshops and toolkits are evaluated for their effectiveness in supporting end-user developers. This evaluation aims to identify gaps and challenges in current tools, focusing on key obstacles such as technical complexity, usability issues, and steep learning curves. Finally, Section 3.3 summarises the key points discussed throughout the chapter.

3.1 | Literature Review and Theoretical Foundations

This section addresses the first guiding question: *What kinds of support have been discussed and developed for end-user developers, generally and within the context of IoT?* We review the existing literature on end-user development (EUD) and explore the theoretical and contextual foundations of support systems in the IoT domain. This analysis provides a broad understanding of the types of support available to non-expert developers.

3.1.1 | End-User Development (EUD)

This section explores the concept of End-User Development (EUD) and the broader discourse on empowering non-expert developers. This discussion lays the groundwork for understanding these developers' specific needs in the IoT context.

The emergence of End-User Development (EUD) in the 1970s and 1980s was closely tied to the miniaturisation and cost reduction of computing technologies [1, 23]. This period marked a pivotal shift where individuals without formal computing training could increasingly engage in software development. As Paternò [143] highlights, the initial focus of EUD was on graphical user interfaces (GUIs). These interfaces played a critical role in lowering the barrier to entry by making complex systems more accessible, as documented in the literature [23, 117, 142]. However, while GUIs facilitated access to desktop-based software creation, they were less effective in addressing the demands of later technological shifts, such as the complexities of mobile computing and the challenges of IoT integration. This limitation suggests that the early focus on GUI-based tools did not fully anticipate non-experts' needs as technology evolved, a gap that subsequent research in EUD has sought to address. This gap aligns with the initial question guiding this review, which aims to understand the evolution of support mechanisms for end-user developers, particularly in the context of IoT. Addressing this need requires a deeper understanding of what constitutes EUD and how it enables non-professional developers to actively shape digital artefacts. Lieberman et al. [110] define EUD as a collection of methods, techniques, and tools that empower non-professional developers to create, modify, or extend software artefacts. While centred on software, this definition captures a broader ambition: enabling non-professional developers to engage with and shape digital artefacts across varying domains.

Paternò [143] traces the evolution of EUD through distinct technological shifts, starting in the late 1980s with graphical desktop systems that met basic development needs. As the Web and open interfaces emerged, EUD evolved to support more dynamic interactions, eventually leading to the third generation of touch-based mobile devices that facilitate user-driven development activities. These shifts are not merely incremental; they signify a more profound transformation in end users' interaction with technology. Today's generation extends this trajectory as smart things and robots equipped with sensors and actuators push the boundaries of what EUD can achieve. The integration of Machine Learning and Digital Twins further exemplifies this evolution, providing non-experts with powerful tools for optimising their IoT environments, particularly in complex tasks such as energy management [35]. This ongoing development underscores the need for EUD frameworks that can keep pace with these technological advancements while addressing the growing complexities faced by non-expert users.

The rise of digital fabrication technologies, including 3D printers and laser cutters, has significantly blurred the distinctions between digital and physical artefacts [5, 116]. These advancements and the proliferation of open hardware have propelled the Maker Movement. This movement has broadened the scope of EUD, incorporating new approaches and socio-technical environments that empower end users to create digital artefacts through integrating software and hardware [72, 121]. The intersection of these technologies highlights a shift where end users are not just consumers but active creators within increasingly hybrid digital-physical ecosystems. While these tools facilitate ideation and prototyping stages, their contribution to testing and long-term maintenance remains limited. This observation directly addresses the second guiding question of this review, emphasising that while the Maker Movement has provided new avenues for early-stage development, the literature is less clear on how these tools support the entire lifecycle of IoT projects.

The motivations for project development can significantly vary between professional and non-professional developers [154]. Understanding the difference is crucial for addressing the first guiding question of this review, which concerns the kinds of support developed for end-user developers. For professionals, extrinsic motivations—such as financial rewards and career growth—often dominate, though intrinsic motivations also play a role, especially within open-source communities [90, 103, 190]. Conversely, non-professionals are primarily driven by intrinsic factors, including a desire for practical tinkering, creative expression, or a curiosity for emerging technologies [17]. Their engagement often stems from a direct and immediate interest in solving personal challenges or exploring the potential of new tools, reflecting a fundamentally different orientation towards development activities.

Education and training also distinguish professional from non-professional developers. While many individuals receive basic computing education at the primary and secondary levels [67], only a fraction pursue advanced degrees in Computer Science or Software Engineering [163]. These advanced degrees provide the theoretical knowledge necessary for professional development, including understanding data structures, computer architecture, programming languages, and software engineering methods. In contrast, end-user developers typically lack formal training in development and do not consider it their profession [17].

The literature acknowledges a continuum in the progression of professionals, from novice to expert [42, 65]. This thesis defines a novice as an individual with training but limited practical experience [65], while a non-expert refers to an individual without professional or specialised knowledge in a particular subject. A novice developer might be a recent Computer Science graduate, whereas a non-expert could be anyone attempting to develop a project without prior training.

Without formal training, non-experts face a longer progression toward expertise. Although their knowledge can increase, the time required for progress is often significantly longer. In common usage, a *"steep learning curve"* is often associated with something difficult or challenging. However, psychologists use learning curves to represent the relationship between practice and improvement, where a steeper curve technically indicates faster learning [162]. A novice with formal training may experience a steep learning curve, implying faster progress, while a self-taught individual may encounter a much shallower curve, indicating slower, more gradual improvement.

This thesis aims to explore the support mechanisms available to end-user developers with varying levels of knowledge, focusing on those at the lower end of the expertise spectrum. End-user developers often overlap with non-experts, as they generally lack formal training and engage in development activities driven by personal needs or interests rather than professional obligations. Recognising this overlap is crucial to understanding their challenges and the types of support that can empower them in IoT development. It is beyond the scope of this thesis to discuss how these individuals transition to becoming experts.

Previous research has highlighted the high complexity of IoT ecosystems [12]. Considering software requirements alone, the coexistence of diverse devices, protocols, architectures, and programming languages demands knowledge across various areas [32]. End-user developers could benefit from adequate tools for creating IoT projects [144]. However, in the early days of the IoT, research primarily focused on technical considerations such as connectivity and security [123]. As discussed in Chapter 1, the focus has not been on making the technology more usable or helping end-user developers fully realise the potential of IoT.

3.1.2 | Supporting Structures in IoT Development

This section explores the essential structures and frameworks that underpin and support End-User Development (EUD) in IoT projects, focusing on how these structures address the unique challenges faced by non-expert developers. It covers the critical role of documentation and community resources, such as online forums, in facilitating knowledge sharing and collaborative problem-solving. Additionally, key conceptual frameworks, such as meta-design and semiotic engineering, are introduced, providing theoretical foundations for creating socio-technical systems that empower end users to engage in continuous, meaningful development processes.

Documentation is essential for programmers to learn new software, create digital products, and integrate hardware and software components. Nassif and Robillard [134] highlight the relevance of the area by noting that various techniques have been proposed to automate documentation creation [24, 87, 125]. For documentation to be adequate, it must include relevant information and present it in a user-friendly format. As documentation grows in complexity and aims at a broader audience, its effectiveness depends on addressing issues such as vocabulary, technical jargon, and lack of context [3]. Despite advances in documentation techniques for software systems, a notable gap exists in documentation tailored explicitly for IoT development, particularly for end-user developers. While considerable research has focused on automated documentation processes and general best practices, there appears to be a lack of concentrated exploration on optimising documentation to meet the needs of non-expert developers working with IoT technologies. This absence highlights an opportunity for further investigation into how documentation can be effectively designed and utilised to support end-user developers in the IoT domain.

A *community of practice* (CoP) is a group of individuals who share a common interest or profession and engage in collective learning and knowledge-sharing through regular interactions. These communities are characterised by a shared domain of interest, a commitment to advancing knowledge within that domain, and the development of a shared repertoire of resources, such as experiences, stories, tools, and strategies. Members learn from one another by participating in activities, discussions, and collaborative problem-solving, often leading to the refinement of skills and the generation of new ideas. Over time, these interactions foster a sense of identity and belonging among members as they contribute to and benefit from the community's collective expertise. This concept is widely applied in various fields, including education, organisational development, and technology, where fostering a community of practice can enhance professional development, innovation, and the effective dissemination of knowledge. Forums such as *Stack Overflow* and Reddit's *r/programming* exemplify CoPs by enabling programmers to exchange knowledge, collaboratively solve problems, and enhance their skills. Similarly, technology-specific forums like those for Arduino or Raspberry Pi also function as CoPs, providing platforms for users to assist each other with technical issues and project-related advice. In educational contexts, CoPs have been effectively applied, particularly in programming courses, to help bridge student skill gaps. As Mercieca [126] observed, the concept of a CoP originally emerged from studies on apprenticeships, where learning a particular practice involved integrating into the community that upheld and practised that expertise. This approach can enhance programming skills by simulating the collaborative dynamics of real-world software development teams, helping students progress from observation to active participation. However, while CoPs naturally evolve among experienced developers, deliberate efforts are needed to extend these benefits to end-user developers, who may find such spaces unfamiliar or intimidating. Ensuring that all learners, including end-user developers, can engage in collaborative, supportive environments is crucial for their growth and development [28].

Meta-Design is a framework for creating socio-technical environments that empower users to engage in continuous systems development rather than being restricted to predefined solutions [73]. It focuses on building both the technical and social infrastructures that allow users to co-design systems continuously. This approach is particularly relevant to IoT, where the integration of diverse hardware and software requires ongoing adaptation as new technologies emerge and user needs evolve. Meta-Design enables end-user developers—who may lack formal training but possess domain-specific knowledge—to actively modify and refine IoT systems over time [34, 189]. Although the framework has been established for many years [74], there is no evidence in the literature of its application specifically to IoT projects by end-user developers.

In HCI, Semiotic Engineering is a theoretical framework that interprets human-computer interaction as a communication process between system designers and users [49]. It emphasises how system interfaces convey the designers' intentions and understanding of user needs, making it a valuable tool for designing systems that non-expert users can adapt to their contexts. For IoT, this is crucial as it guides the creation of interfaces that support user appropriation and customisation. Using Semiotic Engineering, Ferrari et al. [71] discusses the concept of appropriation of IoT technology. Their study involved novice users of smart devices—recruited from undergraduate courses in Computer Science—focusing on how they adopt, use, and adapt to IoT technology over time. Appropriation – *"the way in which technologies are adopted, adapted and incorporated into working practices"* [63] – is complex, involving various interactions across different contexts and times, making it challenging to evaluate and design for. Using Semiotic Engineering to analyse the data, the researchers identified communicative breakdowns at the appropriation level. These insights reveal design principles that can help facilitate the successful adoption and adaptation of IoT technology by non-experts, addressing challenges that arise when systems fail to align with users' evolving needs.

While Meta-Design offers a flexible approach for continuous adaptation, it lacks the prescriptive guidance seen in frameworks like Semiotic Engineering. This suggests a trade-off between adaptability and ease of use that IoT-specific support systems have yet to fully address.

The discussion on the context and theoretical foundations of IoT development for non-experts illustrates that end-user development (EUD) has evolved significantly, now encompassing both hardware and software in creating digital products. The distinctions between professionals and non-professionals, alongside the various motivations driving their development activities, highlight the unique challenges non-experts face. Supporting these individuals effectively requires considering their learning processes and the structures that facilitate their engagement with IoT technologies. While the value of documentation, communities of practice, and frameworks such as meta-design and semiotic engineering could contribute to creating an empowering ecosystem, the literature does not provide clear evidence that this is happening. These elements underscore the importance of a holistic approach to supporting end-user developers, ensuring they have the necessary tools, resources, and theoretical grounding to navigate the complexities of IoT projects.

3.1.3 | IoT Development Process

This subsection investigates the IoT development process by reviewing the literature on the support mechanisms available for end-user developers across various stages. By focusing on ideation, prototyping, and troubleshooting, this subsection assesses how well the existing literature addresses the unique needs of non-expert developers. The goal is to identify the strengths and gaps in the support provided at each phase, offering insights into where further improvements are necessary to enhance the development experience for end users.

Developers can use sequential or iterative methodologies to create hardware and software artefacts. Identified stages of the development process include research, planning, design, development, testing, setup, and maintenance [55, 155]. Existing research also identifies approaches to software development that incorporate *evolutionary application development*, which challenges the conventional view of design-before-use and includes end-user developers in the process [110].

Ideation

Ideation is the phase where creators generate ideas using methods such as brainstorming, mind-mapping, or storyboarding [43]. Good ideas often result from generating numerous concepts [97]. Ideation methods stimulate divergent thinking, essential for generating creative solutions.

The Design Council's *Double Diamond* and IDEO's *Design Thinking* methodologies emphasise the importance of divergent thinking in ideation [20, 36]. Physical objects, such as blocks, tokens, or cards, can help foster this type of thinking [91]. Participants of ideation sessions benefit from using tangible cards as sources of inspiration [115].

Toolkits are defined by Ledo et al. [105] as *generative platforms designed to create new interactive artefacts, provide easy access to complex algorithms, enable fast prototyping of software and hardware interfaces, and enable creative exploration of design spaces.* Recent research has produced several IoT ideation toolkits, many of which consist of decks of cards, while others include complete tabletop games with boards, cards, and tokens [102].

To understand how toolkits support the creation of IoT projects, ten toolkits produced between 2014 and 2018 were identified: *Better IoT Know Cards, Co-create the IoT, IoT Design Deck, IoT Design Kit, IoT Ideation Cards, IoT Service Kit, Karakuri IoT, KnownCards, Mapping the IoT,* and *Tiles IoT Toolkit* (Table 3.1). These toolkits were organised using the categories proposed by Wölfel and Merritt [186], which include Intended Purpose Scope, Duration of Use, System or Methodology of Use, Customisation, and Formal Qualities. In a subsequent section, we will review and compare these toolkits within the context of the goals of our investigation, analysing

[1] https://sites.google.com/studiodott.be/research/iot-ideation-cards
[2] https://karakuriiot.com/
[3] https://know-cards.myshopify.com/
[4] http://mappingtheiot.polimi.it/
[5] http://www.stembertdesign.com/cocreation-and-the-iot.html
[6] https://iotservicekit.com/
[7] https://www.iotdesigndeck.com/
[8] https://iotdesignkit.studiodott.be/
[9] https://www.tilestoolkit.io/
[10] https://github.com/betteriot/betteriot-knowcards

TABLE 3.1: IoT toolkits.

Toolkit	Goal	Elements	Publications	Cards	Boards	Tokens	Extra
General/Repository cards							
IoT Ideation Cards[1]	Thinking new products and services	54 cards in 4 categories		✓			
Karakuri IoT[2]	Ideating and presenting conceptual solutions	27 cards, 5 joker cards	[133, 188]	✓			
KnownCards[3]	Brainstorming and definition	162 cards in 4 categories	[9, 10]	✓			
Mapping the IoT[4]	As reference, for brainstorming, concept definition and evaluation	70 cards, four activity guides, eight analysis cards and a features map	[7, 177, 180, 179, 178]	✓	✓		
Participatory design cards							
Co-create the IoT[5]	Guide innovation with a focus on user-engagement	cards, templates, signs	[64, 164, 175]	✓	✓	✓	
IoT Service Kit[6]	Co-create IoT experiences	100 cards, 4 boards, 11 tokens		✓	✓	✓	
IoT Design Deck[7]	Co-design	cards	[60, 61]	✓	✓		
IoT Design Kit[8]	Defining ideas	13 design canvases and 3 decks of cards	[47, 48]	✓	✓		
Tiles IoT Toolkit[9]	Support creative thinking, provide critical lenses to analyse outcomes, support transforming ideas into prototypes	110 cards, one board	[81, 83, 120, 124, 127, 128, 129]	✓	✓		✓
Context-specific/Agenda-driven cards							
Better IoT Know Cards[10]	Test industry interest in the principles of an open IoT certification mark	33 cards, two posters		✓			

how well they align with the needs identified in the previous scenarios and how they support end-user developers in creating and deploying IoT solutions. This comparative analysis will provide a clearer understanding of each toolkit's strengths and limitations in addressing the specific challenges faced by non-experts in the development of bespoke IoT projects.

Prototyping

The term *prototyping* carries multiple meanings across different disciplines. In HCI, UX/UI, and industrial design, a prototype serves as a tangible representation of a design idea, with each field emphasising different aspects of its creation and use. From a Human-Computer Interaction (HCI) perspective, a prototype is often used to explore and validate interactions between users and digital systems [150]. These prototypes are critical for usability testing, allowing researchers and developers to observe user engagement with a system's functionality, ranging from simple wireframes to more sophisticated, interactive models aimed at refining system requirements.

In UX/UI (User Experience/User Interface) design, the focus shifts to creating prototypes that simulate the user experience and interface design [29]. These prototypes help evaluate both usability and aesthetic appeal, ensuring alignment with user needs and behaviours. UX/UI prototypes typically include wireframes, mockups, or interactive models that illustrate user flows and design coherence, tested for functionality, visual design, and ease of use.

In industrial design, prototypes are usually physical models that demonstrate the form, function, and usability of a product [38]. These models allow designers to test materials, ergonomics, and manufacturability, ranging from low-fidelity models to fully functional products. Industrial design prototypes are crucial for assessing how a product interacts with users in real-world contexts, focusing on both appearance and practicality.

Lim et al. [111] explain that prototypes not only support evaluation but also play a generative role, enabling designers to reflect on their process and explore a design space. Prototyping is a crucial phase in the IoT development process, where initial concepts are transformed into functional models. This stage presents significant challenges for end-user developers due to the complexity of integrating hardware, software, and network components. While the end-user developers in our study may not iterate and refine their artefacts extensively—often stopping once the artefact functions as intended—we adopt this broader understanding of prototyping to better capture the creative process. In this thesis, we use the term to describe the process of creating a tangible artefact that embodies design ideas.

Despite the growing availability of toolkits and methodologies designed to assist in IoT development, the prototyping phase remains significantly under-supported, with many existing resources failing to adequately address the needs of non-experts during this critical stage. Most research has focused on isolated aspects such as customisation [53] or hardware development [146], without addressing the full spectrum of challenges non-experts face. Many published studies have either not included end-user testing [99, 148] or have been conducted under conditions that may not generalise [157].

Key challenges include:

- **Complexity of Integration:** Non-experts struggle with the technical intricacies of integrating IoT components, such as sensors, actuators, and communication protocols. This complexity is exacerbated by the lack of user-friendly tools.
- **Limited User Involvement:** Studies often overlook end-user involvement in prototyping, resulting in solutions that do not fully address usability concerns.
- **Insufficient Guidance:** Existing resources lack detailed guidance on debugging, iterative testing, and refinement, leaving non-experts without the necessary support to progress from conceptual ideas to functional prototypes.

F. V. Gianni [83] highlights the significant difficulties non-experts face when moving from ideation to prototyping, noting that while some tools assist in this phase, they often remain incomplete or overly complex, requiring further development to be truly effective for non-experts.

Troubleshooting

Troubleshooting represents a critical aspect of the IoT development process, where non-expert developers diagnose and resolve technical issues that arise during the creation and deployment of IoT solutions. Due to the inherent complexity of IoT systems, which involve diverse hardware, software, and network components, troubleshooting is a multifaceted challenge for non-expert developers.

In her research, Booth [18, 19] explores how end-user developers can be supported in overcoming development challenges through targeted assistance and support mechanisms. Her studies offer insights into troubleshooting behaviours, particularly with circuit bugs, and suggest potential tools to enhance problem resolution, such as physical card-based tools.

Despite the development of support tools, significant gaps remain in the troubleshooting phase, particularly regarding the guidance provided to non-experts. Future research must explore hybrid physical-digital approaches to better support novice developers, including more detailed guidance and structured troubleshooting methodologies.

In conclusion, while some progress has been made in developing support tools for the IoT development process, significant gaps remain in areas such as prototyping and troubleshooting. Addressing these gaps is essential for empowering non-experts to effectively develop, test, and refine their IoT projects.

3.2 | Review of Practical Toolkits and Workshops

This section answers the second guiding question: *What specific types of support have been created for IoT development, and how do these align with different stages of the creation process, such as ideation, prototyping, testing, and maintenance?* Here, we examine practical toolkits and workshops designed for end-user IoT developers. We evaluate how these tools align with the different stages of the IoT development process identified earlier.

3.2.1 | Workshops in IoT Development

This subsection presents workshops and IoT development techniques used to assist non-expert developers in creating and managing IoT projects. The discussion compares and contrasts these approaches, examining their effectiveness in supporting end-user developers. The aim is to highlight the strengths and limitations of existing approaches and identify areas where further innovation is needed.

Numerous studies on end-user development for IoT employ a workshop format to explore support strategies for non-expert developers [4, 54, 62, 79, 82, 168]. Some of these focus on researcher and practitioner discussions to develop methodologies for supporting end-user developers, yet they do not involve actual end users [54, 62, 168]. In other cases, workshops include end users, though often research team members create the prototypes instead of participants [4], or participants are not required to build a prototype [79].

One notable example is the work of F. Gianni et al. [82], who presented RapIoT, a software toolkit for non-expert developers, during a workshop. However, the study found that participants faced difficulties understanding JavaScript syntax, and the study did not report a proven solution for overcoming these challenges.

In addition to workshops, several IoT development techniques have been proposed to support non-expert developers. These include *Explanatory Debugging*, the *Jigsaw Metaphor*, *Modular*

Electronics, *Programming by Example (PbE)*, *Trigger-Action Programming*, and *Visual Programming*. Each of these techniques offers unique approaches to assisting developers, but they also present specific challenges.

Explanatory Debugging aims to empower end users to personalise machine learning systems with greater effectiveness and efficiency [100]. Building on this idea, M. Burnett and Kulesza [22] suggest that this technique could extend to IoT systems, offering potential for customisation, control, and issue resolution—though further user studies are necessary to validate its efficacy in this domain. A related approach, the Jigsaw Metaphor, conceptualises block programming using puzzle-like pieces that interlock to form complete programs [33, 44]. While this metaphor simplifies the coding process for beginners, it can become cumbersome when applied to more intricate IoT tasks.

Moving from software to hardware, Modular Electronics introduces a physical parallel to the jigsaw approach. It allows developers to use electronic components like interchangeable LEGO bricks, lowering the technical barrier for engaging with hardware [146]. However, while this approach addresses hardware challenges, it does not fully resolve issues related to software integration. Similarly, Programming by Example (PbE), sometimes referred to as Programming by Demonstration (PbD), allows users to craft programs by performing actions that the system subsequently replicates [88, 101, 109]. Despite its potential for simplicity, studies like those of Li et al. [108] highlight its limitations, especially in mobile-based IoT automation scenarios.

Trigger-Action Programming serves as a user-friendly approach to configure system behaviours by linking specific triggers with actions, such as "if motion is detected, then turn on the lights" [172, 173]. IFTTT (If This Then That) is a prominent example of this approach [31, 173]. Nonetheless, research has revealed that users frequently experience challenges when debugging these IF-THEN rules [33, 122], pointing to a broader need for improved support mechanisms [30, 31].

In contrast, Visual Programming shifts the focus towards graphical manipulation of code through a visual programming language (VPL), replacing traditional text-based coding with an intuitive interface [137]. Well-known examples, such as Scratch, Alice, and Greenfoot, trace their origins back to foundational educational tools like Papert's Turtle Graphics and Pattis' Karel the Robot [174]. Nonetheless, the application of visual programming within IoT has shown mixed results. Even simple tasks, such as making an LED blink, may require extensive graphical configurations [147], and key research gaps persist in the domain [160].

Despite the promise of these techniques, recent studies have not convincingly demonstrated through user studies how *Explanatory Debugging*, the *Jigsaw Metaphor*, *Modular Electronics*, *PbE*, *Trigger-Action Programming*, or *Visual Programming* can effectively support end-user developers in creating or tailoring custom IoT projects. More comprehensive evaluations are needed to identify where these techniques succeed and where further refinement is required to empower non-expert developers fully.

TABLE 3.2: IoT toolkits end-user developer acceptability.

Toolkit	Acceptability
General/Repository cards	
IoT Ideation Cards	+/-
Karakuri IoT	+/-
KnownCards	✗
Mapping the IoT	✗
Participatory design cards	
Co-create the IoT	+/-
IoT Service Kit	+/-
IoT Design Deck	+/-
IoT Design Kit	✓
Tiles IoT Toolkit	✓
Context-specific/Agenda-driven cards	
Better IoT Know Cards	✗

✗ Not acceptable +/- Might be acceptable ✓ Acceptable

3.2.2 | Comparative Analysis of Toolkits

Developing IoT projects presents unique challenges for non-expert users who may lack formal training in the necessary technologies. This subsection compares the existing development and support systems for IoT, focusing on their effectiveness for end-user developers. By evaluating the usability and suitability of these systems, we aim to highlight their contributions and identify how they address gaps in existing research. This comparative analysis also advances the understanding of IoT development for non-experts.

Following the introduction of ten toolkits designed to support IoT project development in Subsection 3.1.3: IoT Development Process, we assess them based on their acceptability for end-user developers. Acceptability refers to how well a toolkit accommodates individuals with limited technical knowledge, especially in avoiding specialised terminology or complex concepts. This evaluation demonstrates how these toolkits align with or diverge from the needs of non-expert developers, contributing to the broader comparative analysis of development and support systems.

Table 3.2 presents an overview of the toolkits and their acceptability ranking for end-user developers. Toolkits with high acceptability are then evaluated in-depth, contrasting their strengths across the following criteria:

1. Supporting the definition and scoping of a problem.
2. Supporting the ideation of an IoT solution.
3. Supporting the realisation of a functional IoT project.

TABLE 3.3: Cards suits included in the two kits.

IoT Design Kit	Tiles IoT Kit
Environment (17)	Feedback (10)
Interaction (2)	Human actions (9)
Object (10) 🔴	Sensors (12)
Person (13) 🔵	Services (15)
	Things (30) 🔴
Misc (Expectations) (22)	Missions (14)
Wildcards (11)	Criteria (10)
System Map (2)	Personas (9) 🔵
	Scenarios (12)

Colour dots indicate groups of cards that are similar.

According to the criteria presented in the preceding paragraph, the two toolkits identified as most acceptable for end-user developers were *IoT Design Kit* and *Tiles IoT Toolkit*. We compare their characteristics in Table 3.3, Table 3.4, Table 3.5 and Table 3.6. Both kits provide distinct approaches, but they are different in several ways. The IoT Design Kit offers extensive canvases and cards that help users define their projects comprehensively, though the cards provided may not always fit the specific needs of non-experts. On the other hand, the Tiles IoT Toolkit uses a single board and a more streamlined approach, offering a seven-step methodology that guides users through the creation of an IoT project.

In Table 3.7, the methodologies of these two kits are ranked according to their relevance for non-expert developers. Although both kits contribute significantly to supporting problem definition and ideation, neither offers a comprehensive solution for realising functional IoT projects, thus leaving a gap in fully supporting end-user developers from ideation to implementation.

In conclusion, while the IoT Design Kit and Tiles IoT Toolkit present valuable components for ideation and conceptualisation, their lack of guidance for moving from concept to practical implementation limits their usefulness for non-expert developers.

3.2.3 | Identified Gaps and Challenges

Despite the availability of a variety of toolkits designed to aid in IoT development, significant gaps remain, particularly when it comes to helping non-expert developers transition from initial ideation to the creation of functional prototypes. This subsection aims to explore these gaps in greater detail by identifying and analysing the primary obstacles non-expert developers encounter. These challenges include not only the inherent technical complexity of IoT systems but also issues related to usability, which can hinder effective interaction with development tools. Additionally,

TABLE 3.4: Person and Personas cards enumeration.

similarity	IoT Design Kit | Person	Tiles IoT Kit | Personas
✓	wildcard	Custom Persona
✓	Child	Child
✓	Me	Yourself
✓	Senior	Elderly
	Cousin	Construction worker
	Coworker	Disabled
	Doctor	Emergency worker
	Friend	Refugee
	Parent	Tourist
	Police officer	
	Uncle	
	Youngster	

the learning associated with mastering IoT technologies further complicates the process, making it difficult for non-experts to bring their ideas to life in a tangible and functional form.

While the comparative analysis in Subsection 3.2.2: Comparative Analysis of Toolkits demonstrated the strengths of the IoT Design Kit and Tiles IoT Toolkit in terms of ideation and problem scoping, both toolkits fall short in supporting non-experts through the entire IoT development lifecycle. Neither kit adequately addresses the technical challenges that arise during the realisation phase, such as integrating hardware and software components or debugging IoT systems.

Various development techniques, such as *Explanatory Debugging* and *Visual Programming*, have been proposed to support the development process. However, recent studies do not convincingly demonstrate that these techniques successfully bridge the gap between ideation and realisation for non-expert users. In their opinion paper, M. Burnett and Kulesza [22] discuss explanatory debugging as a conceptual method to assist with customising IoT systems, though their work lacks empirical results or formal evaluations. Consequently, user studies in this domain remain incomplete. Additionally, research on visual programming for IoT has shown that, while it may simplify certain aspects of development, it often fails to offer substantial benefits for more complex tasks, such as integrating sensors [147, 160].

Another key challenge lies in the lack of sufficient guidance for non-expert developers during the troubleshooting phase of IoT development. Tools like the RapIoT toolkit, introduced by F. Gianni et al. [82], attempt to address this issue, but participants in related workshops reported difficulties in understanding JavaScript syntax, and no proven solution was found.

TABLE 3.5: Objects and Things cards enumeration.

similarity	IoT Design Kit | Object	Tiles IoT Kit | Things*
✓	Bike	Bike
✓	Car	Car
✓	Clothing	Clothing
✓	Furniture	Furniture
✓	Pen	Pen or pencil
✓	Wallet	Wallet
✓	wildcard	wildcard
	Bottle	Bench
	Dumbbell	Boat
		Building
		Camera
		Coffee cup
		Eyewear
		Headgear
		Jewelry
		Keychain
		Luggage
		Medication
		Office desk
		Pets
		Piggy bank
		Plant
		Public bin
		Public transport
		Refrigerator
		Shoe
		Shower
		Sport equipment
		Stove
		Street

* Cards on this table are collected from different versions of the Tiles IoT Kit.

In summary, while significant progress has been made in developing tools and methodologies to support IoT development, several gaps remain in helping non-expert developers realise functional projects. These gaps are particularly evident in the prototyping and troubleshooting phases, where the current tools fail to provide sufficient support for integrating diverse components and resolving technical issues. Further innovation is needed to develop solutions that guide non-experts through the entire IoT development lifecycle, from ideation to implementation.

3.3 | Summary

This chapter has provided a detailed review of the literature and practical toolkits that offer support mechanisms for end-user developers working on IoT projects. We have addressed the guiding questions posed at the outset by examining theoretical frameworks, practical toolkits, and the challenges faced at various stages of the IoT development process. The following sections synthesise these insights, highlighting the contributions and gaps identified through the analysis.

TABLE 3.6: Boards/Canvases enumeration.

	IoT Design Kit		Tiles IoT Kit
+/-	Intro	✗	Select a persona and scenario
+/-	Expectations	+/-	Refine the mission
✓	Frame Idea	✓	Select objects central to the user
✓	Frame Problem	✓	Define triggering actions
✗	Frame Product	✓	Define feedback
+/-	Frame Tech	✓	Flesh out the idea in a storyboard
✗	Understand Journey	✗	Reflect and improve
✗	Understand Lifecycle		
+/-	Understand Tech		
✗	Opportunity Mapping		
✗	Stress Test		

✗ Not relevant +/- Might be relevant ✓ Relevant

TABLE 3.7: Boards/Canvases comparison.

Criteria	IoT Design Kit		Tiles IoT Kit	
High level definition	✓✓	Canvases to frame the problem, the idea and the stakeholders	✓✓	Comprehensive board with scenarios, missions, objects, triggers, feedback
Technical definition	✗	Assumes pre-existing knowledge on technology	+/-	Includes cards with sensors and services
Realisation	✗✗	Not supported	✗✗	Not supported

✗✗ Not at all ✗ Poor +/- Fair ✓ Good ✓✓ Very good

At the beginning of this chapter, three key questions guided our review:

Question 1 | What kinds of support have been discussed and developed for end-user developers, generally and within the context of IoT?

Question 2 | What specific types of support have been created for IoT development, and how do these align with different stages of the creation process, such as ideation, prototyping, testing, and maintenance?

Question 3 | Where do the gaps exist in the current support systems for IoT development, particularly regarding their effectiveness for non-expert developers?

Support for IoT End-User Developers

The review has highlighted the growing body of literature aimed at supporting end-user developers in IoT, but it also underscores several critical gaps. Theoretical frameworks such as End-User Development (EUD) [23, 143], Meta-Design [73], and Semiotic Engineering [49, 71] provide the conceptual foundations for non-expert engagement in IoT development. However, practical implementations of these frameworks tailored to IoT are still limited. Although toolkits like the *Tiles IoT Toolkit* and *IoT Design Kit* show promise, especially in the ideation phase, they do not offer sufficient guidance for the realisation of fully functional IoT projects. The gap between ideation and implementation remains a significant challenge for non-expert developers.

Development Stages and Support

Various toolkits and methodologies exist to support different stages of IoT development, particularly in the early stages. For example, ideation is well-supported by tangible toolkits such as cards, boards, and tokens that help non-experts brainstorm and define IoT solutions. The *Tiles IoT Toolkit* [83, 120, 128] and *IoT Design Kit* [47, 48] are particularly effective during this phase. However, as F. Gianni et al. [81] and F. V. Gianni [83] noted, non-expert developers face significant difficulties in transitioning from ideation to prototyping, which highlights a critical gap in the literature.

As discussed in Subsection 3.1.3: IoT Development Process, the prototyping phase presents several challenges for non-experts, who often lack the technical knowledge required to integrate hardware, software, and networking components effectively. Furthermore, the literature offers limited support for the later stages of IoT development, such as maintenance and ongoing tailoring, which are crucial for long-term project success. Current tools tend to focus on ideation and early development but do not cover the entire IoT project lifecycle.

Gaps in IoT Support

This review has identified several critical gaps in the support systems available for IoT development, particularly in relation to non-expert developers. While there are numerous toolkits designed to assist in the ideation and early development stages, such as the *Tiles IoT Toolkit* and *IoT Design Kit*, few tools provide comprehensive guidance throughout the entire development lifecycle. The prototyping and maintenance phases, in particular, are under-supported. Additionally, Booth [19]'s research on troubleshooting highlights the need for more robust and long-term support mechanisms, which are currently lacking in the literature.

Empirical studies involving non-expert developers are also scarce, making it difficult to evaluate the effectiveness of the existing tools and frameworks. Further research is needed to involve non-experts in testing and validating these support systems, ensuring they address the unique challenges faced by this user group.

Concluding Remarks

In summary, this literature review has revealed a growing but incomplete body of research aimed at supporting non-expert developers in IoT. While existing frameworks and toolkits provide a strong foundation, they fail to address the full spectrum of challenges associated with IoT development, particularly in the transition from ideation to implementation and ongoing project maintenance. The findings from this review highlight the need for more comprehensive, user-friendly tools and methodologies that can guide non-experts throughout the entire lifecycle of IoT development.

These insights lay the foundation for the subsequent chapters of this thesis, where we will explore novel solutions to address the identified gaps. The next chapter will present a case study involving an end-user developer, which will provide further insights into the practical challenges faced by non-experts and opportunities for developing more effective support systems.

II Nucleus

Chapter 4

Observing an End-User Developer

A Longitudinal User Study

> *When danger approaches, Sting glows blue,*
> *anticipating its own need and use.*
> *It is a trusty weapon, an infallible*
> *warning system, a handsome object,*
> *and a fantastic companion—for a hobbit.*
>
> — **David Rose**
> *Enchanted Objects* [151]

UNDERSTANDING THE SPECIFIC CHALLENGES faced by end-user developers in creating bespoke IoT projects is crucial for improving the accessibility of IoT technologies. This chapter addresses the first Research Question posed in Chapter 1:

> **RQ1** | What barriers do end-user developers face in creating bespoke IoT projects?

In Chapter 2, we explained that the components needed to create a custom IoT project are available and affordable. We hypothesize that, while a non-expert can technically undertake such a project, the number and complexity of challenges they face make completing the project exceedingly difficult.

To address RQ1, we conducted a long-term observation over several months in 2017 and 2018, tracking an end-user developer as they created an IoT project. The objectives of this research were:

1. Understand the time frame required to bring a custom IoT project to an acceptable functional state,
2. Comprehend the different phases that a non-expert must navigate to complete their plan, and
3. Identify obstacles that could stop or delay the project's progress.

FIGURE 4.1: Mood Squeezer: squeeze box and digital floor display
Reproduced from *Sarah Gallacher*, https://sarahgallacher.com/.

The rest of this chapter is organised as follows: Section 4.1: Study Design presents the planning phase of our user study, including a preliminary study with the researcher as the sole participant. Section 4.2: Study Method explains the methodology employed in the user study. Section 4.3: Analysis provides an in-depth analysis of the data gathered during the study. Section 4.4: Discussion explores the insights derived from the analysis. Finally, Section 4.5: Summary summarises this chapter's main findings and their implications for the subsequent user studies.

4.1 | Study Design

This section provides an overview of the approach taken to explore how non-experts can develop bespoke Internet of Things (IoT) projects. By examining both the planning and execution phases of a project, we highlight the practical considerations and methodological strategies employed.

The study, as it is presented in this chapter, is divided into two stages: the *Preliminary Study*, an initial proof of concept conducted by a researcher (P1), and the *Main Study*, where an end-user developer (P2) builds upon the initial stage. This dual-stage process seeks to identify key factors that enable non-experts to effectively utilise IoT technologies, contributing valuable insights to the broader objectives of this thesis.

Our inspiration for this study came from the *Mood Squeezer* project by Gallacher et al. [77]. In that project, a group of people can indicate their current mood via the Squeeze Box (Figure 4.1 (left)), while an aggregation of their inputs is shown as a floor light installation (Figure 4.1 (right)). The Mood Squeezer utilises a group of technologies that can be considered IoT. In previous research, we have noted that *"IoT are hardware/software systems that are embedded in everyday objects, communicating via the Internet"* [78]. However, Gallacher et al. do not explicitly mention the use of the Internet or the Internet of Things (IoT); their study could have been conducted using a Local Area Network (LAN).

This research examines the barriers non-expert end-user developers face when creating bespoke IoT projects, including technical challenges, unintuitive tools, insufficient resources, and gaps in programming and system assembly knowledge.

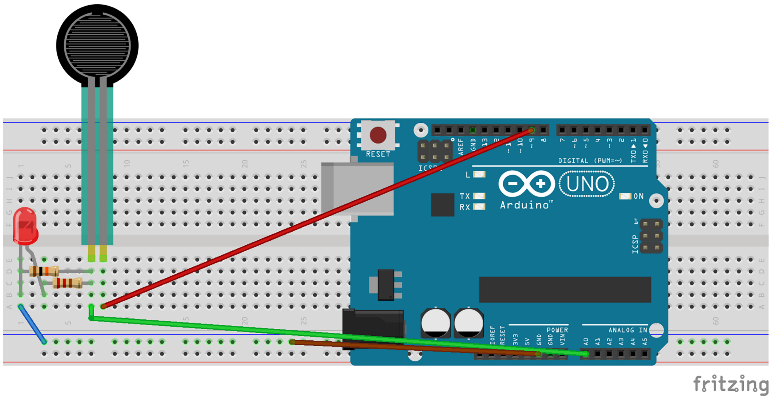


FIGURE 4.2: Preliminary study. Circuit diagram used as the blueprint for constructing the device shown in Figure 4.3.

FIGURE 4.3: Preliminary study. Exterior: cardboard box with Rugby stress ball (left). Interior: Arduino MKR1000, FSR sensor, and LED (right), built according to the circuit diagram in Figure 4.2.

The findings aim to identify these challenges and provide insights into how they hinder the ability of non-experts to independently construct IoT solutions. By understanding these obstacles, the research seeks to uncover critical areas where support mechanisms and resources are most needed to enable successful IoT project development for end users.

4.2 | Study Method

This section outlines the specific techniques and procedures used to conduct our research. It details the stages of the study, the participants involved, and the data collection methods employed. By providing an overview of the study method, we aim to outline the steps taken to gather insights into how non-experts can develop bespoke IoT projects, thereby ensuring the clarity and reliability of our findings.

4.2.1 | Preliminary Study

The preliminary study aimed to assess commercial off-the-shelf technology for the main study, verifying the basic data flow and key components as starting points. This involved identifying, comparing, estimating, selecting, purchasing, and assembling parts into a proof-of-concept pilot project. Given our hypothesis that creating a bespoke IoT solution was too complex for a non-technical user, the preliminary study acted as a safety net, ensuring that the selected technology could support the main study's development. P1 conducted this study part-time in his office over four months, from August to November 2017.

Considering the Mood Squeezer [77], we identified the following components and data flow for the project: a) Sensors that react to changes in the environment (e.g., the pressure applied to a stress ball); b) A Wi-Fi-capable microcontroller that converts the sensor readings into digital data and transmits it to a cloud-based database; c) A cloud-based database that stores and processes the generated data; d) An actuator that transforms the digital data into something perceivable by the senses (e.g., coloured lights).

These components and data flow were compared with the diagram presented in Figure 2.1, where we considered three distinctive areas: *Sensing*, *Processing*, and *Actuating*. We examined these areas to produce the prototype of an IoT project, aiming to identify suitable sensors, a compatible microcontroller, a cloud-based database, and open-access Application Programming Interfaces (APIs) to facilitate seamless data flow between the system components.

- **Sensing**
  A pressure sensor (Force Sensitive Resistor, FSR) inside a red stress ball from University merchandise was used for data input instead of buttons. An Arduino MKR1000 with Wi-Fi and battery power was chosen for its versatility and housed in a cardboard box (Figure 4.3). The Arduino was programmed in C++ using the Arduino IDE, similar to the Mood Squeezer [77].

- **Processing**
  Firebase was chosen for cloud processing and storage due to its clear documentation, simple API, examples, and user-friendly interface. After setup, the capturing module was programmed in JavaScript to upload data. An LED on the box indicated when data was sent.

- **Actuating**
  During the preliminary study, visualisation was simulated through a basic web interface displaying records appearing in the database.

By the end of the preliminary study, we successfully tested force-sensitive resistors (FSR), the Arduino MKR1000 microcontroller with Wi-Fi connectivity, and the Firebase cloud database with an understandable API. The detailed circuit configuration is illustrated in Figure 4.2. These components served as the building blocks for the second stage of the study, presented in the following subsection.

The *Preliminary Study* helped us understand the complexities of assembling an IoT project, verifying that building the project was indeed possible, thus ensuring that we were not sending the participant in the *Main Study* to a dead end.

4.2.2 | Main Study

The second stage of our case study involved developing a bespoke project based on the proof of concept conducted by a researcher (P1), as presented previously. We named this project *The Mood Squeeze Converter*, as it converts physical energy—pressing a stress ball—into light by turning on and changing the colour and intensity of an LED strip.

In this phase, the IoT application to develop included all the parts identified in the data flow diagram shown in Chapter 1: *Sensing* user interactions, *Processing* data locally and in the cloud, and *Actuating* through LED lights (Figure 2.1).

The objectives of the *Main Study* were as follows:

- To understand the time frame required to achieve an acceptable functional state for the project.
- To comprehend the different phases a user must go through to complete it.
- To identify obstacles that could hinder the project's progress.

The user was tasked with developing a functional prototype using a set of requirements and a list of technological components within a reasonable time frame. To facilitate this process, we provided a consolidated group of components, including hardware, software, the outcome of the first stage, and a collection of educational material.

These were the resources that Participant P2 could use to start developing a custom-made solution:

a) a list of hardware to be used (sensors and microcontroller),
b) an Arduino Starter Kit (an Arduino Uno board, a selection of the most common electronic components, and a book with 15 simple projects to get familiar with the creation of circuits and the use of the Arduino developing environment (IDE)),
c) a circuit diagram of the setup used in the preliminary study (Figure 4.2),
d) a stipulated database (Firebase), including a link to the Firebase Web Codelab tutorial [1].

4.2.3 | Interviews

After completing the main study, we interviewed the participant (P2) using a semi-structured format and following the guiding questions presented in Appendix C. To provide a contrasting perspective, we subsequently interviewed a Computer Science postgraduate student who had been working on a bespoke IoT project. At the outset of their respective projects, P2 and the

[1] https://codelabs.developers.google.com/codelabs/firebase-web/index.html

postgraduate student self-identified on a five-step scale, ranging from non-expert to expert, with P2 identifying as a non-expert (level 1) and the postgraduate student as intermediate (level 3).

The interviews were conducted with the following objectives:

1. To assess the developers' self-perception of expertise in IoT and determine if it evolved throughout the project.
   We hypothesise that engaging in a project enhances expertise, even in incremental ways.

2. To identify the obstacles non-expert developers encountered while creating a custom-made IoT project.
   Addressing these challenges is essential for answering the broader thesis question: *Is IoT accessible to everyone, or is it a fallacy?*

3. To evaluate whether developers can recognise how they identified and resolved issues.
   Understanding their problem-solving processes will inform the development of better support systems for non-experts creating custom IoT solutions.

4. To clarify gaps in the observational data collected during the development process.
   This objective specifically applies to the non-expert developer in our case study. The interviews aimed to address areas that may not have been fully captured through other methods of data collection, providing a deeper understanding of the development process.

4.2.4 | Participants

The preliminary study, conducted in our lab by the principal researcher, aimed to build a proof of concept and verify the project's feasibility. The main study then focused on understanding the support needed by end-user developers, particularly those with limited expertise. Finally, a set of interviews was conducted with P2, another postgraduate student in Computer Science, and the supervisor of the latter.

For the main study, participant P2 was selected based on availability and alignment with the study's aims, rather than through a predefined sampling criterion. As a student in the Master of Computer Graphic Design programme, P2 had basic programming experience but no prior exposure to electronics or physical computing. This level of expertise was inferred through ongoing supervisory interaction and direct observation, rather than formal testing. P2 was not chosen for technical skill, but for their demonstrated resilience in tackling unfamiliar challenges. They were provided with the appropriate technology and basic instructions as outlined in the materials list.

Given the size and context of the research environment, recruitment options were necessarily pragmatic. Throughout the research, P2's technical capability and confidence demonstrably evolved. Although no formal pre- or post-test was employed, this development was evident in the digital diary, which captured increasing levels of autonomy, experimentation, and reflection. This naturalistic mode of assessment aligns with the exploratory, design-led nature of the study and its broader aim of understanding how non-expert developers navigate and appropriate IoT technologies over time.

This extended, single-participant study served as a mapping-the-territory exercise—a foundational inquiry into how a non-expert might navigate the creation of an Internet-connected device from first principles. At its outset, no fixed timeline was established; instead, the project unfolded pragmatically as P2 encountered and worked through a series of real-world challenges related to IoT prototyping. The study took place within the context of establishing an IoT laboratory in the School of Computing and Mathematical Sciences. At the time, P2 was completing a Master's degree, unrelated to the present research topic. Although external factors supported sustained involvement, motivation fluctuated, particularly during periods of technical failure and conceptual uncertainty, as reflected in the participant's diary. The experiential insights drawn from this long-form engagement directly shaped the subsequent studies (Figure 1.1). The need to support ideation (Chapter 5), concretisation through making (Chapter 6), and troubleshooting in the face of failure (Chapter 7) each emerged from lessons gained during P2's iterative journey.

4.2.5 | Data Collection

The study design incorporated three distinct phases of data collection: the preliminary study, the main study, and the post-study interviews. These phases were methodologically linked through a sequential, interpretative process rather than through formal triangulation. Observations from the preliminary study were primarily instrumental, informing the setup of the main study and ensuring technical feasibility. The main study generated a substantial corpus of qualitative material in the form of a digital diary, comprising extensive notes, embedded media, and code. This diary was analysed through close observation and supported by visualisation techniques that allowed patterns and recurring challenges to surface, consistent with a design-led interpretative approach. Interviews were subsequently used to clarify and contextualise findings from the diary, offering reflective insight into processes already observed. While video recordings were made to support memory recall, they were not formally coded or systematically analysed. Instead, they functioned as an auxiliary aid during interpretation. This approach reflects a design logic in which each phase of data collection feeds into and refines the next, supporting a situated understanding of end-user development practices.

Data Collection Procedures: Preliminary Study

The preliminary study was documented through digital notes and a video recording of a demonstration showcasing the functioning prototype (see Figure 4.3). The code developed during this phase was saved and provided to Participant P2 as a reference and starting point for the main study.

Data Collection Procedures: Main Study

The data collection during the Main Study, spanning from October 2017 to September 2018, involved Participant P2's part-time engagement in the second stage of the case study. Data was gathered through two primary methods: the development of the *Mood Squeeze Converter* prototype and the maintenance of an extensive digital diary. The *Mood Squeeze Converter* was constructed as three Internet-connected input devices (Figure 4.5) using an Arduino MKR1000 microcontroller, with data stored in a cloud database and LED lights controlled via an API. P2

Watched three of the Shiffman Firebase videos.

Wisdom earned:
Anode = Positive = power = long leg
Cathode = negative = ground = short leg

Tue 24 Oct 2017

Arduino
Began to familiarise with Arduino
Completed the first three example projects in the Arduino projects book

First major hurdle - identifying what resistor to use as the images in the book are not the same as the resistors in the kit.
Also, 220 Ohm and 10K Ohm resistors look very similar.

Wisdom earned: Always check that you have initialized the correct pin in the setup and are pointed at the right pin in the loop.

Fri 20 Oct 2017 16:04

Arduino
Getting familiar with Arduino.
Do the first tutorials / projects with the Arduino Kit.

Firebase
Talk about doing the Firebase Web Codelab tutorial
https://codelabs.developers.google.com/codelabs/firebase-web/#0

Coding
Shifman's Programming A to Z (look at the Firebase example)
http://shiffman.net/a2z/

FIGURE 4.4: The first page of the digital diary, initiated on 20 October 2017.

FIGURE 4.5: One of the final input devices.

documented the process thoroughly in a digital diary, capturing successes and failures through 231 pages of notes, code, and multimedia elements. These records provide a comprehensive dataset, analysed in the following sections.

P2 approached the development task as an open-ended design challenge rather than a rigidly structured exercise. While a predefined set of components, tools, and instructional resources was provided—including a reference circuit diagram and access to the code from the preliminary study—P2 was explicitly encouraged to experiment and adapt the materials as needed. This pragmatic framing enabled autonomy, with the task treated as a situated problem to be explored rather than executed. Accordingly, P2 modified and extended both the hardware and software components of the system, notably commissioning a custom-printed circuit board (PCB) from an overseas supplier to enhance system robustness.

The study further revealed emergent behaviours indicative of the social dimensions of end-user development. P2 sought assistance from individuals beyond the formal research context, informally assembling a mentorship network. This reflects known dynamics in peer-supported and distributed learning environments. However, such interactions also introduced risk: in one instance, misguided advice from a presumed expert resulted in the damage of a microcontroller. These occurrences highlight both the affordances and vulnerabilities of socially mediated learning among non-expert developers navigating bespoke IoT projects.

Data Collection Procedures: Interviews

The interviews were conducted as semi-structured sessions, following the guiding questions presented in Appendix C: Semi-structured Interviews. Two interviews were conducted with P2, along with one interview with the postgraduate Computer Science student who had been working on a bespoke IoT project. All interviews were recorded on video for further analysis.

The interview data were analysed using an inductive approach, with themes emerging through repeated engagement with the transcripts. Analysis was undertaken following a design approach rather than with a formal coding framework more typical of HCI or social science. The transcripts were manually reviewed to identify patterns and points of thematic convergence. This approach

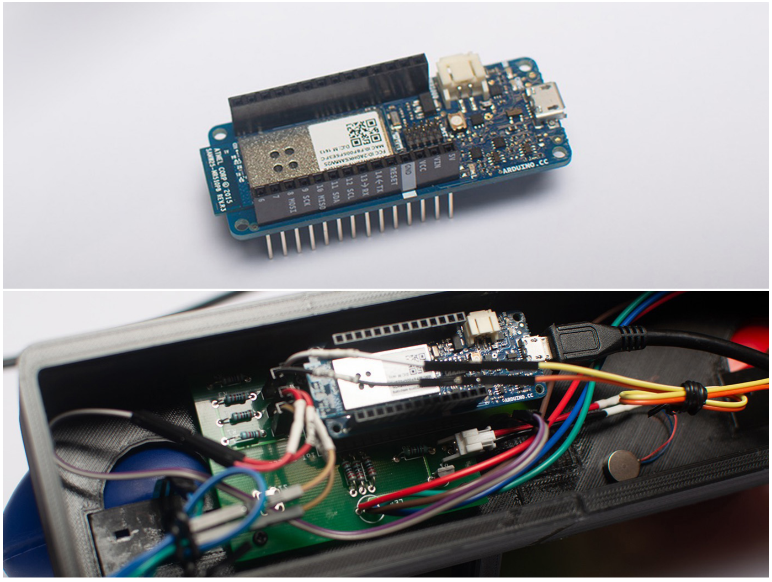


FIGURE 4.6: Arduino MKR1000 microcontroller used in the case study (top). Internal view showing the microcontroller connected to a custom PCB and a vibration motor (bottom).

FIGURE 4.7: Actuator: LIFX Z LED strip lights.

aligns with the principles of inductive thematic analysis, adapted here through a designer's mode of inquiry that privileges close reading and emergent interpretation. To support sensemaking, visualisation techniques such as word clouds were employed, facilitating pattern recognition through spatial and graphical representations.

Although no formal inter-rater reliability checks or software-assisted coding tools were applied, interim findings were discussed within supervisory meetings. These sessions served as an informal triangulation mechanism, supporting the refinement of interpretations. We recognise that this analytic process constitutes an exploratory entry point into qualitative HCI research, reflecting a transition from a design-oriented methodology towards more structured interpretative practice.

4.2.6 | Ethical Considerations

The study received ethical approval from the School of Computing and Mathematical Sciences (SMCS) Ethics Committee at the University of Waikato. The approval letter is included in Appendix A. All participants were informed of the study's aims and procedures and provided their consent before participating. This applies to all forms of data collection, including interviews, digital diaries, and video recordings. No specific ethical concerns were identified regarding participant vulnerability or data sensitivity.

4.3 | Analysis

This section is dedicated to interpreting the data gathered during our study. It is structured into three subsections: Analysis of the Diary, Analysis of Phases, and Analysis of the Interviews. Through these subsections, we thoroughly evaluate the participants' activities, stages, and reflections, aiming to uncover critical insights into developing bespoke IoT projects by non-experts.

4.3.1 | Analysis of the Diary

We present an analysis of participant P2's digital diary to understand the time required to achieve a functional project state, the phases a user must navigate, and obstacles that may impede progress. Returning to our initial hypothesis, we aim to confirm that despite accessible, affordable components, assembling them into a functional system remains too complex for a non-technical user.

Preparing the Data Corpus

Before analysing the diary, we pre-processed the content to create a clean text version by removing photographs, screenshots, diagrams, 3D images, URLs, and code. The text was organised by project days (PDs) and stored in a MySQL database for easy querying. We then extracted a text file with one line per project day for analysis using NLP tools like spaCy[2] and Textacy[3].

spaCy is an open-source Python library for advanced NLP, supporting tokenization, tagging, parsing, and named entity recognition (NER). In contrast, **Textacy**, built on spaCy, handles tasks like stemming and lemmatisation. Using these tools, we created a Doc for each project day, with metadata indicating the project day (PD). These Docs were then compiled into a *Textacy corpus*, allowing for efficient processing and tagging with specific terms (e.g., *'squeezy ball'*) using spaCy's rule-based matching[4].

Analysing the Corpus

Transforming the diary into a corpus revealed that over 343 days (from 20 October 2017 to 28 September 2018), P2 recorded activities on 157 non-consecutive days. Of these, 148 are considered

[2] https://spacy.io

[3] https://chartbeat-labs.github.io/textacy/

[4] https://spacy.io/usage/rule-based-matching#entityruler

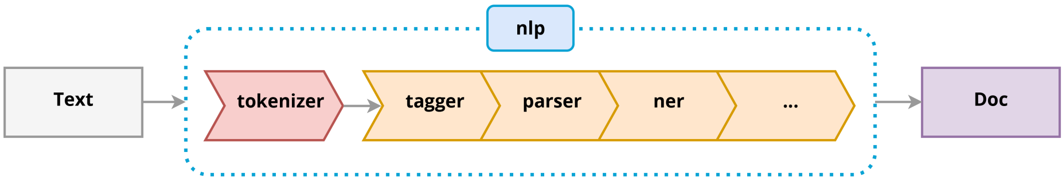


FIGURE 4.8: spaCy's pipeline: from text to Doc.
Reproduced from *spaCy*, https://spacy.io.

FIGURE 4.9: Process used to transform the digital diary created in Google Docs to a spaCy corpus.

'project days' with relevant diary entries. The concept of **148 Project Days** is central to the corpus analysis, which employed various visualisation techniques, including WordClouds, as detailed in the following subsections.

Visualising the corpus

After preparing the corpus, we generated visual representations of the text for various project days. These visualisations, including verbosity levels and word clouds for individual days and categories, revealed patterns that helped identify different project phases related to specific topics.

Basic Statistics and Verbosity

The initial analysis of the corpus with Textacy revealed 148 Docs (one per PD), 3,165 sentences, and 46,048 tokens. Using Textacy's TextStats, we calculated the number of sentences and words per Doc. We visualised this data in a bar chart to show verbosity levels throughout the project (Figure 4.10).

Word Clouds per Day and by Category

To visualise term usage across different days and project areas, we used *word clouds*, where term frequency determines the size or colour of words, revealing patterns. Textacy was employed to generate word clouds for each Project Day (Figure 4.11), including creating bag-of-terms, lemmatisation, removing stop words, using unigrams, bigrams, and trigrams.

While reviewing these word clouds, we identified the need to refine our analysis by categorising the content to focus on specific project areas and more accurately identify relevant keywords. Some concepts, like *Arduino project*, were clear, while others, like *complete project*, were ambiguous. Categorisation allowed for more precise insights.

Initially, we categorised sections based on the data flow diagram (Figure 2.1), such as **interacting** (*sensors*, *microcontroller*), **processing** (*database*), and **visualising** (*lights*). Further analysis added categories like *box* (enclosure), *electronics* (internal components), *hardware* (assembly elements), and broader categories like *software*, *protocols*, and *learning* (Table 4.1 and Table 4.2).

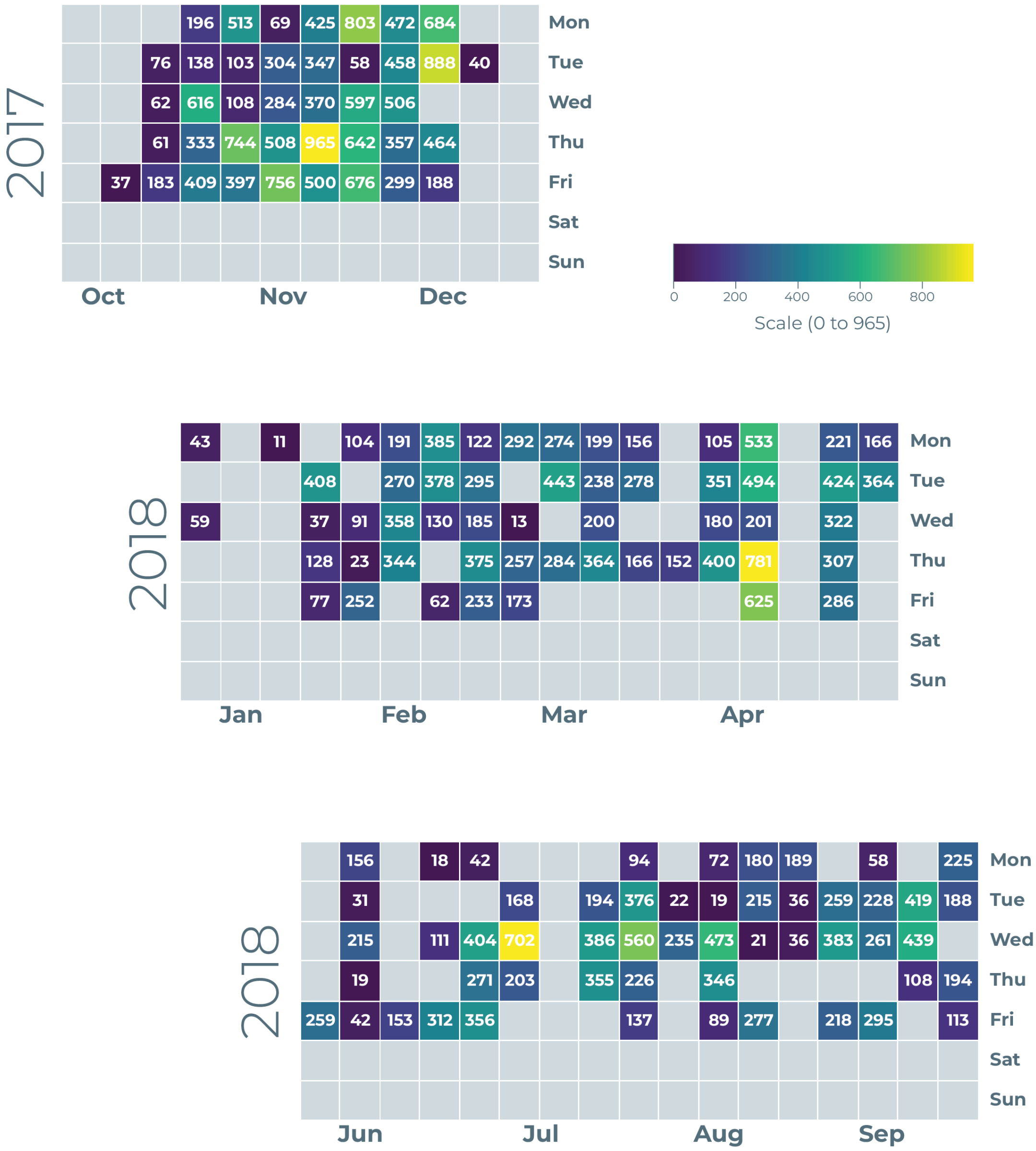


FIGURE 4.10: Word count per PD, illustrating verbosity levels throughout the project.

To ensure consistency, we used lemmas—the base forms of words—as keywords. A Python script compared stems to tokens in the corpus, generating these lemmas. Each Doc in the corpus was evaluated against the keyword lists, resulting in category-specific word clouds for the 344 project days. The output included word clouds, blank images, “x” for non-project days, or circles for weekends (Figure 4.12), which were then stitched together to create strips of word clouds for each category (Figure 4.13).

This approach, akin to zooming in with a microscope, revealed that breaking down content by categories was crucial for a deeper understanding of the project phases.

TABLE 4.1: Table of categories.

Data Flow	Observation	Generic
sensors	box	software
microcontroller	electronics	protocols
database	hardware	learning
lights		

TABLE 4.2: Keywords for the Box category.

Box		
3D	epoxy	squeeze
ball	epoxying	squeezing
box	glue	squeezy
epoxy	lid	

Problems and Solutions

While the word cloud visualisation provides a topical overview over time, some sentences were misclassified as *false positives*. For instance, on PD 37, the keyword *"box"* was wrongly categorised under the physical project box, though it referred to a computer dialogue box:

> *The new network device (or whatever that dialog* ***box*** *says) should pop up.*
>
> *Whenever the MKR1000 is plugged, the computer produces a dialog* ***box*** *that says there is a new network available.*

Similar misclassifications appeared in the *lights* and *electronics* categories, where keywords like *light*, *red*, *green*, and *thing* were wrongly linked to unrelated elements.

To address this, we added a filtering process to the script generating the word clouds, allowing it to skip sentences identified as false positives or belonging to different categories. An interactive Python script further aids in identifying and excluding misclassified sentences, storing them in a category-specific dictionary for accurate analysis.

Verifying Analysis with Entity Visualisation

The analysis of the word cloud strips for different categories was supported by spaCy's entity visualiser, which highlights named entities and labels in the text. This process allowed us to verify whether the content in the WordClouds for the 148 project days accurately matched the corresponding categories or included false positives. Through iterative evaluation and comparison with the entity visualisations, we were able to confirm the consistency of the categorisation and identify any discrepancies.

Summary

The diary analysis methodology involved multiple iterative processes, recognising the need to slice the material for effective analysis. While the use of NLP tools, parsing techniques, and visualisation instruments proved useful and could be further developed for analysing large unstructured texts in other contexts, expanding this methodology is beyond the scope of this thesis.

4.3.2 | Analysis of Phases

This section presents the analysis results outlined in Subsection 4.3.1. Our case study aims to understand the time frame needed to bring the project to an acceptable state, identify the phases a user must navigate, and pinpoint obstacles that could hinder progress. The results enhance our

FIGURE 4.11: Wordclouds per day for PD 1 to 5

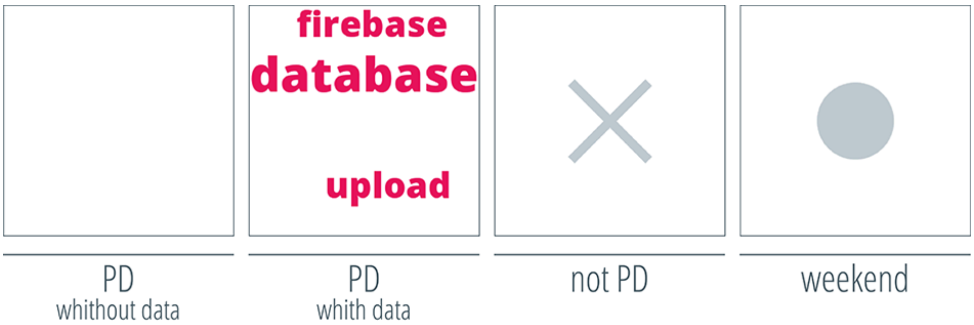


FIGURE 4.12: Potential word cloud outputs include a PD without data, a PD with data, a day without a diary entry, or a weekend.

FIGURE 4.13: Database category segment of the strip, showing Project Days 81–83.

understanding of the process a non-expert user undergoes to complete a simple IoT project and provide valuable insights to verify our hypothesis: the complications of assembling an IoT project are too significant for a non-expert to accomplish.

Through the diary analysis, we identified fifteen distinct phases, which occurred over three separate periods due to external circumstances like university holidays. Diagrams and visualisations, such as those in Figure 4.14, Figure 4.15, and Figure 4.16, were employed to gain a deeper understanding of the complex processes followed by our participant.

The following list outlines the fifteen identified phases, divided into three periods based on time:

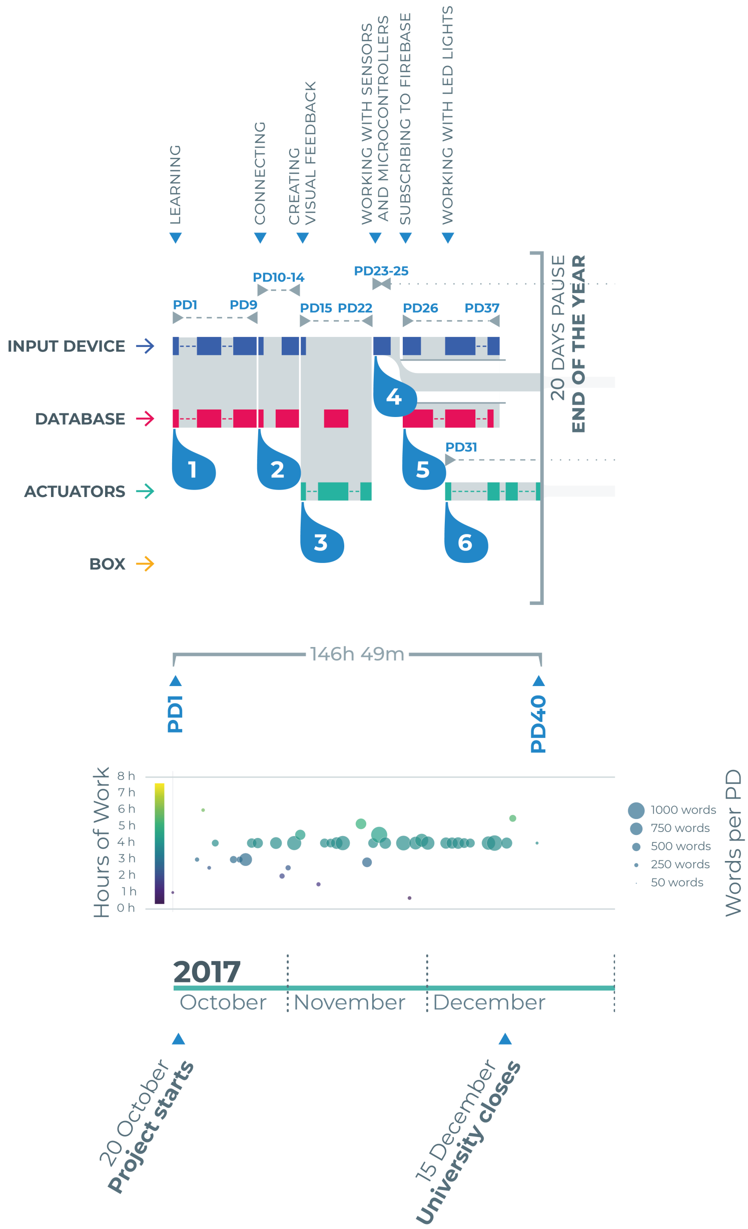


FIGURE 4.14: Initial Period: PD1 to PD40.

PD1–PD40, PD41–PD95, and PD96–PD148. However, when viewed conceptually, these phases can be grouped into three segments: the initial steps (Phases 1–3), the expansion and creation of components (Phases 4–9), and the final stages (Phases 10–15). While these conceptual segments do not strictly align with the temporal divisions, key events within the time periods often trigger the transition between segments.

– **Initial Period:** PD1 to PD40 (Figure 4.14)

Phase 1 – Learning

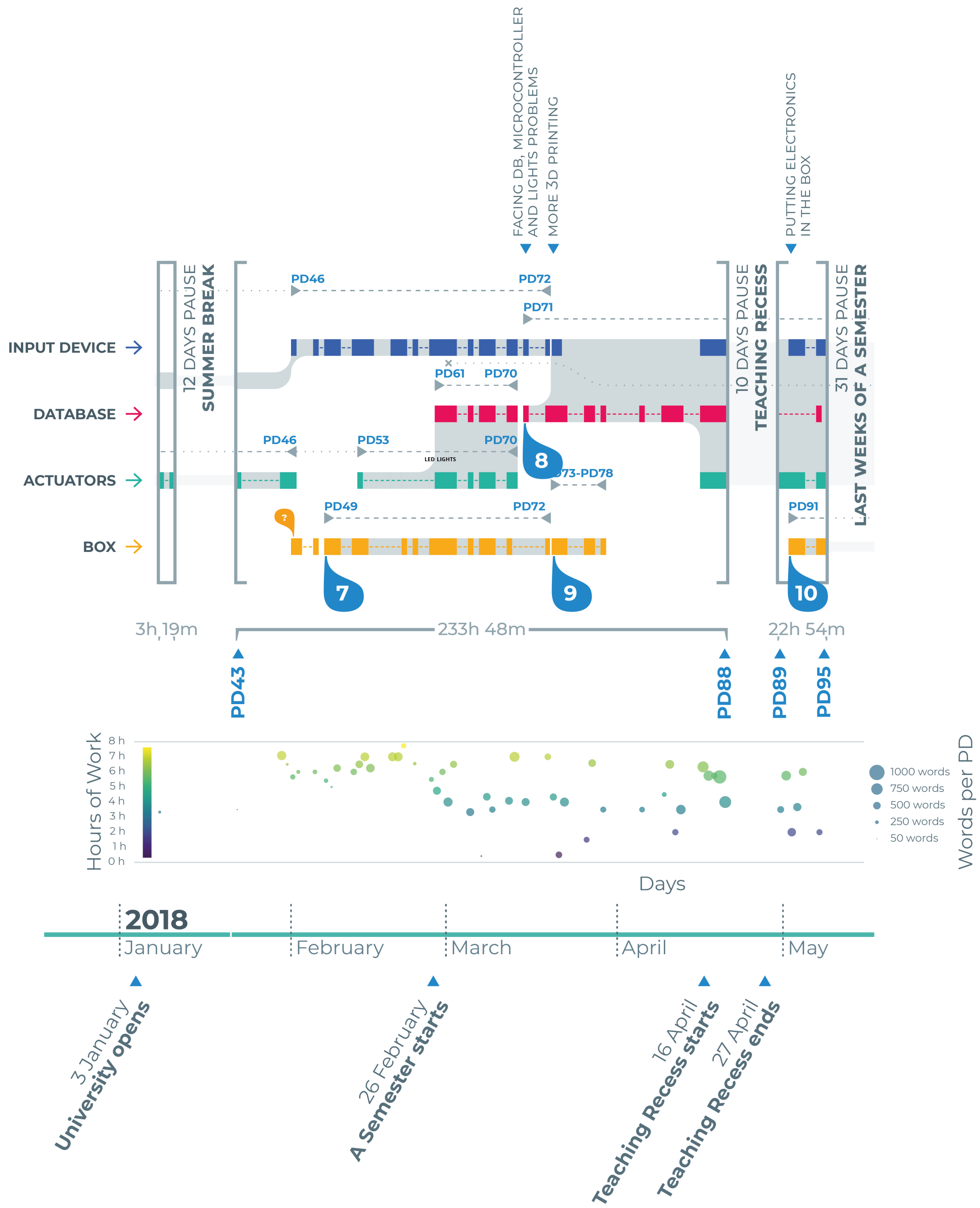


FIGURE 4.15: Middle Period: PD41 to PD95.

Phase 2 – Connecting

Phase 3 – Creating Visual Feedback

Phase 4 – Working with Sensors and Microcontrollers

Phase 5 – Subscribing to Firebase

Phase 6 – Working with LED Lights

– **Middle Period:** PD41 to PD95 (Figure 4.15)

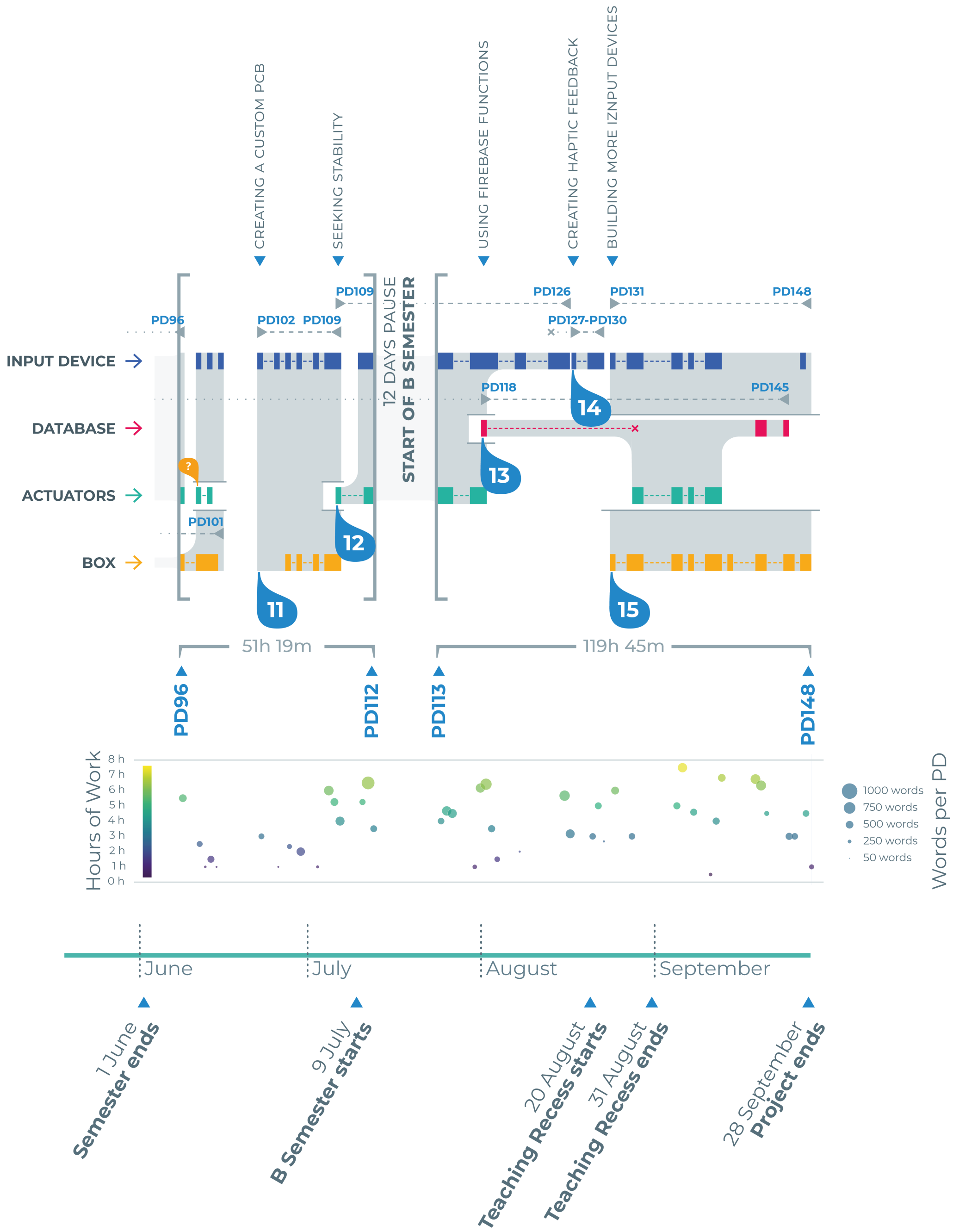


FIGURE 4.16: Final Period: PD96 to PD148.

Phase 7 – 3D Printing

Phase 8 – Encountering DB, Microcontrollers, and Light Problems

Phase 9 – More 3D Printing

Phase 10 – Putting Electronics into the Box

- **Final Period:** PD96 to PD148 (Figure 4.16)

Phase 11 – Creating a Custom PCB

Phase 12 – Seeking Stability

Phase 13 – Using Firebase Functions

Phase 14 – Creating Haptic Feedback

Phase 15 – Building More Input Boxes

Phase 1 – Learning

In the project's early days, our participant focused on learning how the initial components worked, starting with the input device and the database. They received an Arduino kit with fifteen tutorials and a link to the first Firebase tutorial (Friendly Chat). While the tutorials were not directly related to P2's project, they provided a solid foundation for the work ahead.

The diary shows that P2 spent nine days (PD1 to PD9) on these materials, totalling 29 hours and 30 minutes (an average of 3 hours and 17 minutes per day). By PD9, the participant noted that they had completed the Arduino and Firebase tutorials. Diary excerpts illustrating these points are provided in Appendix B: Phase 1 – Learning.

Phase 2 – Connecting

The second phase focused on connecting the microcontroller to the cloud database. This phase began when the participant received an Arduino MKR1000, a board with Wi-Fi connectivity, recommended for those with minimal networking experience. Although P2 completed Arduino and Firebase tutorials in the previous phase, none covered integrating these technologies, leading to the first set of challenges.

According to the diary, this phase spanned from PD10 to PD14, with P2 working 16 hours and 30 minutes (an average of 3 hours and 18 minutes daily). Excerpts demonstrating these insights can be found in Appendix B: Phase 2 – Connecting.

Phase 3 – Creating Visual Feedback

Between PD15 and PD22, our participant focused on creating visual feedback as an initial approach to actuating (Figure 2.1), marking the project's third phase. P2 began by blinking the onboard LED of the Arduino Uno and then created a website to visualise database records in real-time. However, using simple actuators was later abandoned in favour of more visually striking options like smart LED light strips.

During this phase (PD15-PD22), P2 worked for 30 hours (an average of 3 hours and 45 minutes daily). Supporting diary excerpts are documented in Appendix B: Phase 3 – Creating Visual Feedback.

Phase 4 – Working with Sensors and Microcontrollers

On PD23, our participant began connecting the force-sensitive resistors (FSRs) to the Arduino MKR1000, marking the start of a new project phase that spanned the first 40 project days and extended into the second part after the summer break. Unlike the earlier phases, this and subsequent ones often overlapped and progressed in parallel.

In the first part of this phase (PD23 to PD25), P2 worked for 12 hours and 30 minutes (an average of 4 hours and 10 minutes per day). The second part (PD46 to PD72) ran concurrently with phases

6 and 7, during which P2 worked for 148 hours and 48 minutes (an average of 5 hours and 31 minutes per day). Estimating that about a third of this time was dedicated to this phase, we conclude that 49 hours and 36 minutes were spent, totalling 62 hours and 6 minutes for this phase. Diary notes exemplifying this process are detailed in Appendix B: Phase 4 – Working with Sensors and Microcontrollers.

Phase 5 – Subscribing to Firebase

On PD26, our participant began working on subscribing to Firebase, aiming to build a system that could react when new rows were recorded in the database. From PD26 to PD37, P2 spent 44 hours and 50 minutes (an average of 3 hours and 44 minutes per day) working on integrating the input device with the cloud database. However, during this time, they were also working on sensors and microcontrollers (Phase 4) and starting on LED lights (Phase 6), so not all those hours were dedicated to the subscription task.

P2 experimented with various approaches, including using a different microcontroller, the SparkFun ESP8266 Thing, due to its available Firebase library. The goal was to create a system where a microcontroller could subscribe to Firebase and send messages to the actuator API. Although P2 eventually succeeded, the solution was unstable and required a cumbersome process of unplugging and plugging the device back in, which was far from ideal. Illustrative diary entries are available in Appendix B: Phase 5 – Subscribing to Firebase.

Phase 6 – Working with LED Lights

In the final days before the summer break, P2 began working on setting up the LED lights, which would serve as the project's actuating part. This development occurred in PD31 to PD46 and PD53 to PD70.

Between PD31 and PD46, P2 spent 67 hours and 35 minutes (an average of 4 hours and 13 minutes per day), with some of that time also dedicated to finishing Phase 5 – Subscribing to Firebase, which ended on PD37. From PD53 to PD70, P2 worked for 97 hours and 27 minutes, while also continuing work on Phase 4 – Working with Sensors and Microcontrollers) and Phase 7 – 3D Printing. As P2 became more immersed in the project, the phases began to overlap rather than proceed sequentially, making it difficult to determine the exact time dedicated to each phase, unlike Phases 1, 2, and 3. Examples from the diary are detailed in Appendix B: Phase 6 – Working with Led Lights.

Phase 7 – 3D Printing

In early February 2018, P2 began creating a box to enclose the electronics (microcontroller and sensors) and support the rugby stress ball. After initial discussions, it was decided that the box would be sized to fit three balls side by side and would be 3D printed using the School's on-site printer.

This phase spanned from PD49 to PD72, during which P2 worked for 131 hours and 8 minutes (an average of 5 hours and 28 minutes per day). However, P2 was also working on Phase 4 – Working with Sensors and Microcontrollers and Phase 6 – Working with LED Lights simultaneously, so it is unclear how much time was specifically devoted to this phase. Based on the average daily work time and diary notes, most hours were likely spent on the box's creation and printing.

The 3D printing process was iterative, involving trial and error, and took several days to produce a usable object due to the slow printing speed. Although 3D printing is known as *"rapid prototyping"*, in this case, the process was slower than traditional methods for small-scale production. The diary excerpts shedding light on this are in Appendix B: Phase 7 – 3D Printing.

Phase 8 – Facing DB, Microcontrollers and Light Problems

At this point in the development, P2 had worked on all project parts. However, as the process was not streamlined or plug-and-play, various problems arose in nearly all areas: the database, the microcontroller, and the LEDs. Between PD71 and PD96, the focus shifted to solving these issues.

From PD71 onward, many issues involved the ESP8266 microcontroller (SparkFun The Thing), used to subscribe to Firebase (Phase 5 – Subscribing to Firebase) and connect to the LIFX API. Although this approach was eventually abandoned, P2 spent significant time troubleshooting it.

Over these 24 project days, P2 worked 107 hours and 20 minutes (an average of 4 hours and 8 minutes per day). This period included breaks of 10 days (Teaching Recess) and 31 days (end of Semester A), which slowed progress. During this time, P2 focused almost entirely on problem-solving, likely dedicating most of the 107 hours to this phase.

The diary reflects P2's frustration during this phase, as isolating and resolving problems was challenging. However, their determination paid off, and by PD96, the system was working, with P2 noting, *"the beast lurches back into life."* This is supported by diary entries found in Appendix B: Phase 8 – Facing DB, Microcontrollers and Light Problems.

Phase 9 – More 3D Printing

Beforehand, in Phase 7 – 3D Printing, P2 developed a box to enclose all the electronics and attach the stress rugby balls. That phase ended on PD72 when the first box parts were printed. Phase 9 began on PD73 and lasted until PD78, focusing on printing additional boxes.

During these days, P2 worked for 27 and 25 minutes (an average of 3 hours and 24 minutes daily). While the provider was printing the boxes, P2 simultaneously worked on solving issues with microcontrollers, the database, and LED lights (Phase 8 – Facing DB, Microcontrollers and Light Problems). Since an external provider handled the 3D printing, P2 spent limited time on this phase, mainly overseeing the process and checking progress. A few days later, the provider printed all the boxes. Documentation from the diary is included in Appendix B: Phase 9 – More 3D Printing.

Phase 10 – Putting Electronics in the Box

Once all the components were functioning together—though some issues remained—the priority shifted to placing the circuit in the enclosure. This task, constituting Phase 10, spanned from PD91 to PD101.

During this period, P2 worked for 25 hours and 10 minutes (an average of 2 hours and 17 minutes per day). Simultaneously, they completed Phase 8 – Facing DB, Microcontrollers and Light

Problems, so not all time was dedicated to enclosing the electronics. The lower work rate, below the project's average of nearly 4 hours per day (3h 54m), suggests that P2 felt fatigued, particularly after dealing with challenging issues in Phase 8.

As P2 assembled the circuit using a breadboard, they realised the need to reorganise it. This led to the idea of creating a custom printed circuit board (PCB) after attempting point-to-point wiring to fit the components in the enclosure. Diary references that illustrate this can be found in Appendix B: Phase 10 – Putting Electronics in the Box.

Phase 11 – Creating a Custom PCB

After days of struggling to fit the circuit inside the enclosure, P2 enlisted help from S1, an electrical engineering undergraduate they knew from a role-playing game. Together, they developed the idea of creating a custom-printed circuit board. Phase 11 covers PD102 to PD109.

During these 8 project days, P2 worked for 24 hours and 35 minutes (an average of 3 hours and 4 minutes per day), primarily focusing on this phase. The overlap with the next phase began on PD109. The average working time per day was slightly below the project's overall average.

This phase also introduced two key helpers: S1, the electrical engineering student, and S2, a Research Programmer at the University. Their assistance was invaluable in getting the system to a functioning state. Evidence from the diary is presented in Appendix B: Phase 11 – Creating a Custom PCB.

Phase 12 – Seeking Stability

At this project stage, everything was in place, but the system was still unstable, and P2 focused on refining details to achieve consistent functionality.

This phase spanned PD109 to PD126, during which P2 worked 61 hours and 50 minutes (an average of 3 hours and 26 minutes per day). Although there was a slight overlap with the next phase, most of the time during these project days was dedicated to this phase. The diary records these instances in Appendix B: Phase 12 – Seeking Stability.

Phase 13 – Using Firebase Functions

At this stage of development, using a second microcontroller (SparkFun The Thing) to control the actuators (LIFX LED strip) had become more of a problem than a solution. As seen in Phase 12 – Seeking Stability, this approach was unstable and difficult to implement. P2 realised that returning to an idea discussed months earlier—using Firebase functions to control the lights—might be a better solution.

The concept of using Firebase functions first appeared in the diary on PD63 but was not implemented until much later. Distinct points mark this phase in time: PD63 (the first discussion), PD118 (reviving the idea), and the period from PD133 to PD145, when it was finally implemented.

Excluding the initial markers, this phase spans PD133 to PD145. During these 12 project days, P2 worked 49 hours (an average of 3 hours and 46 minutes daily). Work on the final phase—completing all required input devices—also took place during this time, making it unclear how the work was distributed. Supporting details from the diary are located in Appendix B: Phase 13 –

Using Firebase Functions.

Phase 14 – Creating Haptic Feedback

As the project neared completion, P2 focused on incorporating haptic feedback into the input device. The feedback, created with a vibration motor connected to the circuit, indicates when a message is sent to the database, ensuring the user knows their interaction was successful.

The vibration motor was connected using solderless JST PA connectors. To attach the motor to the JST PA female connector, P2 needed to use *crimps*, a type of solderless electrical connection. This process required some learning, as crimps are attached using specific tools, which can be costly. P2 found alternative solutions and got it working.

This phase spanned 8 working days, from PD123 to PD130, during which P2 worked for 19 hours and 30 minutes (an average of 2 hours and 26 minutes per day). Relevant diary passages are included in Appendix B: Phase 14 – Creating Haptic Feedback.

Phase 15 – Building More Input Devices

The project's final phase focused on assembling the second and third input devices. Although all components had been tested with one functioning device, repeating the process was not error-free. As a non-expert, P2 still encountered mistakes, as noted in the diary.

This phase spanned 18 project days, during which P2 worked for 66 hours and 30 minutes (an average of 3 hours and 42 minutes per day). Some of this time overlapped with Phase 13 – Using Firebase Functions. According to the diary, P2 assembled the second input device in one day (PD131, 6 hours of work). While the third device likely took a similar amount of time, additional issues extended the process.

The project concluded on PD148, with all input devices assembled and functioning. Diary notes illustrate this point in Appendix B: Phase 15 – Building More Input Devices.

Phases Summary

This section outlined the phases involved in developing an IoT project in our case study, addressing the chapter's second goal: *to understand the phases a non-expert developer must navigate to complete an IoT project*.

In the early phases, project stages were clearly defined with distinct start and end points (e.g., Phases 1-3). However, as the project evolved and became more complex, P2 frequently engaged in multiple areas simultaneously, making phase boundaries less distinct. Some topics were also revisited after an initial pause (e.g., Phase 13).

As mentioned earlier, the project phases can be viewed through two different lenses: a time-based division into three periods—PD1–PD40 (Figure 4.14), PD41–PD95 (Figure 4.15), and PD96–PD148 (Figure 4.16)—and a concept-based grouping. The latter organises the phases into three segments: the initial steps (Phases 1–3), the expansion and creation of components (Phases 4–9), and the final stages (Phases 10–15). While these conceptual segments do not strictly align with the temporal periods, key events within those periods often signal the start or end of each segment.

TABLE 4.3: Obstacles hindering IoT project progress.

Batteries	Having other commitments	Problems and testing
Bluetooth	JavaScript	Pub/Sub
C/C++	Lack of documentation	Sensor location
Confusing electronics	Lack of guidance	Sensors and movement
Data loss	Lack of methodology	Signal noise
Design failure / bad planning	Lack of self-confidence	The lab vs the real world
Errors that cannot be solved	Lack of vocabulary	Too new technology
Errors with libraries	Microcontrollers	Understanding JSON
Errors without explanation	MicroPython / Zenphyr	Understanding NoSQL
Finding the root of a problem	Moving from A to B thoughtfully	Unexpected behaviours
From the breadboard to a real circuit	Not having someone to ask	Wi-Fi
Frustration	Physical aspects	

The next section expands on the diary analysis with interviews of IoT developers.

4.3.3 | Analysis of the Interviews

Based on interviews with participant P2 and a Computer Science postgraduate student, we identified several obstacles developers faced at different stages of their projects (Table 4.3). Drawing on D. Norman's concepts of declarative knowledge ("knowledge of") and procedural knowledge ("knowledge how") [138], we developed a mind map categorising these obstacles into two groups. The mind map in Figure 4.17 illustrates the tasks an IoT developer must carry out (procedural) and the challenges outside their control (declarative). Recognising these challenges is essential for understanding the complexity of IoT projects, particularly for non-experts who may feel discouraged from beginning such projects.

By organising the hindrances, we identified seven subgroups within Procedural Knowledge. These subgroups are examined through the lens of threshold skills, as outlined by Thomas et al. [170]. While Thomas et al. use *'students'* in the context of computing education, in our study, *'individuals'* or *'people'* are more appropriate. Unlike threshold concepts, which focus on theoretical understanding, threshold skills emphasise practical application, making this framework particularly relevant to analysing non-expert IoT development.

Threshold skills possess the following characteristics:

- **Are Transformative**
  Mastering a threshold skill transforms not only what individuals can do but also how they perceive their capabilities.

- **Are Integrative**
  Once a threshold skill is acquired, individuals recognise its broader applicability across different tasks.

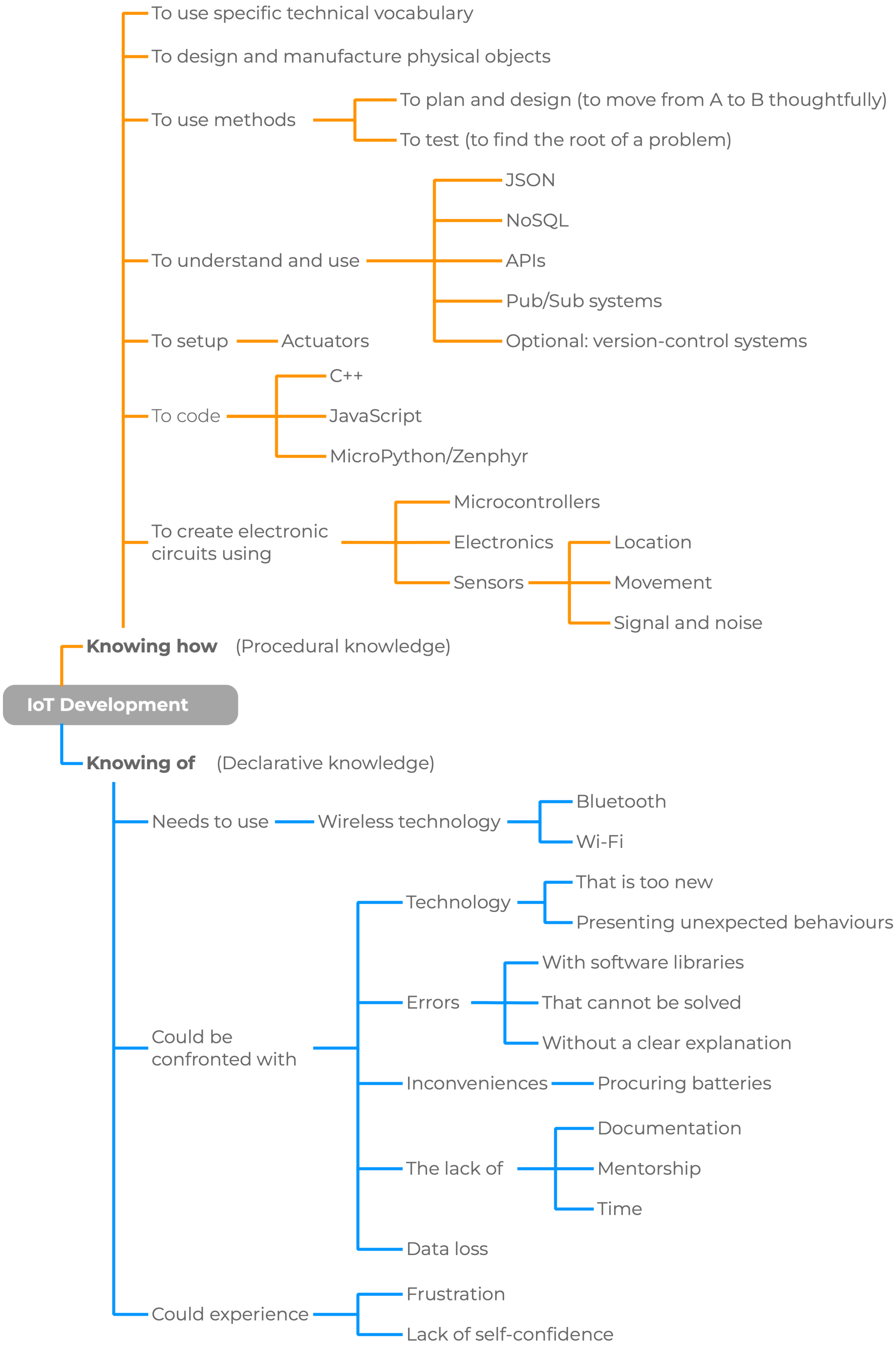


FIGURE 4.17: IoT Development: Distinguishing between Procedural Knowledge (how to do) and Declarative Knowledge (what to know).

– **Are Troublesome**
These skills can be complex, demanding, and time-consuming to learn and maintain.

– **Are Semi-irreversible**
Unlike threshold concepts, threshold skills degrade over time if not practised regularly.

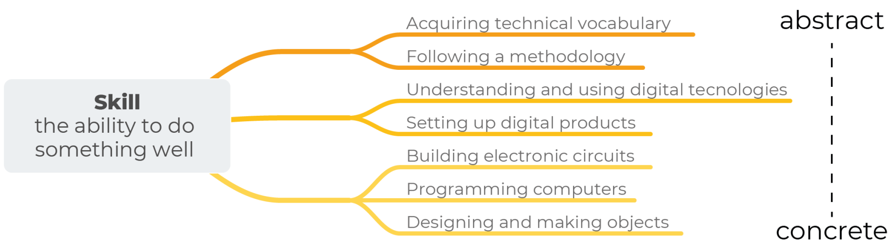


FIGURE 4.18: Skills. From abstract to concrete.

- **Must be practised**
  A skill, by definition, is developed and maintained through practice and repeated exercise.

The psychology literature on skill acquisition covers several theories (e.g., ACT-R, SOAR) that explain the development of skills. Commonalities across these theories suggest that: 1. Practice leads to substantial performance improvements early on, with diminishing returns as practice continues, typically following a power law; and 2. Task transfer is affected by the degree of similarity between tasks [162].

Our analysis of the interviews identified several core activities that are essential for IoT development. These activities reflect the practical challenges developers must address and the key skills they need to acquire. The list below outlines these activities, with specific emphasis on understanding and applying various digital technologies:

- Acquiring technical vocabulary
- Following a methodology
- Understanding and using digital technologies, including:
  - Data interchange formats (JSON)
  - Mechanisms for storage and retrieval of data (noSQL)
  - Application programming interfaces (APIs)
  - Asynchronous messaging systems (Pub/Sub)
  - Distributed version control systems (Git)
- Setting up digital products
- Building electronic circuits
- Programming computers
- Designing and making objects

These skills can be organised along a spectrum from abstract to concrete, as shown in Figure 4.18.

In the following subsections, we evaluate each skill by discussing its characteristics and typical learning environments and mapping it in relation to threshold skills theory. This evaluation helps determine whether these skills qualify as threshold skills, which could offer insights into how they should be approached or the challenges they present to end-user developers.

Acquiring Technical Vocabulary

Mastering technical vocabulary is pivotal for IoT developers, allowing them to engage effectively with peers and experts. This skill is **transformative**, as it empowers developers to tackle complex problems by participating in key discussions. It is also **integrative**, enabling the application of newly acquired terms across various tasks, such as programming and troubleshooting [176]. The constant evolution of IoT technology makes this process **troublesome** for non-experts, who must continuously adapt to emerging terminology.

Technical vocabulary is also **semi-irreversible**, requiring ongoing **practice** to prevent the loss of knowledge. As such, it qualifies as a threshold skill, fundamentally altering how developers approach their work in IoT projects.

Following a Methodology

Adopting a structured methodology is crucial for IoT development, influencing how developers plan, design, and troubleshoot their projects. Interviewees acknowledged that a more organised approach could have greatly benefited their work. Methodologies from various fields, such as design and engineering, bring distinct challenges when integrated into multidisciplinary projects [80].

This practice is **transformative**, as it enhances developers' ability to manage complexity, and **integrative**, with methodologies like design thinking applicable across multiple contexts [132]. Furthermore, mastering methodologies can be **troublesome**, requiring significant time and effort. It is **semi-irreversible**, with skills fading over time without regular use. Given these traits, following a methodology qualifies as a threshold skill in IoT development.

Understanding and Using Digital Technologies

Interviews highlight that IoT developers need a solid understanding of various digital technologies, including JSON, noSQL, APIs, asynchronous messaging systems, and distributed version control (Git). These skills are vital for managing the complexity of IoT projects, and many have the potential to meet the characteristics of **threshold skills**. The following sections evaluate each technology in relation to these criteria.

Data Interchange Formats (JSON)

JSON, widely used for data exchange in IoT, is favoured for its lightweight structure and ease of use compared to XML [182]. It enables communication between different systems, making it an essential skill for IoT developers. Mastering JSON is **transformative**, allowing developers to expand their capabilities, and **integrative**, with applications across multiple fields. Learning JSON can be **troublesome** due to its complexity, particularly for non-experts, as seen in the need for comprehensive resources like *Introduction to JavaScript object notation: A to-the-point guide to JSON* [13]. Regular **practice** is essential to maintain proficiency, making JSON a clear threshold skill in IoT development.

Mechanisms for Storage and Retrieval of Data (noSQL)

NoSQL databases, crucial for handling large volumes of data in IoT, offer scalability and adaptability [145]. Mastering noSQL is **transformative**, enabling developers to manage complex data-driven tasks, and it is **integrative**, with applications ranging from data storage to real-time analytics. Due to the depth of knowledge required, learning noSQL can be **troublesome**, particularly for non-experts. As with many technical skills, it is **semi-irreversible** without regular **practice**, confirming its place as a threshold skill in IoT development.

Application Programming Interfaces (APIs)

APIs facilitate data exchange across mobile, cloud, and IoT environments [95]. While APIs are important for specific tasks, they are not universally **transformative**, with their impact often limited to particular services. They are **integrative** in specialised contexts but remain challenging to master, especially when dealing with complex parameters. API knowledge is **semi-irreversible**, requiring regular **practice** to maintain proficiency. Given these limitations, APIs do not fully qualify as a threshold skill.

Asynchronous Messaging Systems (Pub/Sub)

Pub/Sub systems are vital for efficient communication in IoT, enabling asynchronous messaging without direct interaction [96]. This skill is **transformative**, altering how developers approach system management, and **integrative**, with applications across various domains. The complexity of pub/sub systems makes them **troublesome**, and the skill requires regular **practice** to avoid degradation. While pub/sub is highly valuable, it does not meet all the criteria to be considered a full threshold skill.

Distributed Version Control Systems (Git)

Distributed version control systems like Git are indispensable for managing code repositories and tracking changes [27]. Git is **transformative**, enabling developers to collaborate effectively, and it is **integrative**, with applications extending beyond software development. The challenges of mastering concepts like branching and conflict resolution make Git **troublesome**, and regular **practice** is necessary to maintain proficiency. Despite its importance, Git does not fully meet the criteria of a threshold skill in IoT development.

Setting Up Digital Products

Setting up digital products, such as smart lights, is essential but does not qualify as a **threshold skill**. Although it is important in specific contexts, this task is not **transformative** or **integrative** and is largely limited to the immediate task at hand. While setup can be **troublesome**, requiring time and effort, it lacks the broader impact seen in other threshold skills.

Building Electronic Circuits

Building electronic circuits is fundamental in IoT development, providing the pathways for current flow in devices [114]. It is **transformative**, expanding developers' understanding of technological possibilities, and somewhat **integrative**, though its applications are specific to certain tasks. The process can be **troublesome**, requiring hands-on experience and regular **practice** to maintain proficiency. As such, building electronic circuits aligns with many characteristics of threshold skills.

Table 4.4: Evaluation of Threshold Skills in IoT Development.

	Transformative	Integrative	Troublesome	Semi-irreversible	Must be practised
Acquiring technical vocabulary	✓	✓	✓	✓	✓
Following a methodology	✓	✓	✓	✓	✓
Understanding and using digital technologies					
Data interchange formats	✓	✓	✓	✓	✓
Storage and retrieval of data	✓	✓	✓	✓	✓
Application Programming Interfaces	✗	✗	✓	✓	✓
Asynchronous messaging systems	✓	✓	✓	+/-	✓
Distributed version control systems	+/-	✓	+/-	✓	✓
Setting up digital products	✗	✗	✓	✓	✓
Building electronic circuits	✓	✗	✓	✓	✓
Programming computers	✓	✓	✓	✓	✓
Designing and making objects	✓	✓	✓	✓	✓

Programming Computers

Programming, or coding, is a foundational skill that qualifies fully as a **threshold skill**. Coding is **transformative**, reshaping an individual's ability to solve problems, and highly **integrative**, with applications across various fields [51]. Learning to code can be **troublesome**, particularly due to its inherent complexity, and regular **practice** is required to prevent proficiency from fading. Programming computers is an essential threshold skill in the digital age [98].

Designing and Making Objects

Designing and making physical objects is integral to IoT projects, particularly for tasks like sensing and actuating. This skill is **transformative**, expanding what developers can achieve, and highly **integrative**, with applications across various domains [69]. The process is often **troublesome**, requiring significant time and hands-on experience, and must be maintained through regular **practice**. Given these traits, designing and making objects qualifies as a **threshold skill**.

Threshold Skills Summary

Our analysis of skills identified in interviews with IoT developers reveals that:

1. The majority (6 out of 11, or 54%) fully align with the characteristics of threshold skills.
2. The remaining (5 out of 11, or 45%) partially align but do not meet all the criteria.

A visual summary of this analysis is presented in Table 4.4.

This suggests that the challenges IoT developers face align with the threshold skills concept. As Thomas et al. [170] argue, while threshold concepts inform theoretical understanding, threshold skills are essential for guiding practical tasks. The process of successfully creating a bespoke

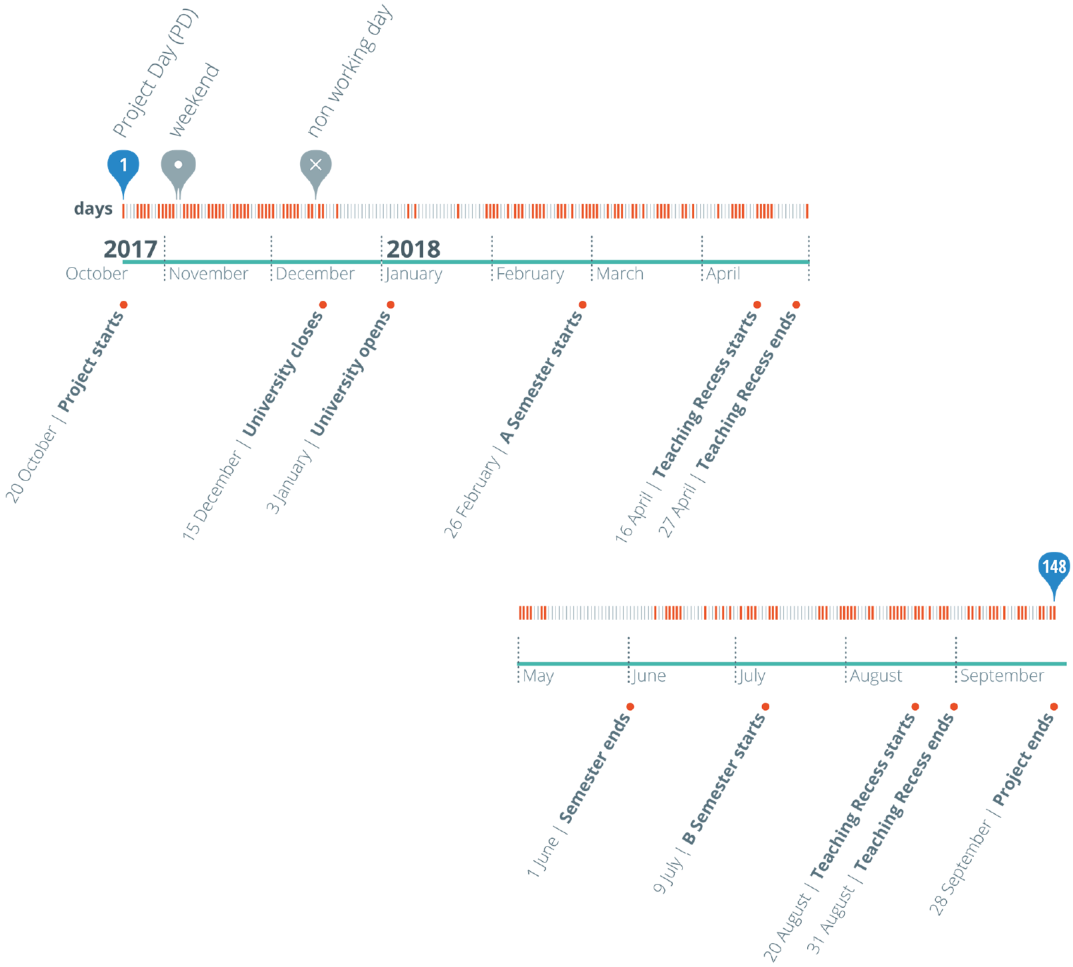


FIGURE 4.19: Project timeline over 344 days, highlighting 148 active Project Days (PD) based on diary evidence.

IoT project requires not only declarative knowledge—knowing *what* to do—but also procedural knowledge—knowing *how* to do it [138]. This dual requirement significantly increases the complexity of the task. Mastering both types of knowledge can be particularly challenging for end-user developers, as they must move beyond theoretical understanding to practical application. In some cases, this combination of declarative and procedural knowledge may render the process unachievable for non-experts, especially without continuous practice and support. This analysis shifts the focus from conceptual knowledge to the practical abilities required to execute IoT projects successfully, highlighting the intricate balance between knowing and doing.

4.4 | Discussion

In this chapter's introduction, we hypothesised that while it may be possible for a non-technical user to bring an IoT project to a functional state, the challenges could be overwhelming. To explore this, we presented a long-term case study conducted over several months in 2017 and 2018, where we observed an end-user developer working through creating an IoT project.

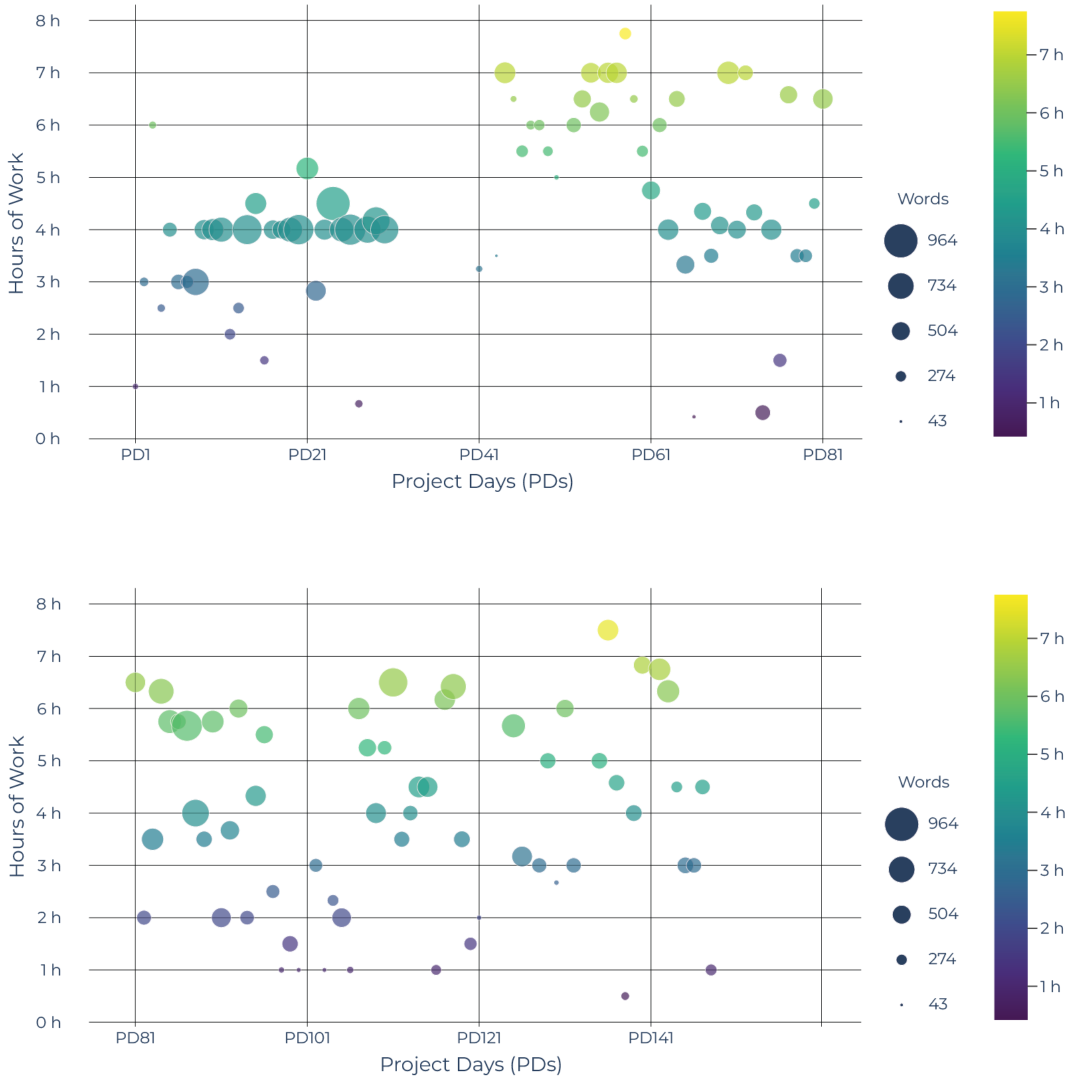


FIGURE 4.20: Hours of work and verbosity per Project Day (PD).

The case study focused on three primary goals:

1. Understanding the time required to achieve a functional bespoke IoT solution,
2. Comprehending the phases a non-expert developer must navigate to complete the project, and
3. Identifying obstacles that could hinder progress.

In the previous section, we analysed the diary in Subsection 4.3.1: Analysis of the Diary, examined the project phases in Subsection 4.3.2: Analysis of Phases, and explored insights from interviews with IoT developers in Subsection 4.3.3: Analysis of the Interviews. The following pages will connect these findings back to the initial goals, providing a deeper understanding of the complexities involved in IoT project development.

4.4.1 | Responding to Goal #1: Time Frame

The first observation from the diary focused on the time frame of P2's project. The project spanned 344 days, from 20 October 2017 to 28 September 2018 (Figure 4.19). Of these, 235 were working days, with 148 days (62% of working days) actively spent on the project. These were identified as Project Days (PDs), ranging from PD1 to PD148.

When P2 began in October 2017, the project's duration was unknown. We did not set a deadline to avoid undue pressure, allowing P2 to work without time constraints. Although a longer duration could have been an issue for the study, this was not the case, and P2 completed the project within the year.

P2 recorded daily work hours on timesheets, providing reliable evidence of time spent. Productivity varied, with some days more productive than others (Figure 4.20). The timesheets show that P2 worked 587 hours and 57 minutes, averaging 3 hours and 58 minutes per day over 148 non-consecutive days.

The goal was determining the time needed to bring a bespoke IoT solution to a functional state. The study found that a non-expert took 578 hours (577 hours and 58 minutes) to complete a fully functional IoT project, working an average of 4 hours per day (3 hours and 54 minutes). This suggests that a non-expert may abandon such a project without significant motivation before completion.

4.4.2 | Responding to Goal #2: Phases

To understand the phases a user must navigate to complete a bespoke IoT solution, we conducted an extensive analysis of the diary, supported by various visualisations that helped identify patterns and phases.

The primary visualisation (Figure 4.14, Figure 4.15 and Figure 4.16) is a timeline showing different work areas. Additional diagrams (Figures 4.21, 4.22, 4.23) expand on the IoT application's data flow (Figure 2.1), detailing tasks required in each project area.

Organised by categories, these visual representations provided insight into the project's phases. Initially, the analysis used detailed visualisations, as explained in Subsection 4.3.1, before simplifying it into colour-coded blocks representing Project Days (PDs). The x-axis shows time, while the y-axis divides the project into four categories: *the input device* (electronics, sensors, hardware, microcontroller), *the database*, *the actuators*, and *the box*. This approach was crucial in identifying fifteen distinct phases, as discussed in Subsection 4.3.2.

The study identified fifteen phases, each with varying complexity. Early phases were sequential, but later phases overlapped and became more complex as the project progressed. Phases began or ended for various reasons, such as meetings or the arrival of components. Notably, no common pattern emerged across the phases; some ended without clear reasons, while others concluded with specific achievements.

4.4.3 | Responding to Goal #3: Obstacles

The third goal of the case study was to identify potential obstacles that might impede project progress. We complemented the diary analysis with interviews to gain deeper insights, speaking to both P2 and an additional, more experienced IoT developer.

From the interviews, we identified several obstacles that could impede a project's progress. These obstacles align with Norman's concepts of declarative knowledge – *"knowledge of"* – and procedural knowledge – *"knowledge how"* [138]. Obstacles related to procedural knowledge correspond to *Threshold Skills* as defined by Thomas et al. [170].

We identified seven key threshold skills, some encompassing multiple sub-skills, that can act as significant obstacles in IoT development:

- Acquiring technical vocabulary
- Following a methodology
- Understanding and using digital technologies
  - Data interchange formats (JSON)
  - Mechanisms for storage and retrieval of data (noSQL)
  - Application programming interfaces (APIs)
  - Asynchronous messaging systems (Pub/Sub)
  - Distributed version control systems (Git)
- Setting up digital products
- Building electronic circuits
- Programming computers
- Designing and making objects

The study suggests that mastering these skills is critical for the successful execution of an IoT project. Since these skills are classified as *Threshold Skills*, the difficulty in acquiring them can impede progress and potentially prevent project completion.

Interestingly, the interviews revealed that, regardless of expertise level, developers encountered similar obstacles. However, the capacity to recognise issues, seek assistance, and the time required to overcome challenges varied significantly between developers.

These findings highlight the broad range of skills essential for IoT development, indicating that such projects may benefit from an interdisciplinary approach. The complexity involved could be exceptionally overwhelming for an end-user developer.

4.4.4 | Flow of an IoT Project

The case study indicates that an IoT project typically progresses through several key stages, with varying processes across the areas of sensing, computing, and actuating:

- Define

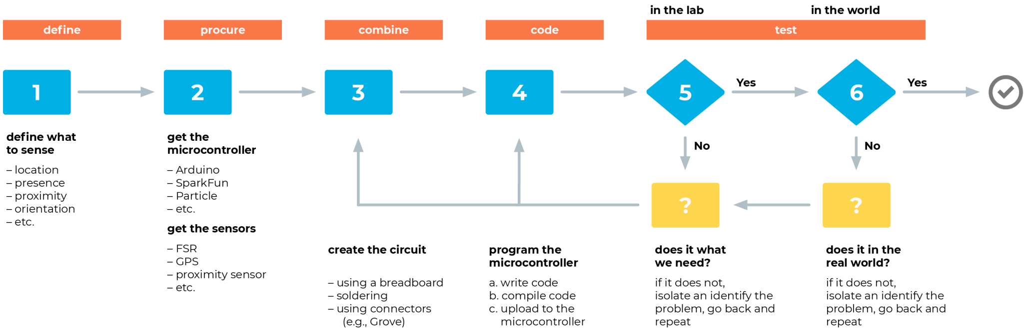


FIGURE 4.21: Process and flow for Sensing/Capturing Data.

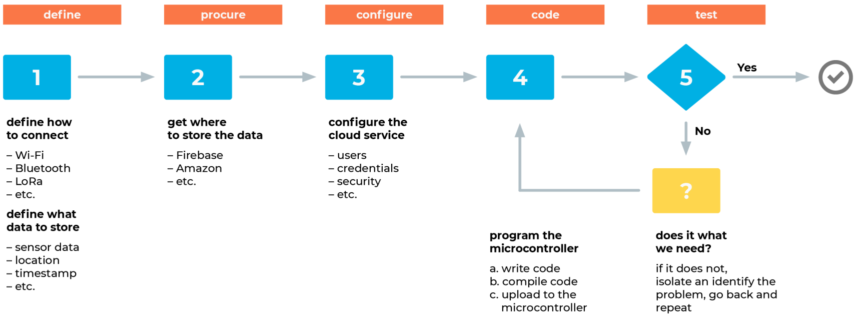


FIGURE 4.22: Process and flow for Processing/Storing in the cloud.

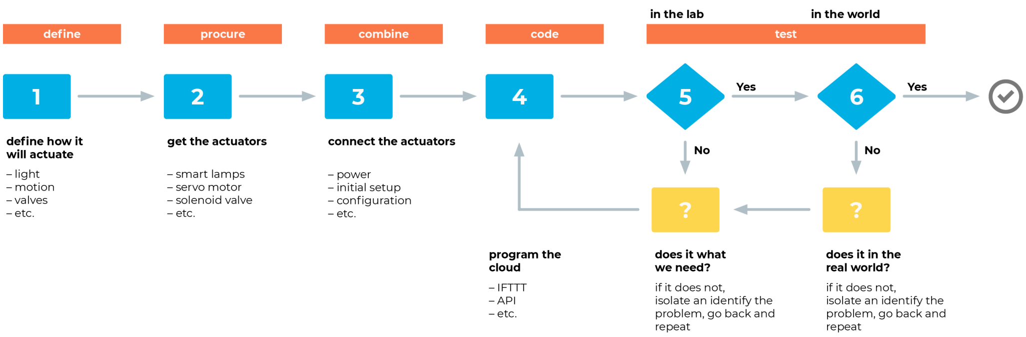


FIGURE 4.23: Process and flow for Actuating/Acting in the physical world.

- Procure
- Configure/Combine
- Test
    - In the lab
    - In the real world

The flow of processes across the three core components of an IoT project – Sensing, Processing, and Actuating – is illustrated in Figure 4.21, Figure 4.22, and Figure 4.23. These diagrams outline the tasks developers face, underscore the complexity at each stage, and showcase the range of options available for each phase of the project. While comprehensive, the diagrams are not exhaustive; additional steps or variations may be necessary depending on specific project needs.

4.4.5 | Discussion Remarks

The findings confirm the conceptual premise that the complexities involved in creating an IoT project are substantial. Our initial hypothesis suggested that the challenges of assembling a functional IoT project would be overwhelming for a non-expert. The case study offers strong evidence to support this hypothesis, indicating that the demands of such projects often exceed the capabilities of those without significant expertise.

4.5 | Summary

This section consolidates the key findings of the case study, addressing the objectives established at the outset of the chapter. By examining the time investment, phases of development, and obstacles encountered, the study offers a comprehensive understanding of the challenges faced by non-experts in completing a bespoke IoT project.

This case study set out to:

1. Understand the time frame required to bring a custom IoT project to an acceptable functional state,
2. Comprehend the different phases that a non-expert must navigate to complete their plan, and
3. Identify obstacles that could stop or delay the project's progress.

The findings revealed that completing a bespoke IoT solution could require up to 600 hours of work. Without strong motivation, non-experts are unlikely to complete such a project. The study also showed that developing an IoT solution involves multiple phases, each with varying levels of complexity. While the earlier stages are more accessible to non-experts, the later phases become significantly more challenging, often exacerbated by insufficient support and guidance.

Moreover, the research underscores that overcoming specific obstacles requires the acquisition of *Threshold Skills*, such as basic programming or electronic circuit assembly, with the complexity of these skills varying according to the non-expert's background. These findings strongly support the hypothesis that the challenges inherent in assembling a functional IoT project are too significant for a non-expert to navigate successfully.

This chapter's research lays the foundation for the next phases of our exploration into bespoke IoT project development by end-user developers. The insights gathered here serve as a critical reference for the following three chapters, which examine the practical aspects of concept refinement and ideation (Chapter 5), the support required for project assembly (Chapter 6), and the implementation of guided troubleshooting strategies (Chapter 7). Together, these chapters build on the findings presented here, offering a holistic approach to the challenges faced by non-expert developers in the IoT domain.

Chapter 5

Concept Refinement and Ideation

User Study on Conceptual Development

> *The only way to engineer the future tomorrow is to have lived in it yesterday.*
>
> — **Bill Buxton**
>
> *Sketching User Experiences* [25]

TOOLS THAT FACILITATE THE IDEATION AND DEFINITION PROCESSES in technology-driven projects could greatly benefit end-user developers, especially those with minimal expertise, to generate, refine and articulate project concepts. These resources should also provide technical guidance, enabling developers to make informed decisions about the feasibility and implementation of their ideas. The scenarios presented in Chapter 2 highlight the diverse contexts in which end-user developers could engage in IoT projects, emphasising the necessity of user-friendly, plug-and-play solutions to reduce initial confusion and ambiguity. Furthermore, Chapter 3 underscores the complexity of IoT ecosystems and the need for structured approaches provided by toolkits designed for ideation and prototyping. Toolkits such as *IoT Ideation Cards,* [46] *IoT Design Kit,* [47] and *Tiles IoT Toolkit,* [127] help generate, refine, and organise ideas, fostering divergent thinking and making the development process more accessible. However, none were created explicitly for end-user developers, highlighting the need for improved tools to bridge this gap.

Additionally, Chapter 4 illustrates the significant challenges end-user developers face due to initial ambiguity and the iterative nature of their projects. Offering aids that help organize and refine their ideas can improve their confidence and capacity to move their projects forward, reducing the obstacles caused by their limited technical expertise. This organisational clarity allows them to approach each project phase with greater confidence and precision systematically.

In this thesis, we focus on end-user developers with minimal expertise, distinguishing between novices with formal training and non-experts without specialised knowledge of digital technologies. Non-experts, lacking formal training, face more significant challenges in progressing their skills and require additional tools and support to clarify and organise their ideas. Their slower learning curve amplifies this need for assistance compared to those with formal training. Therefore, our discussion centres on supporting these end-user developers to effectively bring their ideas to fruition rather than exploring the entire journey to expertise.

Building on this identification, we hypothesise that end-user developers could greatly benefit from targeted support in the ideation phase of their projects. By providing structured guidance, developers can transform their initial ideas into actionable project concepts. This support, which includes technical guidance, not only aids in overcoming the initial barrier of conceptual vagueness but also sets a strong foundation for the subsequent stages of project development. Additionally, by integrating technical constraints and possibilities into the ideation process, these resources assist in understanding technical feasibility, thereby reducing the barriers posed by limited technical expertise.

Our study focuses on identifying effective support mechanisms during the ideation phase. By offering targeted assistance, we aim to empower end-user developers to enhance their creative processes and help them to navigate the technological complexities, which relates to our second research question:

> **RQ2** | In what ways can targeted support during the ideation phase improve end-user developers' ability to conceptualise unique technology-driven solutions?

This question investigates the potential of organised guidance to empower end-user developers. By examining how such support can help clarify concepts in the early phases of a project and bridge significant technical gaps for these developers, we aim to uncover strategies that enhance the initial stages of technology-driven project development.

In response to these challenges, we selected mobile applications as the focus for this study because they represent a common and accessible type of technology-driven project. Mobile applications involve design, prototyping, and technical considerations that challenge end-user developers, making them a practical choice for testing how conceptual tools can aid in refining and articulating project ideas. Leveraging these theoretical foundations, we created a conceptual card deck specifically designed to assist end-user developers in organising their ideas for mobile application development. To test our hypothesis, the card deck facilitates the process by enabling developers to mix, match, and arrange cards in various configurations, allowing a visual and tactile exploration of potential project ideas.

With the deck of conceptual cards in hand, we conducted a user study to evaluate the efficacy of this tool in supporting end-user developers during the initial stages of project creation. The study aimed to gather empirical data on how the cards influenced the developers' ability to refine and articulate project concepts, potentially bridging some of the existing gaps between end-user developers and the technology. Through this user study, we sought to validate our hypothesis that organised support could assist end-user developers in solidifying their ideas within a challenging technological environment.

This chapter is organised as follows: Section 5.1: Creating the Conceptual Deck delves into the design process of our bespoke deck of cards, integrating insights from historical contexts and pertinent literature. Section 5.2: Study Method outlines our user study's methodology, focusing on design and execution. Subsequent sections present the study's initial analysis (Section 5.3: Analysis), and critical evaluation of insights (Section 5.4: Discussion), culminating in Section 5.5: Summary, which summarises the main findings and implications.

5.1 | Creating the Conceptual Deck

This section describes the design process of a bespoke deck of cards created to conduct the user study presented in this chapter. It explains how we combined insights from past research with practical design elements to help end-user developers brainstorm mobile application ideas.

The following subsection, *"Cards in Design, HCI (Human-Computer Interaction), and SE (Software Engineering),"* explores the theoretical underpinnings and practical applications of card-based tools across these disciplines. This examination provides a foundation for understanding how cards have been historically used to facilitate complex processes like design thinking, user experience mapping, and software development. It sets the stage for how we tailored the conceptual card deck to address the unique needs of end-user developers in conceptualising mobile applications, highlighting the cards' role in bridging gaps between abstract ideas and tangible project outcomes. Through this exploration, we aim to contextualise our approach within the broader landscape of design and technology, underlining the significance of our user study in advancing the field.

5.1.1 | Cards in Design, HCI and SE

To consider methods and best practices related to creating artefacts to attain goals – bringing into the real world solutions that were initially only in their creators' imagination—we refer to processes followed by design, HCI, and software engineering. Some identifiable methods indicate what could be required to assist non-expert users in conceptualising a technology-driven solution.

The design process and how designers think have been extensively discussed in the literature. Books such as *Designerly ways of knowing* by Cross [37], *The design way* by Nelson and Stolterman [136], and *How designers think: The design process demystified* by Lawson [104] are dedicated to understanding and explaining these concepts. These authors explain that designers possess particular expertise in the "artificial world," which encompasses the human-made realm of artefacts [37]. They also highlight that at the heart of design as a human activity is the process of envisioning an ideal addition to the world and bringing that idea to life with form, structure, and shape [136]. Moreover, the commonalities between ideas presented by different authors lie in identifying distinct phases of the design process.

Building on these foundational ideas, the work presented in this chapter departs from concepts introduced by the Design Council, the UK's national strategic advisor for design, in 2005 [36] and the Design Thinking process developed at IDEO, a consultancy firm in the US, consolidated by Brown and Katz [20] and taught at the d.school at Stanford University.

The Design Council visualises the design process as a *"double diamond"* (Figure 5.1) and synthesises the process in four steps: Discover, Define, Develop and Deliver. The design thinking approach, as developed at IDEO, presents three core activities – inspiration, ideation, and implementation. The Stanford Design School (d.school) has expanded it into a five-stage process: Empathise, Define, Ideate, Prototype, and Test (Figure 5.2).

In the introductory chapter, RQ2 asks what supports non-expert users in creating their own IoT solutions. Considering the design models, part of the support should come in the divergent parts of the double diamond—Discover and Develop—and, correspondingly, in the Ideate phase of the Design Thinking model. This chapter's conceptual decks of cards exemplify ways to support non-expert users in the divergent stages of the process.

In the HCI literature, conceptual and theoretical aspects of gamification, including game mechanics, dynamics, and motivational strategies have been recognised as providing good results. Game design elements have been considered to enhance user experience and engagement in non-game contexts. Game mechanics such as points, badges, leader-boards, levels, and challenges can motivate and reward users, while game dynamics like competition and collaboration further enhance engagement. Feedback mechanisms and the importance of aesthetics have also been highlighted, emphasising the role of gamification in creating engaging and enjoyable interactions [56, 58, 59, 57]. An area of gamification is the use of bespoke decks of cards.

The use of conceptual cards is a well-established approach across different disciplines. In design, they have been used for defining problems, structuring the design process, making knowledge accessible, supporting creative endeavours, facilitating different design phases, co-designing user experiences, and engaging end users [50, 60, 61, 85, 115, 131, 186]. In HCI, the literature shows examples where cards have been used to integrate human values, address legal and regulatory issues, support metaphorical thinking, and enable data-driven reflections on the design process [45, 75, 113, 118]. In software engineering, CRC cards are used to teach object-oriented programming (OOP) [14, 86, 158].

In 2019, Roy and Warren [153] surveyed the landscape of card-based design tools, identifying 155 different decks. Notably, over 90% of these decks were created after 2000, reflecting a growing interest in and recognition of their value.

5.1.2 | Crafting the Conceptual Card Deck

This section explains the development process of the bespoke deck of cards utilised in our study, integrating historical context, theoretical foundations, and specific design considerations to enhance the ideation phase for end-user developers focusing on mobile application development. Translating abstract ideas into tangible project plans is particularly daunting for non-experts, who often grapple with a limited understanding of technical terminology and the breadth of available technologies. Such limitations frequently lead to an imprecise project definition, necessitating numerous iterations between conceptualisation and prototype development and thus complicating the journey to a viable product. In response to this challenge, we crafted a conceptual card deck

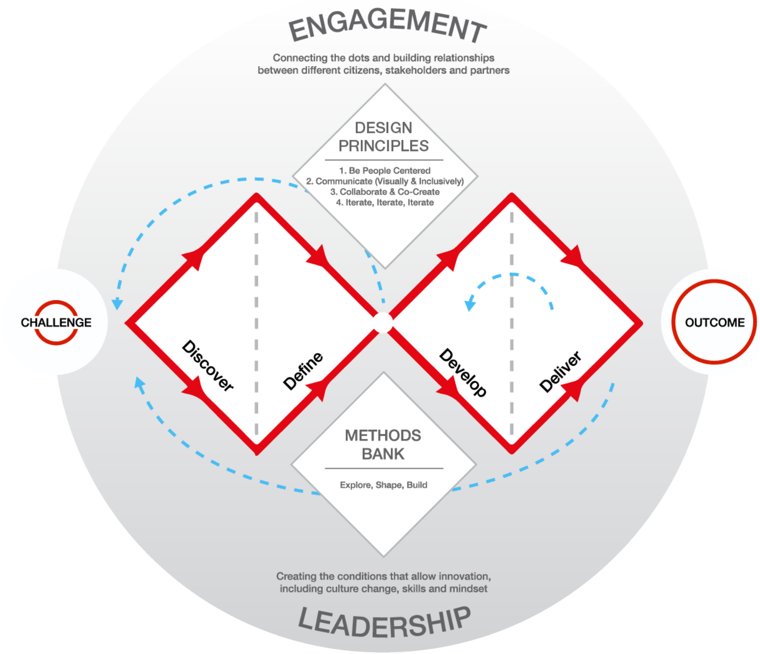


FIGURE 5.1: The Design Council's Framework for innovation. Reproduced from *the Design Council*, https://www.designcouncil.org.uk/our-resources/the-double-diamond/. Licensed under a CC BY 4.0 license.

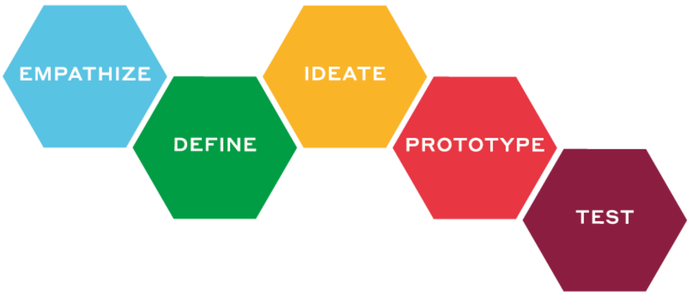


FIGURE 5.2: The five phases of the Design Thinking process. Reproduced from *Stanford d.school*, https://dschool.stanford.edu.

to ease the ideation process, providing a comprehensive overview of technological options and enabling users to explore various scenarios effortlessly. This tool is designed to help non-experts solidify their ideas, facilitating a smoother transition from vague concepts to detailed project outlines.

Historical References

Conceptual card decks have long been used to stimulate creativity, and several key predecessors have shaped the design of contemporary versions. These foundational decks are essential references for creating new conceptual card sets.

The *House of Cards* by Charles and Ray Eames, dating back to 1952, is one of the earliest examples. It features 54 cards, each showcasing an object, and includes slots for assembling the cards into three-dimensional structures, aiming to enhance creativity through play (Figure 5.3). The

FIGURE 5.3: House of Cards Collector's Edition.
Reproduced from *Eames Office*, https://www.eamesoffice.com/.

Eameses created this deck to encourage experimentation, imagination, and an appreciation for the beauty of everyday objects, fostering a playful yet meaningful engagement with design and structure.

Oblique Strategies, created by Eno and Schmidt in 1975, comprises 113 cards offering suggestions or remarks to overcome creative blockages. This deck has seen multiple reprints and adaptations, including its availability as an Amazon Alexa News Feed.

Method cards, introduced in 2003 by IDEO, is a deck of 51 cards to foster new design approaches and inspiration. Each card describes a specific method, such as *Scale Modelling*, and guides its application, further demonstrating the utility of card decks in creative processes.

The historical context and theoretical foundations significantly informed the inclusion of technical elements in the card design. This approach ensures the cards inspire creativity and guide users through practical technical decisions, bridging the gap between ideation and implementation. By incorporating technical elements into the cards, we ensure that end-user developers are equipped with the necessary knowledge to navigate the complexities of development, thus enhancing the overall utility and effectiveness of the toolkit.

Structuring the Cards

To implement these theoretical foundations, the envisioned card deck aimed to demystify the technological aspects of project development through an intuitive and engaging format. Drawing inspiration from traditional playing cards, the deck was organised into suits representing different components of data flow: input, output, sensors, and storage. Each card adopted a familiar size, such as the standard poker size (2.5"x 3.5") or the B8 size (62 x 88mm), with rounded corners to enhance usability within a game-like context. Aiming for accessibility, the deck ideally consisted of around 52 cards, mirroring the French-suited deck, which consists of four suits: hearts, diamonds, clubs, and spades. This approach helped to avoid overwhelming users with something unfamiliar.

The design utilised a simple yet effective combination of colour, typography, and icons to delineate each technology component, ensuring users could easily navigate and iterate through potential technological solutions.

This structured design not only makes the cards more accessible but also ensures that end-user developers can seamlessly integrate technical considerations into their creative processes, thus fully leveraging the historical and theoretical insights that informed the deck's creation.

Layout and Colours

In designing the deck, we made decisions following what is commonly used in the long tradition of playing cards. By adhering to traditional design principles, we ensured familiarity and ease of use for end-user developers. Aligning our design decisions with historical and traditional elements allowed us to create a deck that resonates with users, fostering creativity and engagement through its novel use without presenting something unfamiliar. Therefore, we designed rectangular cards that adhered to standard ratios (1:1.4 for poker cards or 1:1.419 for B8 cards) and can be oriented in either portrait or landscape formats. We chose the landscape format to optimise writing space. This design choice facilitates the inclusion of blank cards within the deck, allowing users to annotate them with additional actions (verbs) or objects (nouns) not already covered. The blank cards offer ample space for writing with a felt tip pen, like a Sharpie. These cards' overall design and layout maintain traditional playing card aesthetics, emphasising a balanced, centred arrangement of content both vertically and horizontally.

We believe the cards' colour scheme must be carefully chosen to ensure clear differentiation and sufficient contrast against black and white to accommodate colour-impaired users, such as those with colour blindness. While a specific colour palette is unnecessary, we selected coherent colours in brightness and saturation, with hues spaced as evenly as possible to facilitate easy recognition.

To aid in identifying suits, we employed a negative design approach, featuring coloured backgrounds with white elements. This enhances visual organisation and simplification. We adopted a flat design to create a distraction-free interface, utilising two-dimensional elements and solid colours without gradients. This approach promotes a calm and focused environment for users engaging with the deck, prioritising simplicity and usability. By ensuring the colours do not cause discomfort or distraction, we enable users to concentrate fully on their tasks.

Typography and Icons

We decided the typography for the cards to be unobtrusive and neutral. We opted to use Montserrat, a modern sans serif font designed by Argentine type designer Julieta Ulanovsky, as an homage to the neighbourhood of the same name in Buenos Aires. Montserrat is available in multiple weights (light, medium, bold, extra bold) that can be used to suit various contexts without drawing undue attention or distracting users. Consistency was key, with a single typeface used across all cards and related materials, ensuring a cohesive visual language. As with colour selection, the typographic presentation aimed to maintain a calm and subdued tone, supporting a focused, undisturbed user experience.

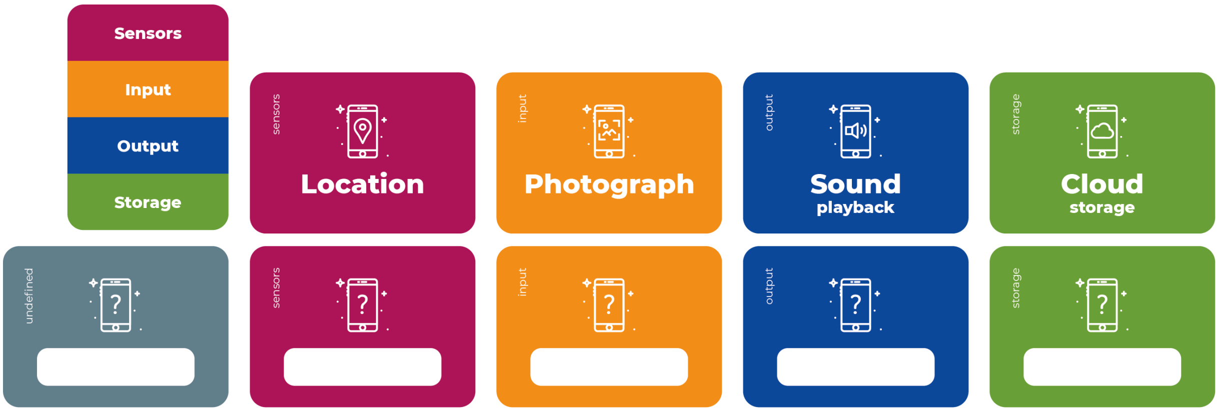


FIGURE 5.4: Sample of the conceptual deck of cards.

We designed each card to prioritise simplicity, with minimal information complemented by intuitive iconography that reinforces the card's message. Icons were part of a cohesive visual family to ensure uniformity. They directly represented the actions (verbs) or objects (nouns) clearly and straightforwardly, aligning with the typographic content without confusion. We employed familiar symbols to depict common concepts (e.g., a clock for time, a dropped pin for location) to prevent the need for new interpretations that might complicate the ideation process. All the icons were sourced from the *Dottie* collection by Xinh Studio, accessible through the Noun Project[1], to guarantee a unified and accessible design language across the deck. Our goal was to facilitate combining different cards to foster the idea-clarification process rather than relying on innovative or unfamiliar imagery on individual cards. Consistent with the chosen colours and typography, the visual language of the icons also contributed to a serene and focused setting, aiding users in organising their thoughts without being distracted by the tool itself.

Ideation Cards

In creating our deck, we based our decisions on having four suits or groups, as explained above. Having identified four key aspects of mobile device data use — Sensors, Input, Output, and Storage — and following an iterative process, we analysed the commonly provided features for each group, such as *Location*, *Motion*, and *Time* for Sensors. Each card incorporates technical elements to help non-experts understand the technical aspects of their projects, even when they lack the necessary technical vocabulary. We aimed to cover all conceivable aspects that end-user developers might need while allowing flexibility for unexpected items through an *Undefined* category. This approach ensures the deck is comprehensive and adaptable, meeting diverse needs while maintaining a familiar, organised format.

Organised in this way, the deck of cards helps in making technical decisions, often without the user even realising it. For instance, a user might recognise the need for a sensor (e.g., motion) or a particular type of input (e.g., voice recording using the device's microphone) simply because the card exists in the deck. Otherwise, some technical aspects are not part of the ideation process because end-user developers do not possess the required vocabulary. As the Austrian philosopher

[1]https://thenounproject.com/

TABLE 5.1: Suits and cards.

Sensors	Input	Output	Storage	Extra
Location	Photograph	List	Cloud Storage	Undefined
Motion	Rating	Map	Internal Storage	
Time	Sound Recording	Photograph		
	Text	Sound Playback		
	Video Recording	Text		
	Voice Recording	Video Playback		

Ludwig Wittgenstein stated, *"The limits of my language mean the limits of my world"* [185]. The cards, with their technical components, expand the world of non-experts by providing them with a ready-made technical vocabulary to use in their projects.

Finally, the deck we created, illustrated in Figure 5.4, consists of 48 cards, including an introductory card. Each category, Sensors, Input, Output, and Storage, is distinguished by colour. The *Sensors* category, marked by a maroon hue, encompasses *Location, Motion,* and *Time* cards. The *Input* category is coloured orange and features cards for *Photograph, Rating, Sound Recording, Text, Video Recording,* and *Voice Recording*. The *Output* category, depicted in blue, includes *List, Map, Photograph, Sound Playback, Text,* and *Video Playback* cards. Meanwhile, the *Storage* category is represented in green, containing *Cloud Storage* and *Internal Storage* cards (Table 5.1). The *Undefined* category is colour-coded in grey. These undefined cards are blank, allowing participants to use them as they see fit without being restricted to a predefined category.

5.2 | Study Method

This section details the methodology for the user study presented in this chapter. Subsection 5.2.1 provides a comprehensive overview of the study's structure and procedures, Subsection 5.2.2 outlines the participant recruitment process, offering insight into our study cohort's diversity and selection criteria. The section concludes with Subsection 5.2.3, presenting a summary of all the data collected while running the user study, and Subsection 5.2.4, which addresses the ethical approval and consent procedures.

5.2.1 | Parts and Procedure

Our study involved an interactive exercise in which participants used the custom deck of cards to outline a mobile app on a large paper sheet, facilitating a visual representation of the app's functionalities. Employing colour markers for notes and connections and with our guidance, participants explored potential features and limitations through an open dialogue devoid of strict time constraints. The objective was to delineate the envisioned app comprehensively.

Following the mapping exercise, participants proceeded to complete a survey designed to gather demographic information and detailed feedback on the experimental process. Additionally, each session was video-recorded to ensure thorough documentation of the discussions and insights generated, capturing nuances and reflections that contributed to a deeper understanding of participant experiences throughout the process.

The decision to focus on mobile applications rather than IoT solutions stemmed from the observation that laymen had a more concrete understanding of mobile apps, given their widespread familiarity after more than a decade of smartphone usage. In contrast, IoT remains newer and less familiar to the average person, who typically lacks computing or digital technology expertise. While not directly addressing IoT solutions, this focus yielded valuable insights into supporting non-experts in creating technology-centric solutions.

Each session was facilitated by the researcher, whose role was primarily observational but also responsive to the participant's needs. Participants arrived with a pre-formed, though often vague, idea for a technology-related project, and the session was framed as an opportunity to explore and refine that idea using the bespoke card deck. Given the diversity of participant backgrounds and project intentions, sessions were necessarily adaptive rather than standardised. The facilitation strategy followed a roughly 70/30 split between observation and intervention. While the researcher aimed to minimise influence and allow ideas to emerge organically, occasional guidance was provided to prevent participant frustration or impasse. The sessions were thus structured around a flexible, participant-led exploration of potential functionalities, rather than a uniform procedure. Ensuring procedural consistency across participants was not a methodological aim. Instead, the study sought to capture a range of engagements with the card-based toolkit, grounded in each participant's project intentions.

5.2.2 | Inviting Participants

We explored the conceptual card deck's impact through individual sessions within the study's framework. The core of our evaluation focuses exclusively on individual participants, each of whom came with a pre-conceived idea for a project they wanted to create. This targeted approach allows us to delve deeply into each participant's unique engagement and experiences with the toolkit. It provides a nuanced understanding of its effectiveness in supporting end-user developers during the project conceptualisation phase.

The recruitment process began with an announcement in the University Newsletter, enticing potential participants with the question, *"Isn't there an app for that?"* This initiative formed Sample A, comprising the first group of individual participants. The growing interest within the university community led to our involvement in the *"Health in the Digital Era"* course, marking our first foray into group-based participant engagement (Sample B).

Further expansion of our participant base occurred during the Summer of 2020-21, facilitated by a Summer Scholar who employed both personal outreach and digital communication, thereby creating Sample C. December of the same year saw us integrating with a Directed Study class at the University of Waikato Tauranga Campus, resulting in Sample D. Although this involved a collective class experience, our focus remained on the individual contributions within that setting.

While participants were recruited through a range of settings—individual outreach, academic programmes, and summer initiatives—the structure and focus of the sessions remained consistent. A key distinction lay in the initial framing: individual participants (Samples A and C) were explicitly invited based on having a personal project idea they wished to develop, whereas classroom-based participants (Samples B and D) encountered the toolkit as part of a course context. Nonetheless, once engaged with the card deck and related materials, participants across all samples interacted in comparable ways, using the cards to articulate, explore, and expand upon their project ideas. Although the classroom sessions occurred in group environments, our methodological focus remained on individual contributions, and each session was facilitated, recorded, and analysed accordingly.

Based on the engagement with 18 individual participants, our analysis provides a detailed examination of the conceptual card deck's utility and impact. This focus enriches our understanding of its role in facilitating the ideation and conceptualisation stages for end-user developers embarking on technology-driven projects.

5.2.3 | Data Collection

Our user study resulted in a significant data collection, comprising survey responses, photographs showing the cards in use and detailed video recordings. These recordings spanned 6 hours, 36 minutes, and 2 seconds and were transcribed into 53,607 words. On average, each participant's participation duration was 24 minutes and 45 seconds, yielding an average of 3,350 words per participant. The data also includes questionnaire responses containing demographic information and four questions about participants' prior experience and thoughts post-utilisation of the cards. The recordings captured participants' interactions with the instructional materials and their reactions during the interviews.

The analysis of the study data followed an inductive approach, consistent with the exploratory aims of the research. Rather than applying a formal coding schema or content analysis framework, patterns were identified through repeated engagement with the transcripts, supported by visualisation techniques. These included word frequency analysis and clustering strategies developed using ad hoc tools such as Python scripts and the SpaCy natural language processing library. This analytic process reflects a transitional phase in the research, moving from a design-led interpretative approach towards more structured methods typical of HCI. The analysis was conducted independently, with emerging findings subsequently discussed in supervisory meetings to validate interpretations and support thematic refinement. No qualitative software packages or inter-coder reliability procedures were applied at this stage.

5.2.4 | Ethical Considerations

The study received ethical approval from the School of Computing and Mathematical Sciences Ethics Committee at the University of Waikato. The approval letter is included in Appendix A. All participants provided informed consent before the session, which included the card-based activity, video recording, and post-session interview. No specific ethical concerns were identified regarding participant vulnerability, prior relationships, or data sensitivity.

5.3 | Analysis

The user study presented in this chapter aims to evaluate the usefulness of a bespoke deck of cards in mapping an idea for a technology-driven project that can lack a level of precision in the mind of an end-user developer. In the following subsections, we analyse the study's results and present useful insights that can be used to scaffold the needed support for end-user developers in creating tech-driven projects.

5.3.1 | Analysing the Survey

Following the completion of the user study, we have compiled a dataset amenable to quantitative analysis derived from the questionnaire administered to participants post-study, as detailed in Table 5.2. This survey was organised into two main sections: a) demographics and b) specific questions related to the study, the outcomes of which are discussed in the subsequent sections.

The questionnaire responses collected during the study were used primarily to contextualise the participant cohort and ensure that no outliers distorted the findings. Demographic data (e.g., gender, age, mobile platform) and self-reported impressions of the card-based activity were reviewed descriptively but not subjected to statistical analysis. The responses were not designed to support inferential claims, but rather to provide a basic characterisation of the sample and corroborate qualitative observations. As such, no detailed coding or metric-based interpretation was undertaken for open-ended answers, and findings from the questionnaire were not central to the study's analytical outcomes.

The analysis of the demographic data included 16 participants. Ten participants were male, and six were female, with all participants choosing to disclose their gender. The age distribution showed a concentration in the younger demographic, with ten individuals aged between 17 and 24. There were also two participants in the 25 to 29 age group, one between 40 and 45, another between 46 and 49, and two aged 50 or older (Figure 5.5). This diverse age and gender representation gave the study the condition that it was not particularly biased in a given direction.

Transitioning to the second segment of the survey focused on the participants' use of mobile technologies, their prior experiences with the concepts introduced by our study, the comprehensiveness of our toolkit, and its perceived utility. The participant group was equally divided regarding mobile technology usage, with eight individuals using Android and another 8 using iOS. Interestingly, 100% of the participants reported having no previous experience with a concept

TABLE 5.2: Participant's questionnaire.

Demographics	
D1	What is your gender?
D2	What is your age?
D3	Which mobile system do you use?
Questions	
Q1	Do you have previous experience with this type of exercise?
Q2	How useful do you find it?
Q3	Did the cards cover most of the aspects that you will consider?
Q4	Do you think that there are useful to have more ideas?
Q5	What are the two things that you like the most? Explain why.
Q6	What are the two things that you would like to improve? Explain why.
Q7	Do you have any other suggestions?

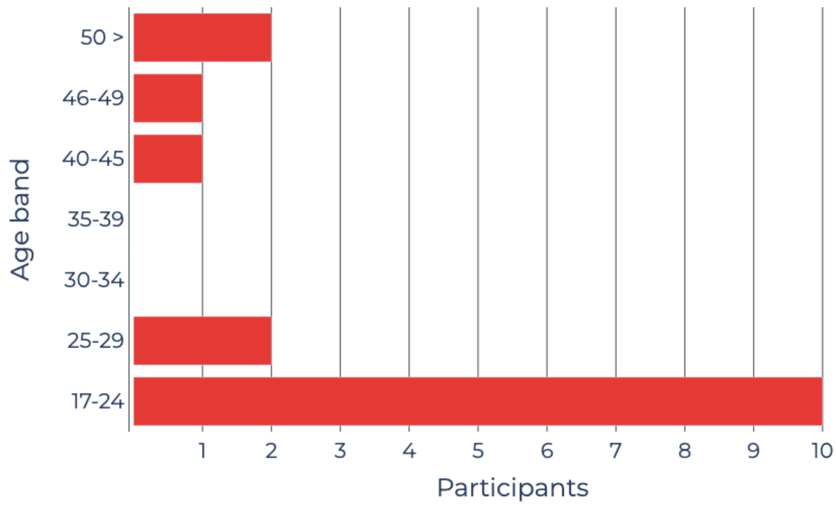


FIGURE 5.5: Age of the participants.

akin to our study's proposal. However, a majority acknowledged the utility of our toolkit: 10 participants rated it as *Very useful*, four as *Useful*, and 2 perceived it as *Average*, with no participants considering it *Not so useful* or *Not useful at all*—the other options provided on the Likert scale. Furthermore, 100% of the respondents agreed that the toolkit comprehensively addressed their creative needs. Regarding its contribution to generating more ideas, 9 out of 16 participants found it *Very useful*. At the same time, 7 rated it as *Useful*, with none rating it as *Average*, *Not so useful*, or *Not useful at all* (Figure 5.6).

The last three survey questions, Q5 to Q7, focused on what participants liked most about the exercise, what they felt could be improved, and any additional suggestions they had. Only five

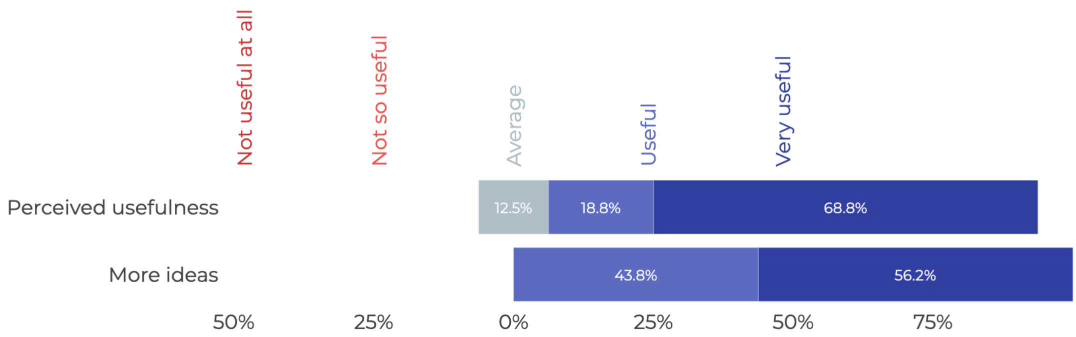


FIGURE 5.6: Perceived usefulness of the toolkit and increasing of quantity of ideas.

participants answered these questions, giving us feedback to help us understand their experience. The feedback from participants on questions Q5 and Q6 underscores a pronounced appreciation for the exercise's user-friendliness, particularly its visual design elements like colour and graphics, which facilitate easy idea visualisation and organisation.

About Q5, *"What are the two things you like the most?"*, participants noted:

- *"Good for drawing out ideas and thought process."* | PARTICIPANT N7
- *"Colours help to categorise each area, making it easy to use."* | PARTICIPANT N8
- *"Easy because it is colourful and colours help to categorise."* | PARTICIPANT N9
- *"Make developing base concept easy..."* | PARTICIPANT N10

Participants praised the exercise for its ease of use and the ability to categorise thoughts effectively through colour coding. Including elements allowing for user input and creativity, such as blank cards for personal ideas, was also highlighted as a key feature that enhances user engagement and satisfaction.

When asked about two things they would like to improve, participants mentioned the possibility of having more options and colours, showing a desire for an expanded range of options to stimulate creative thought further and provide new perspectives. Participants expressed a need for more diverse features within the exercise to broaden its applicability and enhance its capacity to inspire new ideas.

None of the participants responded to Q7, indicating potential satisfaction with the current setup or possibly uncertainty on what additional suggestions to offer.

The survey revealed that participants found the toolkit valuable in consolidating their ideas. This positive feedback underscores the toolkit's role in bridging the conceptualisation and practical implementation gap. A group of them appreciated how a set of colourful cards helped them organise their thoughts and better understand the project's requirements.

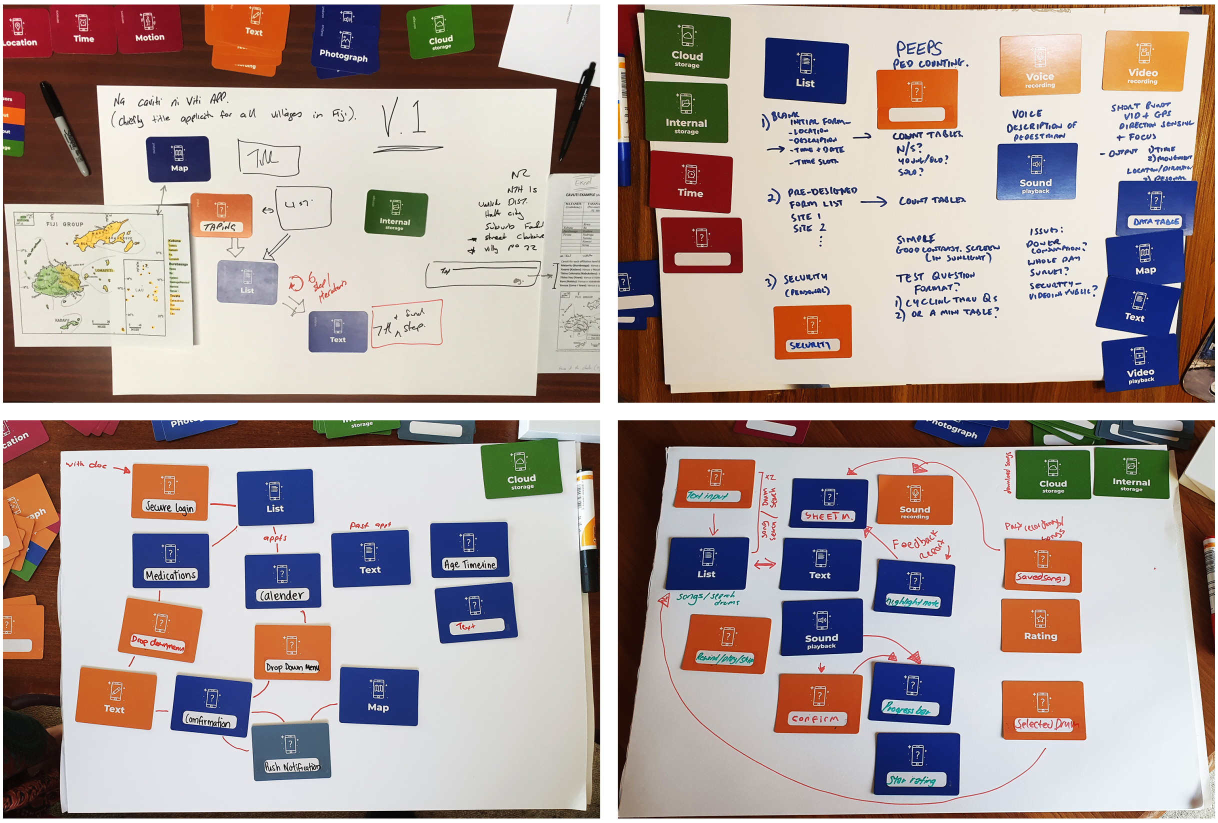


FIGURE 5.7: Cards used by participants N1, N8, N13 & N14.

5.3.2 | Observations from the User Study

The following paragraphs present observations from the user study. Through detailed analysis of participant engagement, tool utilisation, and facilitator interaction, we aim to unpack the dynamics of creative design processes in a controlled setting. Each subsection delves into a different aspect of the study: the duration and participant engagement, the impact of the ideation tools used, patterns in utilising the toolkit, and the integral role of the facilitator. By examining these elements, we seek to illustrate how structured tools and facilitation can significantly enhance the ideation phase in application design for end-user developers.

Duration and Engagement

The study did not impose a strict timeframe for session completion, allowing participants to engage with the exercise for up to one hour. Analysis of session lengths revealed an average duration of 32 minutes, ranging from 23 to 46 minutes. This variability underscores an effective engagement period, with participants typically requiring only thirty minutes to comprehensively explore the facets of their proposed mobile applications.

Impact of Ideation Tools

Participants' initial assumptions regarding the clarity of their application concepts were often challenged during the exercise. This re-evaluation process underscores the significance of iterative design and feedback in refining project ideas, an approach that end-user developers may not inherently possess. Utilisation of ideation tools, specifically a deck of cards designed to facilitate application planning, proved pivotal. Notably, 64% of participants acknowledged a shift in their perspective concerning their application's post-exercise development. This finding highlights the efficacy of tactile tools in enhancing cognitive processes implied in the early phases of creating a tech-driven project.

Refining or Altering Initial Ideas Based on Technical Information

Participants provided various testimonies that illustrate how their initial ideas were refined or altered based on the technical information provided by the cards. These insights can be categorised into five key areas: *Storage Decisions*, *Data Management and User Interaction*, *Sensors Use*, *Output and Media*, *General Feedback and Utility*.

Storage Decisions Participants often re-evaluated their storage choices based on the cards, deciding between internal and cloud storage to meet their project's needs.

– *"Zero connectivity. It needs to be internal storage."* | PARTICIPANT N1

– *"I think as soon as they knew it was on the cloud, they're going to get rid of the data on the internal storage."* | PARTICIPANT N2

– *"Yeah, so definitely stored on the cloud..."* | PARTICIPANT N11

– *"I definitely think that cloud storage is important, but you would still have some internal storage, you know, the map system things."* | PARTICIPANT N16

Data Management and User Interaction The cards helped participants reconsider how they managed data and interacted with users, leading to more robust and user-friendly results.

– *"Yeah, pretty much creating an account... Yes, so they create an account first. That's such a good idea. I'm not good with anything technical."* | PARTICIPANT N3

– *"I suppose there'd then be an opportunity for more text to be input or edited or something later..."* | PARTICIPANT N12

– *"Yeah, like you could click the notification, you know how you get a text message you say like mark as read or reply something, you could just do that, with the notification..."* | PARTICIPANT N13

Sensors Use The cards provided technical insights into sensor functionalities (e.g., motion), which helped participants better integrate these features into their projects.

– *"Okay, because if they do voice recording, that would be more data, right?"* | PARTICIPANT N3

– *"I was just thinking, do you have any others? Because the motion is also helpful, because I was thinking you are moving, and then say, did you forget to log in or something?"* | PARTICIPANT N4

– *"So we were thinking that you would, so these things [sensor cards], do I also understand the process?"* | PARTICIPANT N5

– *"So location is the phone knowing where the person is that's using it?"* | PARTICIPANT N7

- *"Yeah, yeah, the output is time, movement which is at location X moving in direction Y. So location and direction."* | PARTICIPANT N8
- *"Yeah, your actual location and where you are on a map... So that would be location as well..."* | PARTICIPANT N10
- *"So probably definitely time, so I think the app needs to record, well it will have to record time..."* | PARTICIPANT N11
- *"It would be motion because I feel like they'd want to go in and look around first."* | PARTICIPANT N15

Output and Media Participants refined their ideas about outputs and media integration, considering how various media elements would function within their applications.

- *"Yes, so this is like the social media sharing. So this could be done also automatically?"* | PARTICIPANT N4
- *"A sound, it could be a sound actually."* | PARTICIPANT N6
- *"Yeah, like take photos and videos and save them to their phone..."* | PARTICIPANT N9
- *"So green is just storage, blue is output in the app... Do you have an output outside of the app?"* | PARTICIPANT N5
- *"Yeah. Definitely photos, I think, like user photos, things like that."* | PARTICIPANT N16

General Feedback and Utility Participants provided general feedback on how the cards helped them solidify their ideas and better understand the technical process, reflecting their overall utility.

- *"Yes, this has been very helpful to really solidify what I can think of as well."* | PARTICIPANT N4
- *"So we were thinking that you would, so these things [sensor cards], do I also understand the process?"* | PARTICIPANT N5
- *"So I want it to be able to, like, listen to your playing as well, like, as an option to, like, receive feedback."* | PARTICIPANT N14
- *"I mean, I don't know about like a, like sound recording or voice recording, like if you want to, like, record your ideas on the app, if that makes sense?"* | PARTICIPANT N15

Card Utilisation Patterns

On average, participants engaged with 13 out of the 48 cards available per session (Figure 5.8). This level of engagement indicates that the card deck is mainly effective in capturing participants' interest and facilitating their ideation processes. The frequent selection of *"Undefined"* cards, in particular, suggests potential areas for expanding the card deck to better cater to user needs and stimulate further creativity. This behaviour highlights gaps in the current deck where additional predefined categories might be beneficial. Moreover, using *"Undefined"* cards shows

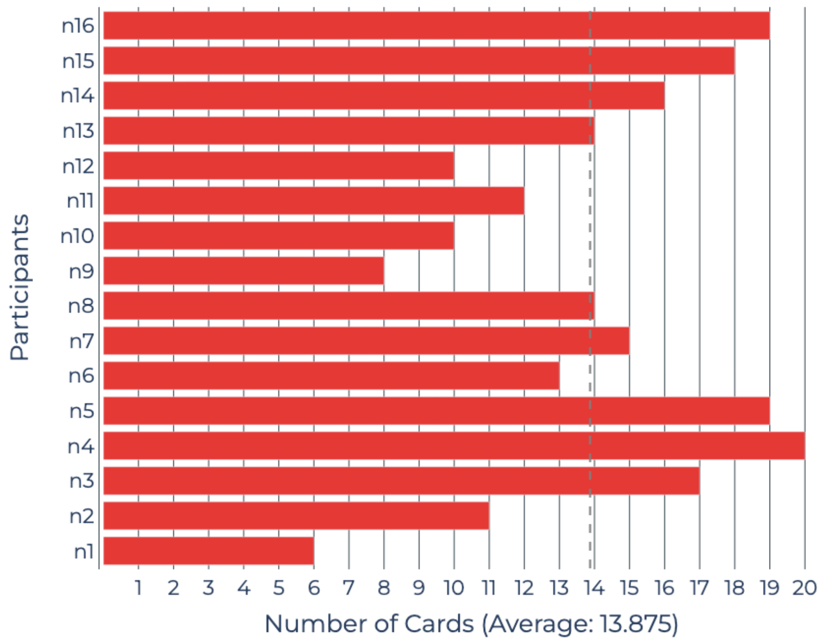


FIGURE 5.8: Number of cards used by participants in the study.

that participants are thinking beyond the existing cards and categories, indicating that the deck successfully encourages creative thinking and idea generation. These insights are invaluable for guiding the iterative design process, helping to refine and improve the deck based on user behaviour and preferences. Ultimately, this iterative approach aims to create a more user-friendly and effective tool for end-user developers. By understanding which cards are frequently used and where users need to go beyond predefined categories, a more comprehensive and adaptable tool can be developed to meet the diverse needs of end-user developers. This data validates the deck's practical impact on project conceptualisation and underscores the importance of customisation and flexibility in design tools.

Role of the Facilitator

The facilitator, acting as a researcher, crucially guided participants through the mapping process of the mobile application using the card deck. This guidance was pivotal in helping participants navigate technical constraints and explore creative possibilities. The facilitator's role in the user study extended beyond guiding the ideation process; they also helped participants understand and manage technical limitations and opportunities. While the facilitator did not dictate the use of the cards or the sequence of the session, strategic interventions and prompts were significantly impactful. These interventions often made participants realise that their exploration of ideas was not exhaustive, prompting further refinement and development of their application concepts.

Without the assistance of a facilitator, end-user developers may easily encounter limitations and roadblocks during the ideation and conceptualisation stages. Although the card deck can be used independently, these users might struggle to explore and refine their ideas fully, highlighting the challenges they face without expert guidance.

Additional Notes

During the final part of the study, some participants could articulate a few words to express an evaluation of the toolkit in assisting them with the process:

- *"Yeah that's made me so much clearer of what I want..."* | PARTICIPANT N3
- *"This is a really cool tool..."* | PARTICIPANT N5
- *"These cards are really helpful. Like, I would not be able to remember anything I just said!"* | PARTICIPANT N15
- *"No, I think it's, it's getting there, like it's a bit more progress than I had expected."* | PARTICIPANT N16

Although minimal, as clearly participants were more excited speaking about their idea for a project than about the usefulness of the toolkit, these quotes represent the overall perceived effect of using the cards.

Given the exploratory and design-led nature of the study, no formal participant validation procedures (such as member checks) were employed. To mitigate potential bias arising from the dual role of facilitator and analyst, the facilitation was deliberately kept to a minimum, and the subsequent analysis focused strictly on participants' verbal responses and observable actions during the session. Interpretive attention was directed toward how participants interacted with the toolkit and articulated their ideas, rather than on the viability or interest of their proposed projects. This stance reflects a design-oriented approach to qualitative inquiry, prioritising situated observation and emergent meaning-making over structured validation. While this represents a departure from more formalised protocols in HCI, care was taken to maintain fidelity to participants' intentions and perspectives as they engaged with the materials.

5.4 | Discussion

This chapter presented a user study exploring the value of a bespoke deck of cards in assisting end-user developers to consolidate their ideas. The study goals were oriented to accomplish these objectives:

- **Evaluate the efficacy of the conceptual cards**
  The primary goal of the study was to assess how effective the conceptual cards are in supporting end-user developers during the ideation phase. This includes understanding how the cards influence end-user developers' ability to generate, refine, and articulate project concepts.
- **Validate the hypothesis of structured support**
  The study aimed to validate that providing structured support in conceptual cards can enhance the ideation process for end-user developers. It seeks empirical data to confirm whether such support helps developers overcome conceptual vagueness and leads to more concrete, actionable project concepts.

– **Gather insights for refinement**
  A third goal of the study was to gather insights on refining the conceptual card toolkit further to meet the diverse needs of end-user developers. This involves understanding how developers interact with the cards, identifying any challenges or limitations, and determining potential improvements or additional features to enhance the toolkit's effectiveness.

We can assert that all three goals were achieved based on the observations from the user studies and participants' feedback. Firstly, the cards proved highly effective in assisting end-user developers to generate, refine, and articulate project concepts, as evidenced by the study's outcomes. Secondly, the structured support provided by the cards facilitated participants in progressing with their ideas and overcoming initial conceptual vagueness, resulting in more tangible and actionable project concepts. These refined concepts can now serve as a foundation for further collaboration with specialists for project development or guidance. Lastly, the study yielded valuable insights for refining the toolkit. Participants highlighted the need for more precise rules within the toolkit and desired a format they could navigate independently without researcher assistance. This insight could guide future iterations of the toolkit to ensure it effectively meets the diverse needs of end-user developers.

5.5 | Summary

This chapter's user study assesses a method for assisting end-user developers in refining their ideas, particularly in overcoming the hurdles inherent in digital-oriented project creation. The core hypothesis of this thesis posits that, with adequate support, end-user developers can effectively construct customised IoT projects, beginning with a precise delineation of goals and requirements. Although the user study focused on mobile application development, the insights gained—such as the value of integrating technical guidance into the ideation phase—can be extended to the development of customized IoT projects, where similar challenges of aligning creative ideas with technical realities exist. The study's findings substantiate the feasibility of this notion, highlighting the intricacies involved and the need for tailored assistance. Recent literature shows that toolkits using participatory design cards can help users explore the technical constraints and affordances of IoT systems, enabling them to integrate these considerations into their project planning. Although primarily aimed at designers, these toolkits can also support ideation in IoT projects by helping participants align their goals with technical possibilities (see Table 3.1 for a list of relevant toolkits).

The cards assisted participants in organizing their thoughts and incorporating technical vocabulary, allowing them to balance creative ideas with practical constraints and possibilities. Integrating technical guidance into ideation tools is crucial to effectively support end-user developers, ensuring that creative concepts are innovative and feasible.

Throughout the user study, it became evident that end-user developers encounter difficulties defining their objectives and navigating the path to achieving them. While the cards facilitated idea organization, the presence of a researcher offering additional technical support was critical in helping participants overcome specific challenges. This indicates that, while ideation tools are beneficial, end-user developers may still require expert input to fully navigate complex technical aspects.

The following chapter advances this investigation by evaluating a method to assist end-user developers in creating personalised IoT projects, specifically focusing on overcoming challenges related to *building electronic circuits* and *programming computers*.

Chapter 6

Project Enabler

User Study on Support for Project Assembly

> *If necessity is the mother of invention,*
> *then combining previous systems is the father,*
> *and adhocism is the creative offspring.*
> *This is true in both nature and culture.*
>
> — **Charles Jencks and Nathan Silver**
> *Adhocism: The Case for Improvisation* [94]

A LONG-TERM STUDY, presented in Chapter 4, observed an end-user developer creating an IoT project. The study identified a range of key threshold skills, including *building electronic circuits* and *programming computers*. These foundational technical abilities, such as basic circuit design and programming, are essential for progressing beyond the conceptual phase of IoT development. This chapter presents a user study exploring how end-user developers can be supported in overcoming obstacles related to *circuit building* and *computer programming* to create a functional prototype of a bespoke IoT project.

Through the user study, we aim to address RQ3:

> **RQ3** | What essential support strategies enhance end-user developers' success in building and programming IoT projects?

We have extended the *TILES IoT Toolkit*, which was originally designed to support ideation and prototyping, to further assist end-user developers in the more technical stages of building and programming IoT projects. This extension includes additional resources and structured guidance aimed at simplifying these technical tasks, bridging the gap between abstract concepts and fully functional prototypes. Our approach builds on the literature reviewed in Chapter 3 and the user study presented in Chapter 5. We know from the literature that toolkits provide adequate support for defining and ideating IoT projects. The conceptual cards in Chapter 5 helped participants

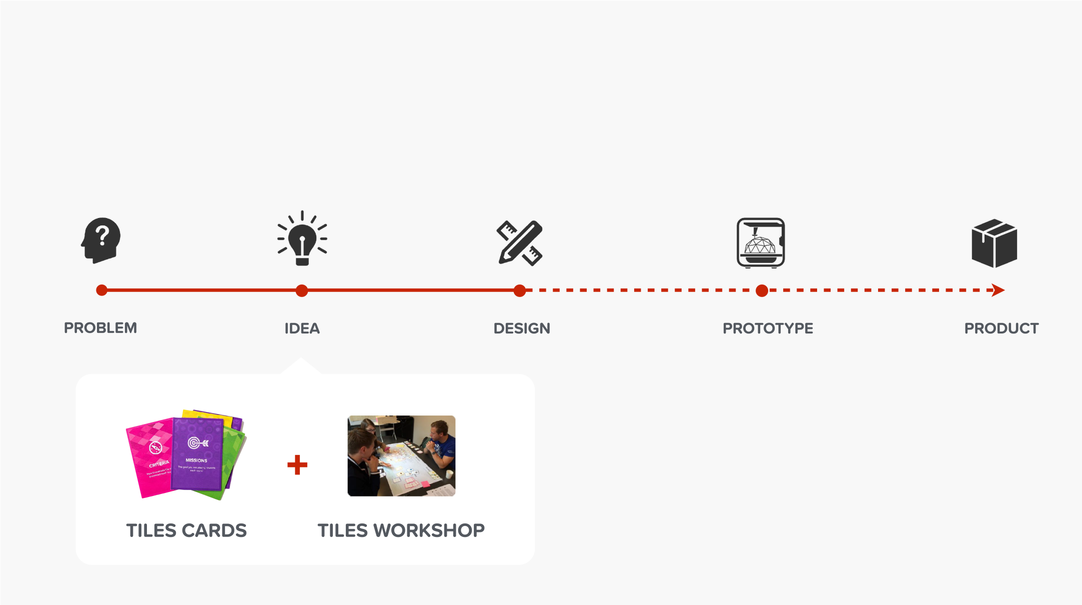


FIGURE 6.1: TILES IoT Toolkit workshop, focusing on the first phases of the process. Reproduced from *Tiles IoT Inventor Toolkit*, https://www.tilestoolkit.io.

clarify and organise their ideas during the ideation phase. However, as this chapter focuses on the technical challenges of building and programming, additional support is required to overcome these significant hurdles.

The chapter continues with the following sections: Section 6.1, presenting how we propose to extend a preexisting toolkit; Section 6.2, explaining how we carried out the study; Section 6.3, presenting the analysis of the data gathered in the study; Section 6.4, exposing a discussion of the insights obtained as a result of the study; and finally Section 6.5, summarising the chapter.

6.1 | Extending the Tiles IoT Toolkit

The creation of bespoke IoT projects presents different phases. After initial phases belonging to *the world of thinking*, such as definition and ideation, end-user developers would move to *the world of making*, in creating a project and materialising their ideas. Currently available toolkits adequately support the initial phases of definition and ideation (see TILES IoT Toolkit workshop focus on Figure 6.1). Still, how to best support them in making a functioning IoT project remains unclear. We consider that presenting concrete components and guiding end-user developers through a defined set of instructions could be beneficial and bridge the gap between these two worlds.

Chapter 3 showed that the TILES IoT Toolkit offers a clear orientation to support end-user developers in a project's definition and ideation phases. However, we consider that the toolkit provides a high level of abstraction that needs to be bridged to complete an actual project. We proposed to extend it to provide more concrete information, connecting it closer with the process of making. With the extended toolkit, we aim to test if end-user developers could assemble their

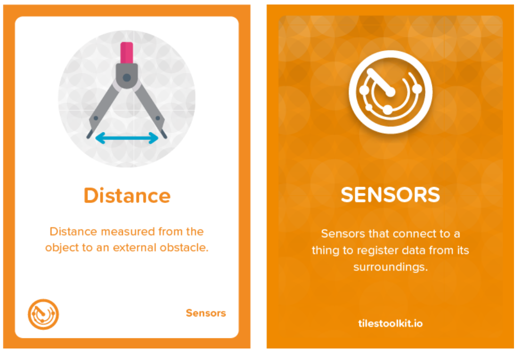


FIGURE 6.2: TILES IoT Toolkit Distance card (front and back).
Reproduced from *Tiles IoT Inventor Toolkit*, https://www.tilestoolkit.io.

own bespoke IoT project using a given set of components and related instructions.

The TILES IoT Toolkit cards present the core information on the front face, while the back indicates the group to which a particular card belongs (Figure 6.2). By providing more information on certain cards, end-user developers can be better supported in building prototypes of their ideas. Given a card's limitations, we propose connecting them to web resources using QR codes. Furthermore, presenting photographs of given electronic components could be beneficial for end-user developers that have no familiarity with electronic components.

Using the cards in the Sensors group as an example, the front face of the cards shows a symbol corresponding to a type of sensor (e.g., an illustration of a measure gauge distance compass to represent a distance sensor). These cards can be extended by presenting a photograph of a specific example of such a sensor on the back face (e.g., a Grove - Ultrasonic Distance Sensor), using concrete objects instead of a symbolic representation of the action in the card (Figure 6.3). Furthermore, the cards can be used to create a connection to specific web resources by using QR codes. We believe that to assist end-user developers effectively, their efforts should be centred around addressing the specific requirements of the project they are working on. This approach is preferable to having them sift through extensive, generic documentation without a clear direction. Following the example provided by the Distance sensor card, the QR code connects the card to a custom-made website with instructions for assembling a measuring distance device using specific hardware and software components[1].

6.2 | Study Method

We conducted a user study using the extended cards and the dedicated website we have created by extending the TILES IoT Toolkit. The focus of the study is to test if an end-user developer provided with the relevant support can successfully build a bespoke IoT project. Subsection 6.2.1 presents the preliminary steps in preparation for the study, Subsection 6.2.2 explains the phases

[1]https://faitap.com/sensors/distance/

FIGURE 6.3: TILES IoT Toolkit Distance card – proposed extension (front and back).

and procedure of the study, Subsection 6.2.3 presents how participants were invited, Subsection 6.2.4 details the structure and methodological rationale of the data collection process, and Subsection 6.2.5 presents the ethical approval details and consent procedures followed in the conduct of the study.

6.2.1 | Preparation

Chapter 3 showed that the TILES IoT Toolkit could support end-user developers in an IoT project's defining and ideation phases. We propose extending the functionality of the TILES IoT Toolkit by incorporating QR codes on cards that could link to specific items, such as a particular type of sensor (Figure 6.3). The QR code directs the end-user developer to an instructional web resource. The extra information helps transition from abstract concepts to concrete examples: a distance sensor becomes a specific electronic component (in the upcoming paragraphs, as we examine potential scenarios and make a selection for the user study, we elucidate the reasons behind our decision to utilize the Grove – Ultrasonic Distance Sensor, produced by Seeed Studio, specifically for the distance-sensing project). With the instructions on that web resource, an end-user developer gets closer to putting together a functioning artefact using hardware, software and cloud services.

To explore the notion of supporting the cards with extra content, we started by imagining simple potential problems that could be solved by creating a bespoke IoT project. As a result of that process, we ended with four different concepts:

– a noise level measuring project,

– a standing desk measuring device,

– a switch the lights project, and

– a squeeze stress balls device to control lights.

Having defined four potential problems, we used the TILES IoT Toolkit to explore these ideas and gain a deeper understanding of the implications of each, developing different scenarios based on our findings (Figure 6.4).

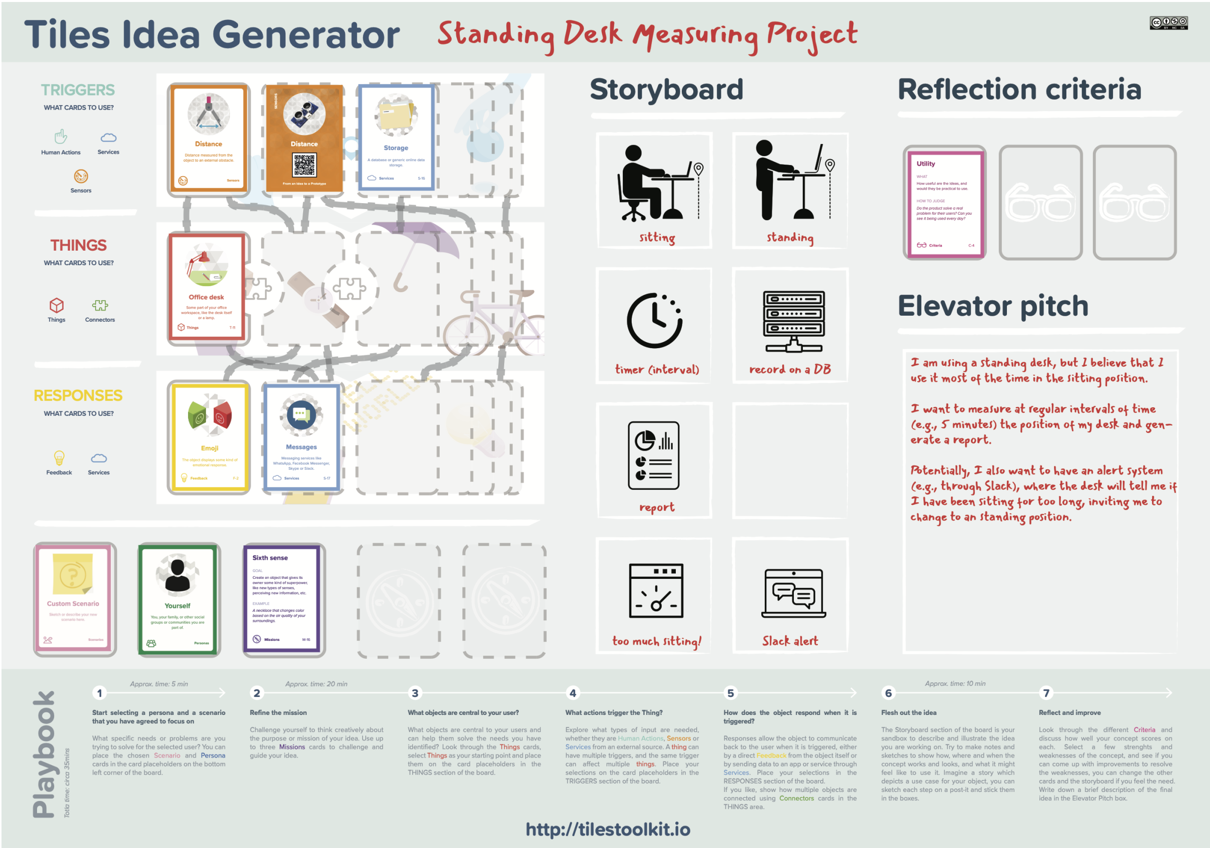


FIGURE 6.4: Scenario used in the study presenting a distance measuring device.

Before conducting the study, the four scenarios were compared in our lab, following the required steps to complete the various projects. It was identified that the scenarios required different sensors and, in some cases, actuators.

Scenario A – a noise level measuring project – required a loudness sensor. Transforming the values captured by the sensor into a measurement recognisable by non-experts (e.g., decibels) proved to be complicated. The value of decibels is not linear in relation to the analogue readings from the sensor. Therefore, the sensor's results must be calibrated using a Sound Pressure Level (SPL) meter.

Scenario B – a distance measuring device – required an ultrasonic distance sensor (Figure 6.4). During our lab testing, it became evident that the sensor and its associated library were straightforward to use and did not pose significant challenges. Based on our testing, we deem it appropriate for a project undertaken by an end-user developer.

Scenario C – a light switcher – required a tilt sensor and light actuators. The sensor was simple to use, but including actuators increased the level of complexity of the project.

Scenario D – using squeezy stress balls to control lights – required several force-sensing resistors (FSR) and light actuators. While the sensors were not difficult to operate, their integration with stress balls added a layer of complexity to the project. The use of actuators also increased the complexity of the project. This scenario is the base for the long-term study presented in Chapter 4.

TABLE 6.1: Comparison of user satisfaction across four scenarios.

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
Overall, I am satisfied with the ease of completing the tasks in this scenario.					
Scenario A – Noise level measuring device	✓	○	○	○	○
Scenario B – Distance measuring device	○	○	○	○	✓
Scenario C – Light switcher	○	○	○	✓	○
Scenario D – Stress balls light controller	○	✓	○	○	○
Overall, I am satisfied with the amount of time it took to complete the task in this scenario.					
Scenario A	✓	○	○	○	○
Scenario B	○	○	○	✓	○
Scenario C	○	○	✓	○	○
Scenario D	○	✓	○	○	○
Overall, I am satisfied with the support information (on-line help, messages, documentation) when completing the task.					
Scenario A	✓	○	○	○	○
Scenario B	○	○	○	○	✓
Scenario C	○	○	○	✓	○
Scenario D	○	○	✓	○	○

It proved to present a high level of complexity for an end-user developer.

After completing the four proposed projects, we conducted a pre-study self-evaluation to assess their relative complexity and feasibility. This evaluation was carried out using the After-Scenario Questionnaire (ASQ) [107], which comprised three statements rated on a Likert scale from *"Strongly disagree"* to *"Strongly agree"*:

- Overall, I am satisfied with the ease of completing the task in this scenario.
- Overall, I am satisfied with the amount of time it took to complete the task in this scenario.
- Overall, I am satisfied with the support information (on-line help, messages, documentation) when completing the task.

FIGURE 6.5: Distances to measure in Scenario B – a distance measuring device.

The results of this assessment are presented in Table 6.1. Scenario B—a standing desk measuring device—received the highest average ASQ rating (4.67/5), indicating the greatest overall ease, efficiency, and support quality. It was identified as the most straightforward to implement in terms of required components and clarity of procedure. Based on this, we selected Scenario B for the user study.

The project required a distance sensor, a microcontroller and data storage in the cloud. The following subsections explain how we defined the sensor, microcontroller and cloud service used to run the study.

Defining the sensor

Distance sensors use ultrasonic or laser to measure the distance among objects. There are many measuring distance sensors in the market, but our scenario presents specific characteristics that simplify the selection process. Used in a sitting position, the tabletop of a desk is 65-75 cm from the floor. In a standing position, the tabletop is 95-105 cm from the floor (Figure 6.5). Therefore, the sensor in this project needs to measure distances in the 65-105 cm range. A second requirement was related to simplicity. Aiming at being built by end-user developers, the sensor must have ways of being connected without the need for soldering cables, and the device has to be more robust than a circuit using a prototyping breadboard. Given those requirements, we decided to use a Grove Ultrasonic Ranger sensor[2] manufactured by Seeed Studio. The measuring range of this sensor is 2-350 cm, and it uses a Grove connector.

The Grove system, created and released in 2010 by Seeed Studio, is an open-source, modular component system that simplifies prototyping. Bell [16] explains the three main characteristics of the Grove system: 1. Modular cabling supporting four protocols (I2C, digital, analogue, and UART), 2. Easy, polarised connectors (no incorrect or reversed connections), and 3. No soldering required. The characteristics of the Grove system present a straightforward method for end-user developers to create electronic circuits, mainly by removing the need for soldering components.

Defining the microcontroller

The IoT project presented in Chapter 4 used Arduino technology. Our observation showed us that Arduino was straightforward and well-supported for an end-user developer. Based on our previous experience, we decided to use an Arduino MKR WiFi 1010 single-board microcontroller for the user study presented in this chapter. The Arduino website offers tutorials that effectively

[2] https://wiki.seeedstudio.com/Grove-Ultrasonic_Ranger/

guide utilizing this microcontroller in straightforward IoT projects. A large community of users actively participating in forum discussions could be consulted if needed. Therefore, we considered the Arduino MKR WiFi 1010 suitable for the IoT project in our scenario.

A Grove shield is needed to connect Grove modules to the Arduino board. Seeed Studio offers four kinds of Grove shields. The specific one for the Arduino MKR board is the Arduino MKR Connector Carrier[3]. The Arduino MKR Connector Carrier is a printed circuit board (PCB), providing Seeed Studio's Grove connectors to an MKR board. The microcontroller is plugged into the board socket. All the microcontroller ports become available through Grove connectors on the sides of the PCB.

Defining the cloud service to store data

To align with the IoT project description in Chapter 1, the user study scenario requires storing distance sensor data using a cloud service. In February 2019, Arduino launched the Arduino IoT Cloud platform [166]. After choosing the Arduino microcontroller and evaluating Arduino IoT Cloud's performance, we confirmed its suitability for this chapter's user study.

Instructional website

To support the creation of a prototype of the defined scenario, we created an instructional website with a step-by-step tutorial[4]. The QR code on the Sensor Distance card (Figure 6.3) directs to the website.

The instructional website created for this study consists of four sections: 1. Introduction, 2. Hardware, 3. Software, and 4. Final notes. The completion of the prototype requires participants to navigate through a series of twenty pages. This process is structured around a wizard-based approach, wherein the intricate task is systematically divided into a sequence of manageable steps. Throughout the user study, participants follow these distinct steps, culminating in the assembly of the prototype upon successful completion.

The Introduction presents the instructional web resource, indicates the minimum requirements, and ends by presenting a Table of Contents (Figure 6.6).

Following the steps in the Hardware section, participants familiarise themselves with the components: an Arduino MKR WiFi 1010 single-board microcontroller, a Grove ultrasonic ranger distance sensor, and a Grove Shield. We ask them to put together the circuit using those components. The instructions allow participants to stop and celebrate micro-achievements before stepping into the next part.

The Software section starts by presenting the Arduino Cloud service. We invite participants to follow the steps to create a thing, add a variable, make the microcontroller IoT-ready, configure the WiFi access, copy-paste the required lines of code and compile and upload the code to the Arduino board. When the project outputs values to the Serial Monitor, indicating that it works

[3] https://store-usa.arduino.cc/products/arduino-mkr-connector-carrier-grove-compatible

[4] https://faitap.com/sensors/distance/

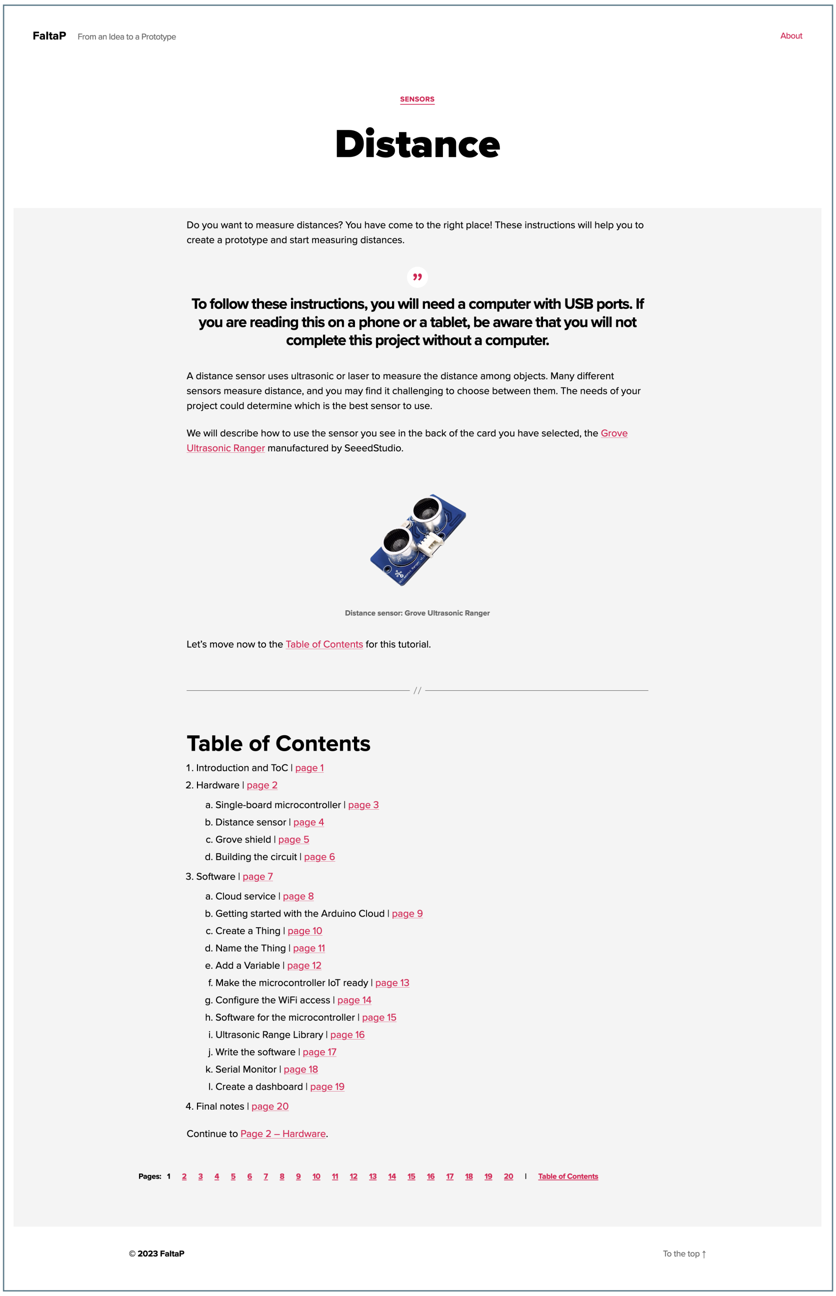


FIGURE 6.6: Initial page of the instructional web resource, accessible to users via the QR code on the Distance card (See Distance — FaltaP).

properly, we ask them to create a dashboard in the Arduino cloud.

6.2.2 | Parts and Procedure

The user study consists of three distinct parts:

1. a survey,
2. the act of putting together the prototype following online instructions, and
3. a semi-structured interview.

First part: survey

The study's first part presents a survey consisting of two sections: demographics and knowledge and experience (Appendix D). The demographic section asks for the age and education level. The knowledge and experience section inquires about the participant's familiarity with digital technologies, including the Internet of Things (IoT), and their awareness of community resources and support.

Second part: building the prototype

In the second section, we introduce the prototype-building exercise. This process was presented to participants as follows:

- **Initial presentation**
  Participants are introduced to the TILES IoT Toolkit cards, which have been enhanced for clarity by incorporating visual elements like photos. These cards are paired with corresponding instructions accessible via QR codes. This step establishes a link between the physical elements and digital guidance, setting the stage for the project assembly process.

- **Scenario introduction**
  Participants are then presented with the project scenario along with the necessary equipment. Additionally, they are shown a card that is connected to a web resource through a QR code. This card-web resource connection serves as the primary source of instructional guidance.

- **Guided step-by-step process**
  Participants are invited to follow the instructions presented in a wizard-based format through the web resource. This process is intentionally crafted to offer a guided series of instructions, guaranteeing that participants can smoothly navigate through the assembly process.

- **Evaluation of assembly capability**
  The study assesses whether participants can effectively assemble the prototype using the provided materials and guidance. Successful completion of this step indicates that the combination of physical elements and digital instructions effectively facilitates project assembly.

- **Response to challenges**
  As participants progress through the guided instructions, the study also aims to observe their reactions and actions when encountering challenges or inconveniences. This insight provides valuable information about how users handle unexpected issues and whether the provided support is sufficient to overcome such hurdles.

Participant completion of the prototype-building exercise was assessed through a structured, step-by-step web-based guide consisting of 20 sequential pages. The guide's different stages incorporated micro-achievements, culminating in a fully functioning IoT prototype by the final page. Successful completion was thus clearly observable, with the functioning prototype serving as an objective indicator. Although assistance was occasionally provided, these instances were not systematically recorded. However, each session was video-recorded, and field notes were taken to document participant behaviour and interaction with the materials. Completion of the prototype also marked the transition to the final questionnaire and semi-structured interview, which provided opportunities for reflection on the process.

Participant behaviour during the prototype-building phase was recorded and later transcribed. These transcripts, together with brief facilitator notes taken during the sessions, were used to inform post-session interpretation. Although actions were not categorised using a coding framework, the documentation of how participants progressed through the step-by-step tutorial, including points of hesitation or success, played a formative role in shaping the subsequent interview questions and interpretative direction of the study.

Third part: semi-structured interview

Finally, a semi-structured interview presents a series of open questions to understand participants' perspectives on the project (Appendix C). The interview aims to get their reflections after completing the project, asking participants to describe the project in their own words, explain how they perceived the time it took, and how their learning curve was in the project. Questions on failures and successful moments allow us to understand the process's bottlenecks and identify potential solutions that could better support end-user developers.

The study ends by thanking their participation and inviting them to participate in a follow-up study in the future.

6.2.3 | Inviting Participants

We created a series of posters to invite participants to the user study. We put them up around the campus of the University of Waikato, focusing on locations further away from the School of Computing and Mathematical Sciences or the School of Engineering. Students and staff in those schools are too technically oriented to participate in the study as end-user developers. The posters included a QR code that directed potential participants to a Google Forms page where they could express their interest in participating. Twenty-one people responded to the call, and twelve participants participated in the study between August 2021 and March 2022.

6.2.4 | Data Collection Structure

All data for the study were collected during a single session, comprising three sequential components: an initial survey, the prototype-building activity, and a concluding semi-structured interview. These components were designed to operate independently, with no intentional influence between them from a study design perspective. The survey provided demographic and perceptual data, primarily used to confirm the overall coherence of the participant group and to identify potential outliers. It was not used to frame or adapt the interview. The survey and interview data were therefore treated as parallel strands, each contributing distinct insights to the study's broader aim of understanding end-user developers' engagement with instructional IoT tools.

6.2.5 | Ethical Considerations

The study received ethical approval from the School of Computing and Mathematical Sciences Ethics Committee at the University of Waikato. The approval letter is included in Appendix A. All participants were informed of the study procedures and provided written consent before participating in each component of the session: the survey, the prototype-building activity, and the semi-structured interview. No specific ethical concerns were identified regarding participant vulnerability, prior relationships, or data sensitivity.

As a result of conducting the study with twelve participants, we collected the survey's responses and 14 hours and 27 minutes of video recordings. The recordings included the time taken by the participants to put together the prototype and the semi-structured interviews.

6.3 | Analysis

In the following pages, Subsection 6.3.1 presents a quantitative analysis of the preliminary survey, Subsection 6.3.2 shows notes coming from observing the process followed by participants putting together the prototype, and Subsection 6.3.3 presents a qualitative analysis of the recordings, including both the phase of assembling the prototype and the semi-structured interviews run after that.

Brumby et al. [21] remarked that understanding people's interactions with technology requires various methodological approaches. In the analysis, we used quantitative and qualitative methods. Among different quantitative methods, surveys provide a way to document people's characteristics, opinions, attitudes, or previous experiences [106]. In the context of HCI, qualitative research emphasises understanding the qualities of a particular technology and how people use it in their lives, how they think about it, and how they feel about it [2].

6.3.1 | Preliminary Survey: Quantitative Analysis

The survey was divided into two primary sections: a) demographics; and b) knowledge and experience (refer to Appendix D). In the first section, our objective was to ensure a well-balanced distribution of participants, avoiding concentration within specific age ranges or education levels (for example, a disproportionate number of participants aged 18-24 with high school degrees). In the second section, labelled "Knowledge and Experience," we aimed to gauge participants' familiarity with digital technologies. We sought to gain insight into their relationship with digital devices and services, as well as their level of awareness concerning public resources such as internet forums, which they might turn to for assistance if the project posed challenges during assembly.

As captured in the first section of the survey, the participants' age was well distributed. Three of them were in the 18-24 band, two in the 25-34 band, two in the 35-44 band, four in the 45-54 band and one in the 55-64 band (Figure 6.7). Regarding education, the distribution was also wide (Figure 6.8): one had a high school degree, three had a Bachelor's degree, five had a Master's

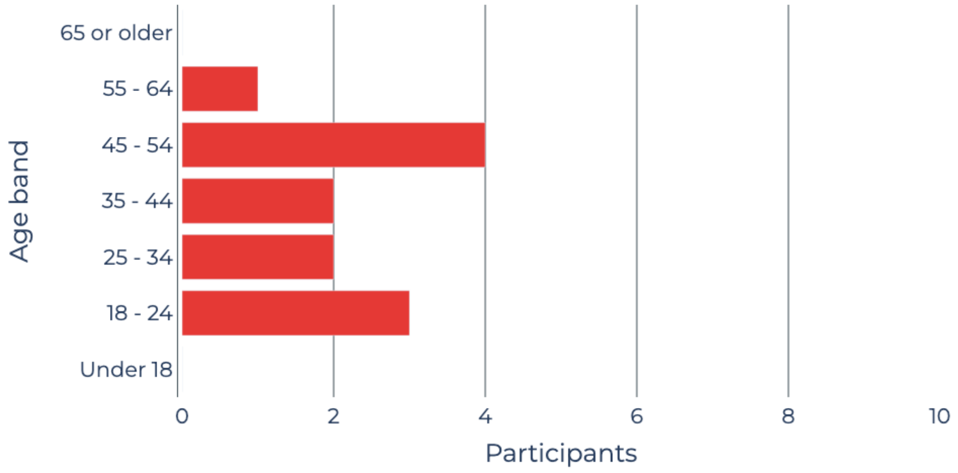


FIGURE 6.7: Age of the participants.

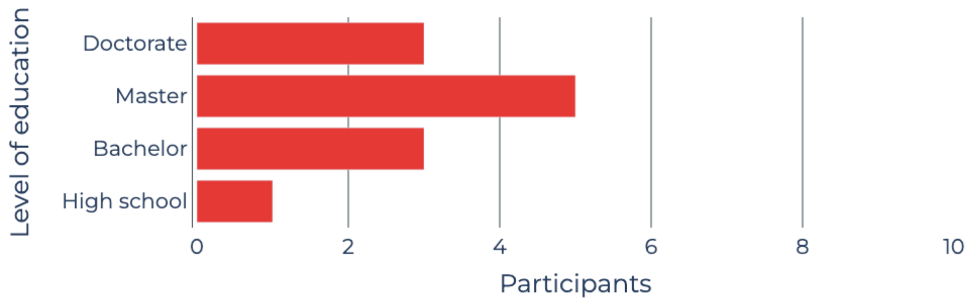


FIGURE 6.8: Highest degree of education completed.

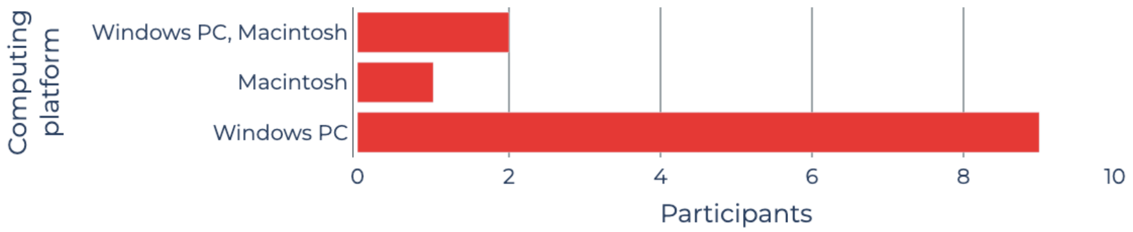


FIGURE 6.9: Computing platforms commonly used.

FIGURE 6.10: Mobile operating systems used.

degree, and three had a Doctorate.

The next section of the survey was oriented to capture participants' knowledge and use of digital technologies. We presented digital technologies to participants as electronic tools, systems, devices, and resources that generate, store, or process digital data (in uniform 0–1 form). Digital technologies include websites, mobile phones, digital televisions, and digital music players.

The first question of that section asked participants which computing platforms they commonly used, with options for Windows PC, Macintosh and Linux. Two participants indicated that they use Windows PC and Macintosh, one indicated Macintosh and nine marked Windows PC (Figure 6.9). The situation of most participants commonly using a Windows PC system could have represented an issue, as the study ran on a Macintosh computer. But in practice, the study relied on Firefox – an Internet browser commonly used in Windows, Macintosh and Linux. Therefore, the lack of experience using a Macintosh computer did not constitute a problem for the participants.

Digital technologies knowledge

In the Knowledge section, we asked participants to qualify their digital technologies knowledge (Figure 6.11). In the Setup and Configure question, the majority (66.67%) considered themselves *Average*. When asked about their familiarity with programming using menus, the method of creating computer programs through a visual interface that presents various options and choices, half of the participants considered themselves *Very poor* (25%) or *Below average* (25%). Only a tiny portion of the users (16.67%) considered themselves *Above average*, and none selected the *Excellent* option. The question of Coding (writing code) showed even more skewed results. A significant majority (58.33%) consider themselves *Very poor*, and one participant (8.33%) considers themselves *Below average*. Several participants consider themselves *Average* (25%), and again, only one (8.33%) considers themselves *Above average*. Regarding Following instructions, only one participant considered themselves *Below average* (8.33%). Half of the participants thought they were *Average* (50%), while a significant number thought they were *Above average* (41.67%).

Satisfaction with digital technologies

We asked participants to identify their level of satisfaction with digital technologies in the following question (Figure 6.12). Participants indicated a high level of satisfaction with Mobile phones: the majority indicated *Satisfied* (58.33%), and a significant number said they were *Very satisfied* (41.67%). Regarding Tablets, almost a third stated *Neither* (27.27%), a considerable number said *Satisfied* (45.45%), and another group said *Very satisfied* (27.27%). In terms of Computers, the majority said *Satisfied* (58.33%), a small number indicated *Neither* (8.33%), and a third stated *Very satisfied* (33.33%). With Car digital tools (e.g., Bluetooth stereo, maps, etc.), participants gave more negative responses: a number of them indicated being *Dissatisfied* (30%), another group said *Neither* (30%), and a third group stated *Satisfied* (30%), while a small number indicated *Very satisfied* (10%). Regarding Home appliances, the majority stated *Satisfied* (58.33%), a small number indicated *Neither* (16.67%) or *Very satisfied* (16.67%) and even a smaller number said *Dissatisfied* (8.33%). With Other technologies (e.g., ATM, online banking, etc.), a quarter of the participants said *Neither* (25%), and a good number of them indicated being *Satisfied* (50%). The other quarter said being *Very satisfied* (25%).

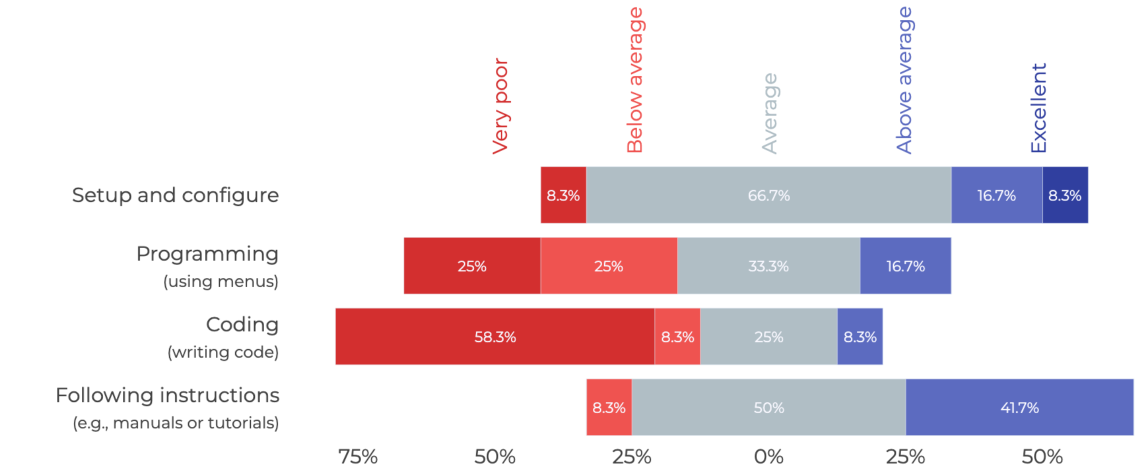


FIGURE 6.11: Knowledge. How would you qualify your digital technologies knowledge?

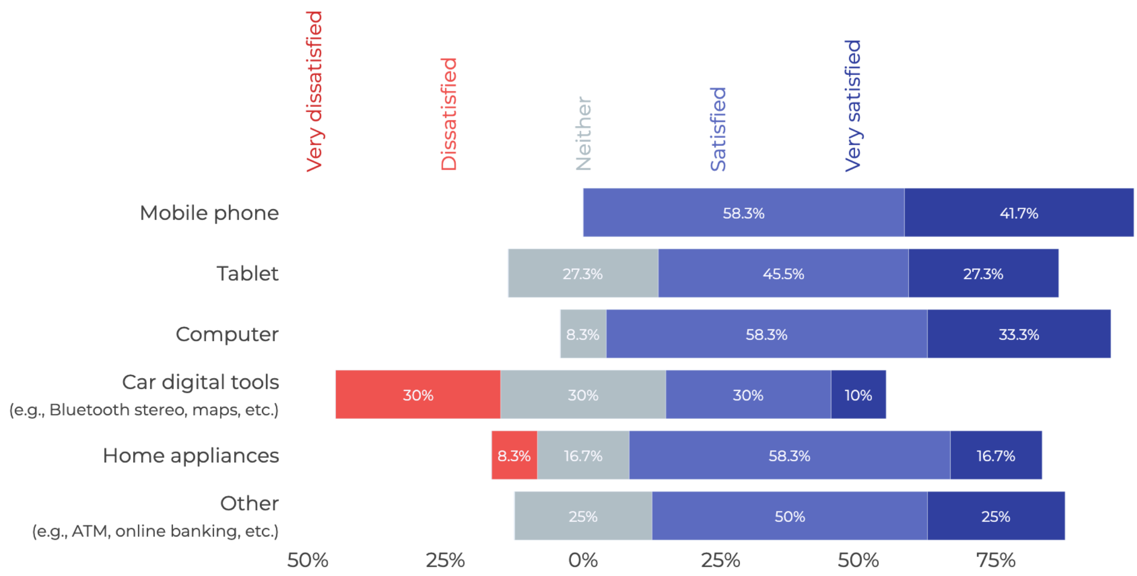


FIGURE 6.12: Experience. How would you consider your level of satisfaction with digital technologies?

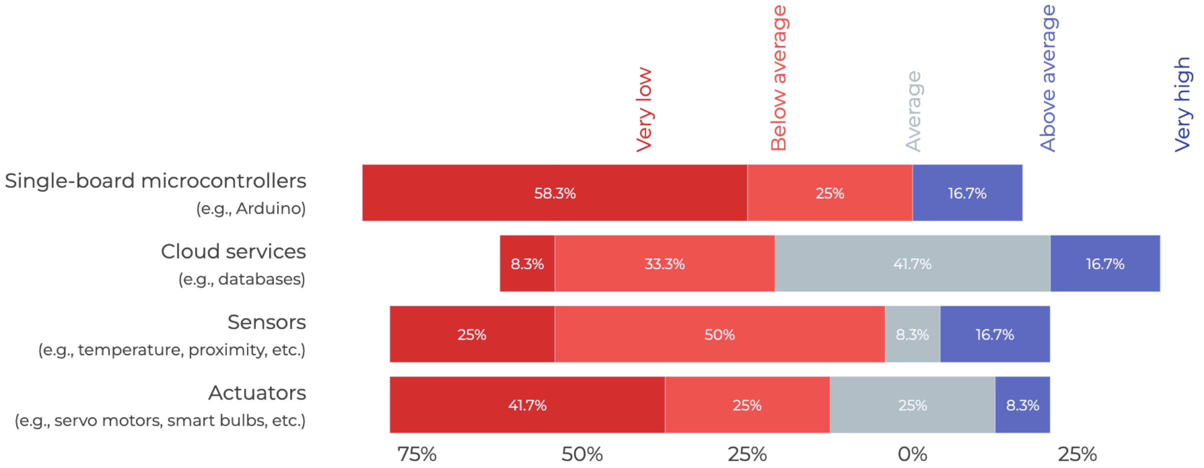


FIGURE 6.13: Internet of Things (IoT). How would you rate your knowledge of IoT technologies?

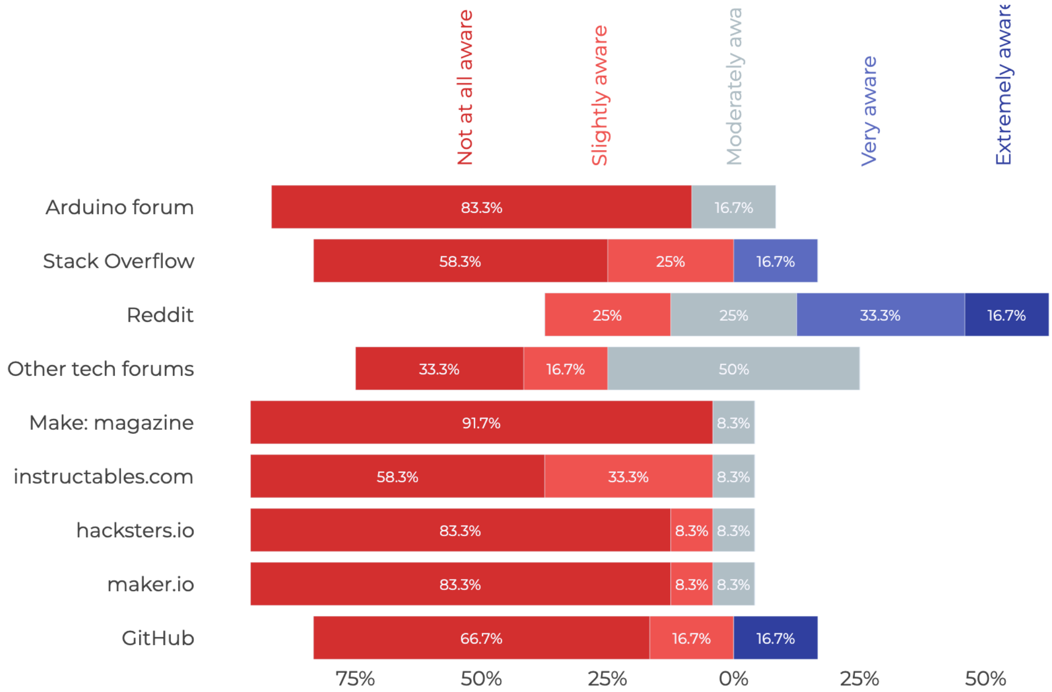


FIGURE 6.14: Community. How would you rate your awareness of the following community-based resources?

Knowledge of IoT technologies

The survey question on IoT asked participants to rate their knowledge of various technologies (Figure 6.13). We opened with Single-board microcontrollers (e.g., Arduino). A high number of participants indicated their knowledge as *Very low* (58.33%), a quarter of them said *Below average* (25%), and a third group indicated *Above average* (16.67%). The next area asked about Cloud services (e.g., databases). A small number of participants said *Very low* (8.33%), a third indicated *Below average* (33.33%), most participants said *Average* (41.67%) and another group considered *Above average* (16.67%). Pointing to their knowledge about Sensors (e.g., temperature, proximity, etc.), a quarter of the participants said *Very low* (25%), half of them said *Below average* (50%), a small number indicated *Average* (8.33%). In contrast, another group said *Above average* (16.67%). The final part of this question asked them to consider their knowledge of *Actuators* (e.g., servo motors, smart bulbs, etc.). A large number of participants said *Very low* (41.67%), a quarter indicated *Below average* (25%), another quarter said *Average* (25%), and a final group selected *Above average* (8.33%).

Awareness of community-based resources

The following question asked participants about their awareness of community-based resources (Figure 6.14). The first option was the Arduino forum. A majority of the participants in the study indicated *Not at all aware* (83.33%), and a minority said *Moderately aware* (16.67%). The next option was Stack Overflow, where more than half indicated *Not at all aware* (58.33%), a quarter *Slightly aware* (25%), and a minority *Very aware* (16.67%). For Reddit, a quarter indicated *Slightly aware* (25%), another quarter *Moderately aware* (25%), a third *Very aware* (33.33%), and the remaining *Extremely aware* (16.67%). For Other tech forums, a third indicated *Not at all aware* (33.33%), a small percentage said *Slightly aware* (16.67%), while half of the participants noted *Moderately aware* (50%). For Make: magazine, most participants marked *Not at all aware* (91.67%), and a small percentage said *Moderately aware* (8.33%). In the case of Instructables.com, the majority said *Not at all aware* (58.33%), a third noted *Slightly aware* (33.33%), and a small percentage said *Moderately aware* (8.33%). The results were the same for Hacksters.io and maker.io: a majority indicated *Not at all aware* (83.33%), a small number said *Slightly aware* (8.33%), while another small percentage indicated *Moderately aware* (8.33%). Finally, for GitHub, the majority noted *Not at all aware* (66.67%), a small number indicated *Slightly aware* (16.67%), while another percentage noted *Extremely aware* (16.67%).

Asking help online

The survey's final question asked participants how frequently they asked for help online (Figure 6.15). The first option asked them about posting a question in a forum. A small percentage said *Never* (16.67%), the majority said *Rarely* (58.33%), and a quarter noted *Sometimes* (25%). The second option was about chatting with experienced people. Several participants said *Never* (16.67%), a smaller group indicated *Rarely* (8.33%), most participants said *Sometimes* (66.67%), and a small number indicated *Often* (8.33%). The final option was for participants to ask on social media (e.g., Twitter, Facebook). Several indicated *Never* (16.67%), others *Rarely* (16.67%), a considerable number said *Sometimes* (41.67%), another percentage said *Often* (16.67%), and a smaller group said *Always* (8.33%).

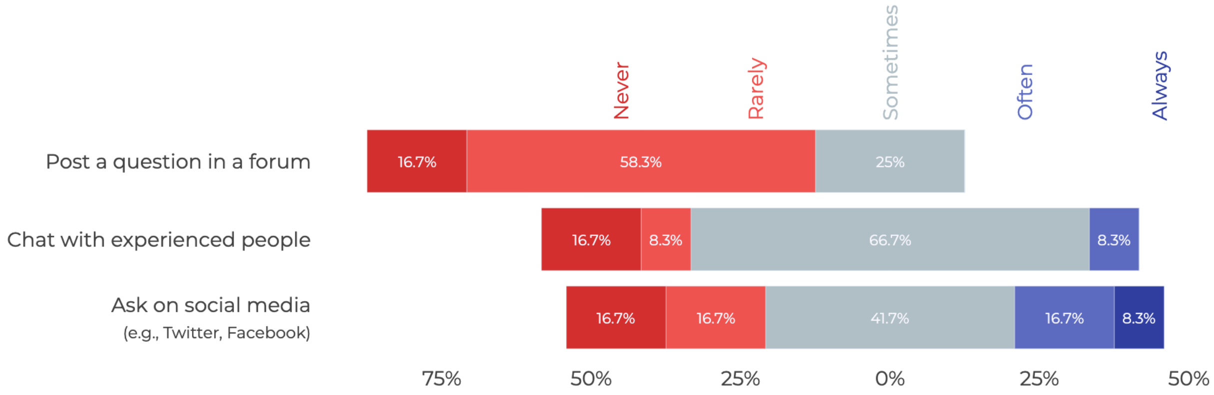


FIGURE 6.15: Help and assistance. How frequently do you ask for help online? (beyond "asking Google")

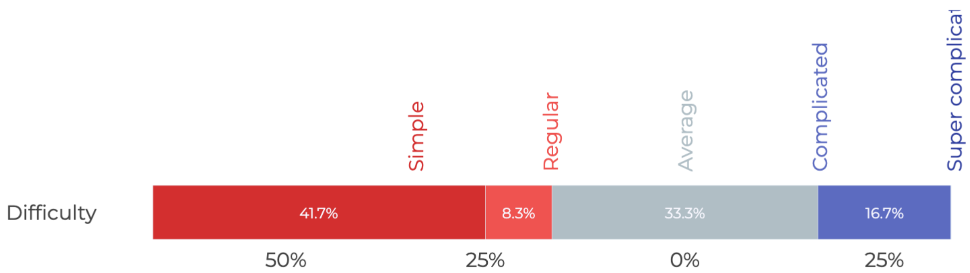


FIGURE 6.16: Participants' evaluation of the project in terms of difficulty.

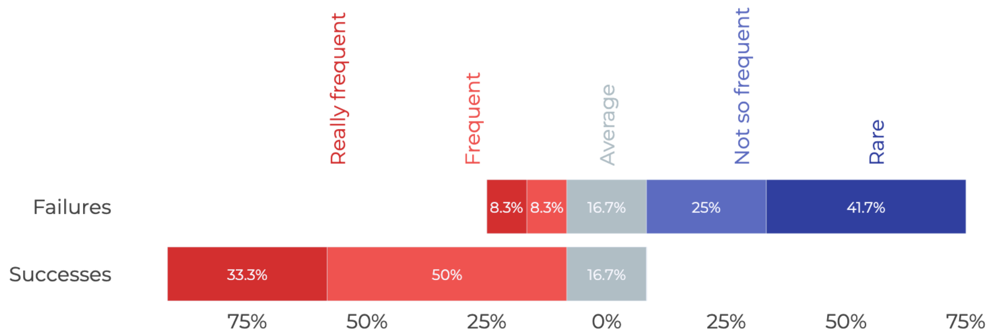


FIGURE 6.17: Participants evaluation of failures and successes frequency.

6.3.2 | Prototype Building Observations

As described in Subsection 6.2.2, the second part of the user study was dedicated to building the prototype for the proposed project. We presented all the required gear after disclosing the scenario and introducing the Standing Desk Measuring Project scenario to participants. We showed how the cards connected to a web resource using the QR code. We stayed in the room while participants followed the instructions and assembled the prototype, responding to questions when they arose and observing how they were doing it. The following paragraphs present a series of observations we made while participants built the prototypes.

Hardware vs. Software

For the participants, the hardware section was straightforward. It took them an average of 12 minutes to assemble the circuit (the minimum was 9 minutes, while the maximum was 19 minutes). The difference in time can relate to the speed of reading the instructions. Some participants showed a can-do approach, moving forward faster, while others were more concerned about following the instructions precisely. As the built-in LED on the Arduino microcontroller lit, several participants (41%) expressed joy when connecting the circuit to power. The feedback gave them confidence, demonstrating that they could adequately assemble the hardware part of the project.

Interestingly, for more than one participant, it was disorienting how to plug the Micro USB (USB-B) connector properly. All participants were familiar with USB, a standard that has been around for decades. Nevertheless, there was a moment of confusion for some of them. The Micro USB connector does not indicate how it should go, and operating with exposed electronics did not help.

The software part of the experiment presented more complications than putting together the hardware. This section comprises twelve distinct steps compared to the hardware section, which has only four steps. Although we can categorise the steps into three distinct areas – a) configuring the cloud service; b) programming the microcontroller; c) creating a dashboard – they were presented to participants as one section.

New vocabulary

The first portion of the software part is related to familiarising participants with the Arduino IoT cloud. Once they had read a few instructions, we invited participants to go to the Arduino IoT Cloud and start the process by *creating a thing*. Unlike the hardware section, this part of the experiment incorporated a significant new and specific vocabulary for the participants. For example, in the experiment context, a Thing – spelt with a capital T – is a connected device that can communicate with the cloud and interact with other Things or anything else in the physical world.

Participants did not verbally express how incorporating these new concepts added a significant cognitive load to building the prototype. Still, it is noticeable in the recordings that the operation slowed down as they entered the core of configuring the cloud and creating the software for the microcontroller.

As the process became more complex, participants encountered situations where it was difficult

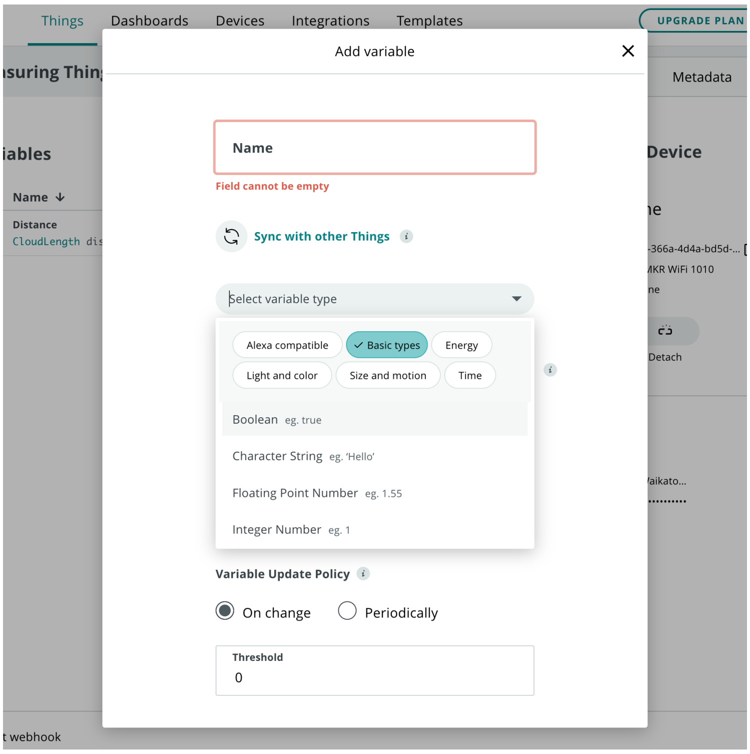


FIGURE 6.18: Arduino IoT Cloud. Add a variable, selecting a variable type.

to identify what they needed to do. Although the Arduino IoT Cloud interface is straightforward for somebody familiar with the concepts, it is puzzling for end-user developers, as they are non-experts and incorporate vocabulary and knowledge while building the project.

Arduino IoT Cloud problems

When configuring a variable to store the data, most participants found the process difficult as the instructions asked them to set up a variable of the type Length – that stores an integer value – but the pull-down menu presents a filtered list by default (Figure 6.18). Participants could not identify that they had to uncheck the button labelled *Basic Types* to access the complete list of possible variable types. Although we have not tested it, we believe they could remember that this needs to be done the second time they have to do it. However, it indeed presented a moment of confusion for all the participants. There was a discrepancy between the instructions and what they saw in the interface. As the creators of the instructional material, we could have created the instructions specifying that. Still, as we do not control the default behaviour of the Arduino IoT Cloud, there could be other discrepancies. What presents as a minor discrepancy helps prove that having diverse possibilities created and controlled by different parties increases the difficulties for end-user developers.

After creating the required variable, participants must prepare the microcontroller to be IoT-ready in the next step. It requires selecting the device from the Arduino IoT Cloud interface. That part of the process did not always work. At times the microcontroller was not immediately recognised, and participants were significantly puzzled, as they could not understand what to do if their system did not behave as the instructions indicated.

TABLE 6.2: Themes and codes.

Themes	Codes
Knowledge or Experience	Big Questions, Learning Curve, Mixed Feelings, Phases, Transferable Knowledge
Negative Self-Perception	Doubts and Confusion, Failures, Fear of Breaking Things, Hands in the Air, Negative Comments, Negative Experiences, Stress and Anxiety
Positive Self-Perception	Curiosity, Done, Enthusiasm, In Awe, Intrinsic Motivation, Positive Comments, Positive Experiences, Satisfaction, Self Perception, Self-Efficacy, Success, Uses
Process and Approach	Advising Others, Learning and Process, Modularity, Process, Proposed Solutions, Suggestions, Understanding, Application, Learning Curve, Videos
Technical Challenges	Complicated, Computer Crash, Hardware, Problems, Technical Difficulties, Technical Support, Technology Frustration, Technology Literacy, Technology Perception, Software

From a technical point of view, the Arduino IoT Cloud uploads software updates after recognising the device. At the time of the experiment, Arduino released a WiFiNINA firmware update. WiFiNINA is a library for Arduino boards that supports using WiFi connectivity through the NINA (New Integrated Network Architecture) family of WiFi modules. A fascinating moment verified our understanding that IoT technologies could be daunting for end-user developers. A participant ran the study in the morning and worked without inconvenience. The next participant, in the afternoon of the same day, got stuck at this part of the process, and the Arduino IoT Cloud could not finish the microcontroller configuration. We considered the session a failure. Later in the day, we could identify the issue through different tests. The microcontroller had the library WiFiNINA 1.4.5. Arduino released a new library version in their IoT Cloud between the morning and afternoon sessions. The computer used for the experiment – an iMac with OS X El Capitan (version 10.11.6) and Firefox 78.12.0esr (64-bit) – could not update the firmware. Connecting the microcontroller to a newer computer – an iMac with MacOS Mojave (version 10.14.6), Firefox 90.0.2 – could update it. Once the firmware update is in place, the older computer can run the experiment without problems. For an end-user trying to make this prototype at home – perhaps having only one computer – facing this issue would have derailed the process. Furthermore, what is worst, they could not have identified what was causing it.

Final notes

All but one of the participants could complete the work and have a functioning prototype at the study's end. The participant that could not finish the project was confronted with the problem with the WiFiNINA library. That issue stopped the participant from completing the study.

6.3.3 | Transcribed Recordings: Qualitative Analysis

The user study video recording started after participants finished the survey, once they consented to be recorded. The 14 hours and 27 minutes of video recordings include participants' time to assemble the prototype and the follow-up semi-structured interview. We present here a qualitative evaluation of the recordings' transcriptions, including the time when participants put together the project and the semi-structured interviews used to finalise the study.

To analyse the recorded data, we employed the thematic analysis method, a qualitative research

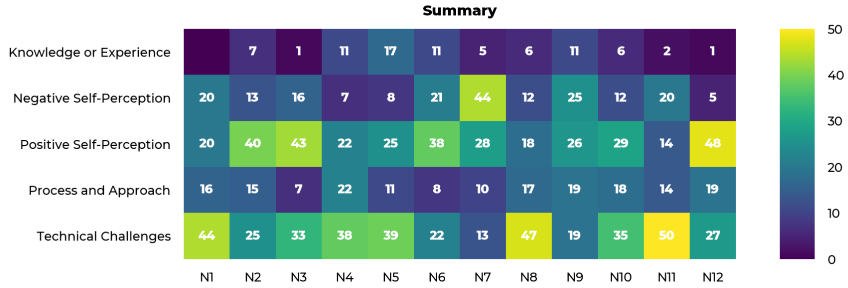


FIGURE 6.19: Heat map showing the distribution of participant quotations by theme.

methodology to identify and analyse patterns, themes, and meanings within data. The process included the following steps:

- Transcribing the recordings,
- Coding the data,
- Identifying themes and patterns, and
- Getting insights.

The video recordings were transcribed to text using AssemblyAI[5]. As a result, the final text consisted of 63,881 words, including all the dialogues between participants and the researcher.

The data were manually coded, assisted by ATLAS.ti[6], a software package used for qualitative data analysis. The process of coding the data was dynamic, starting with twelve pre-created codes and allowing the creation of more codes, as the text suggested. The final list included 45 different codes and a total of 812 unique quotations.

After coding all the data, we grouped the codes into five thematic areas (Table 6.2):

- Knowledge or Experience,
- Negative Self-Perception,
- Positive Self-Perception,
- Process and Approach, and
- Technical Challenges.

We created two heat maps to visualize the distribution of quotations across different codes. This approach allowed us to ensure codes were distributed among participants in a well-balanced manner. By plotting the percentage of quotations on the respective code categories, we assessed the coverage and identified potential errors in the coding process.

[5] https://www.assemblyai.com/

[6] https://atlasti.com/

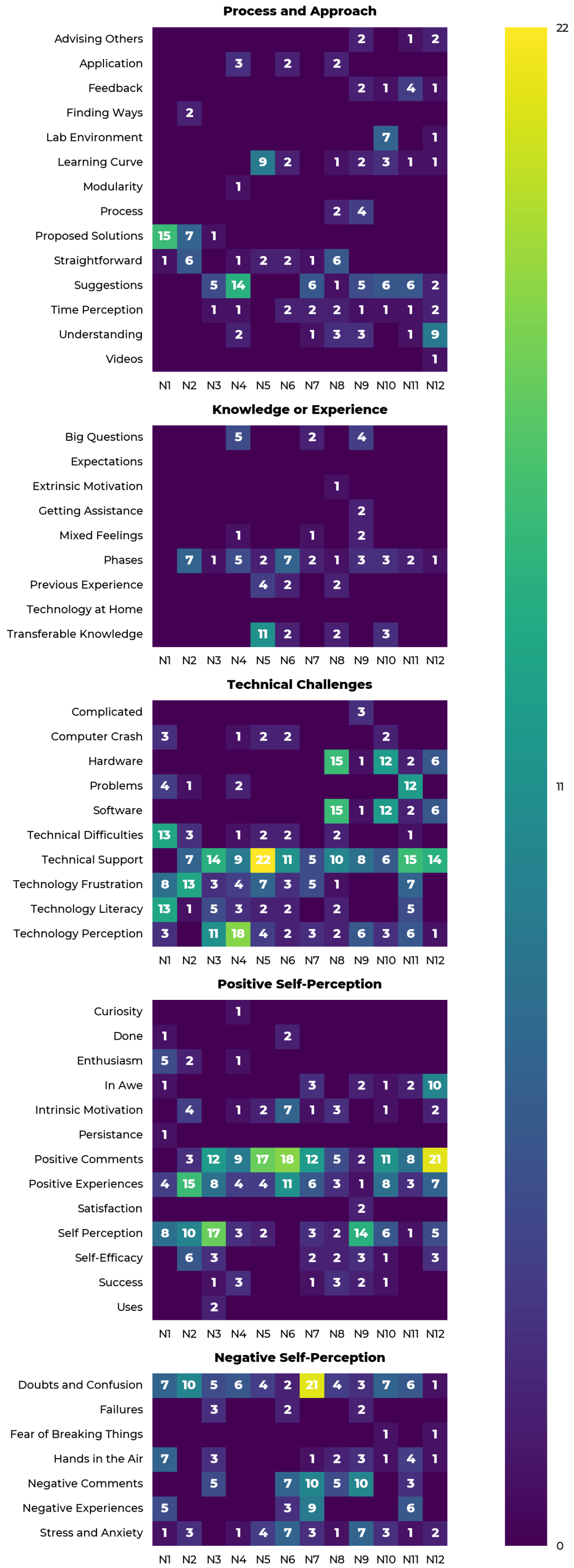


FIGURE 6.20: Heat map showing the distribution of participant quotations by code.

Figure 6.19 presents the distribution of participant quotations corresponding to the five defined themes[7]. Rows correspond to the themes, while columns correspond to participants. Each square shows the percentage of quotations from a participant coded in a particular theme. Therefore, all the columns sum to 100%. Figure 6.20 shows quotations by participants corresponding to the 45 different codes. We can see some concentration areas (e.g., *Technical Challenges* present more quotations than *Knowledge or Experience*). Still, there are no visible skews that could mean the coding process was biased.

This chapter's user study aims to determine if an end-user developer with the relevant support can successfully build a bespoke IoT project. The subsequent sections present several key findings resulting from the user study.

Feasibility of non-experts assembling an IoT prototype

In the controlled conditions of our study, which was conducted in a lab and provided participants with specific hardware, software, and instructions, we found that nearly all participants (11 out of 12) were able to successfully assemble the prototype and achieve a functional project by the conclusion of the experiment. For the participant (N1) who could not complete the project, the problem was a library update not transferring to the microcontroller. The complexity of the problem was too high for the participant to figure out and eventually solve.

Several participants expressed that the process was straightforward. According to one participant (N2), the instructions were *"relatively easy to follow in spite of the jargon, which is always a problem with instructions, because instructions are always written by people who know what they're doing."* Meanwhile, another participant (N5) expressed surprise at how easy the process was, saying, *"I'm surprised at how easy it was."* and a third one (N7) showed a change in perspective by stating *"Did you see when you first opened the box it's a lot simpler than it first appears because from somebody like me who doesn't usually see the inner workings of product, that kind of looks intimidating and I'm thinking I am not going to be able to figure out anything here, but actually it's very straightforward."*

Exploring breaking points in assembling the prototype

Beyond being able to complete the task, the process was not necessarily smooth for all the participants. Through the process, we identified different levels of complexity that led to confusion or frustration. Participants expressed it in different ways, and coding the recordings with the *'Doubts and Confusion'* code captured a significant part of what they expressed – the extreme was N7, where 21% of the quotations coming from their participation were coded with this label (Figure 6.20).

Our objective was to identify breaking points where the confusion or the stress was so high that participants would be willing to drop the idea of finishing the process. A considerable number of participants (8 out of 12) expressed that there were breaking points in the process that would have derailed them if they were not in the controlled environment of the lab and with the researcher supporting them. One of them (3) noted *"I would have given up and probably never would have got*

[7] In this heat map and all subsequent ones, zero values have been intentionally omitted to minimise visual clutter and enhance readability.

around to sitting down and working through that.”, and another (7) expressed *“I mean, if you hadn’t been here, if I’d been at home, I would have been stuck at that point not knowing how to actually fix that problem.”* Another participant (11) articulated well how the process could feel by saying *“So, I think that I would try for a little while. I think if I was to put it into a timeline, I’d give it like an hour, max, and then I probably throw in the towel and be a bit frustrated if it took that long, I would think this is easy enough. That’s probably the most frustrating bit of any use of technology. It doesn’t go as it says, and you can’t figure out what’s wrong, and some of the troubleshooting you do doesn’t work.”* What we can see through their comments is that even if they have the will to create a bespoke IoT project, they would not have the eagerness to do it at any cost.

Through the user study, we have observed that when technology fails to respond in a way they can understand, many participants will not be willing to pay the cost of getting it working.

Hardware and software

In the study, participants articulated a clear distinction in easiness between the hardware and software phases. Participant N2 expressed clearly that *“The software was the most stressful, actually.”*, while participant N8 noted that *“All the software requires a lot more prior knowledge.”* to further explain *“But learning to use the software side with assembling the dashboard and putting all the variables and stuff, there was a lot of information in a very short period of time.”* Contrasting to that, participant N4 noted that *“The hardware part was absolutely straightforward.”* and participant N12 said *“To be honest, I think the hardware was a bit easier Because it’s nice to have something tangible that you can play with.”*

We have observed that the impact of the number of steps and the balance between relying on previous knowledge versus acquiring new knowledge is significant. The hardware part of the project provided to study participants comprises only a few components that easily click together, requiring straightforward assembly steps. However, the project’s software side involves a more extensive set of instructions, and participants face a higher cognitive load when adapting these instructions to an unfamiliar context through a novel interface. Indeed, the Table of Contents of the instructions shows that the Hardware section has only four pages while the Software section has twelve.

While this insight may not universally apply to all scenarios involving the interplay of hardware and software, it is essential to consider when supporting non-expert developers in creating their IoT systems. Specifically, when assisting them in software development, offering greater clarity and guidance becomes crucial to ensure a smoother and more successful creation process.

Transferring knowledge

During the user study, participants gained specific knowledge while assembling a prototype and using IoT technology for the first time. As described in Subsection 6.2.2, they engaged in a hands-on experience to create a customized IoT project. Though the study’s primary objective was not educational, we anticipated that participants would acquire certain knowledge during this process. This included understanding the components involved, how they interacted with each other, and gaining insights into the distinctions between hardware and software and their interplay.

It is important to note that the extent of knowledge transfer varied among participants, depending on their prior experiences. For example, participants with prior coding experience might have different learning levels than those without coding experience. To assess their progress, we inquired about their learning process and whether they could apply the knowledge gained in future experiences.

Throughout the study, participants' responses revealed their awareness of their learning experiences during the process. Participant N5 expressed that they did not feel like they learned much in a short period, leading to some difficulties. They recognised the difference between copying and pasting the code instead of fully understanding it but believed that typing it out manually in an educational setting would reinforce their learning. They likened their experience to learning basic phrases in a new language but not being equipped to form complete sentences. Participant N6 shared their perspective, saying, *"So, this gives me some faith that I can do a simple task. I think my problem is I'm wanting to run before I can walk."* They also added, *"I think it was very easy to very quickly follow the instructions, but understanding is another level of the learning curve. To follow the steps was shallow, easy."* Participant N8 described the relationship between learning and time, stating, *"But learning to use the software side with assembling the dashboard and putting all the variables and stuff, there was a lot of information in a very short period of time."* Participant N10 noted, *"I don't know that it's so much learning, but kind of like connecting thoughts a little bit."*

Regarding the ability to transfer the knowledge gained to different projects, participants expressed mixed feelings despite acknowledging their improved situation after the study. Participant N9 highlighted their progress, saying, *"I think because it's a big leap for me because I had zero knowledge of how it works, but now I think I do have a backbone of the how system works the next thing will be a lot easier."* On the other hand, Participant N6 raised a concern, stating, *"I guess if you gave me, I don't know, a different board or something and said, now connect that up, then perhaps without the instructions, I would may not be able to transfer that knowledge"*, exemplifying that completing the task with instructions is distinct from transferable knowledge and does not guarantee the procedural capacity to do it in a different context without support. Participant N11 pointed out that the study did not provide a deep understanding of how things work, mentioning, *"You don't acquire a lot of knowledge about the understanding of how it works, like how the code makes it work, how the boards connect and how they all work."* Participant N12 recognized the study's impact, explaining, *"So, I think what it's done for me, I don't have enough technical knowledge to apply this immediately to something else, but it gives me some background understanding of what is involved. And I think it gives me a few key search terms that I can do."* Overall, the participants recognized their progress in grasping the concepts, but some expressed reservations about their ability to transfer that knowledge to entirely different scenarios.

In summary, the user study provided valuable insights into participants' learning experiences with IoT technology. While some felt better prepared for future projects, concerns were raised about transferring knowledge to different scenarios. The study offered a foundational understanding, but depth in code functionality and board interactions varied among participants. It serves as a stepping stone for further exploration, identifying key search terms for future learning endeavours and emphasizing the need for balanced hands-on experiences and instructional guidance.

6.4 | Discussion

This chapter presented a user study exploring the support an end-user developer can use to surpass obstacles in *building electronic circuits* and *programming computers* to have a functional prototype of a bespoke IoT project. The study aims to accomplish several key objectives:

- **Testing assembly capability**
  One of the central goals of this study is to ascertain whether end-user developers, when presented with project elements and corresponding instructions, can successfully assemble the project. To this end, the process involves starting with the TILES IoT Toolkit cards. These cards are enhanced by making them more tangible by including concrete elements, such as photos. Additionally, they are connected to instructions through the use of QR codes. This approach creates a bridge between physical components and digital guidance.

- **Evaluating support suitability**
  The study also seeks to assess the adequacy of the provided support. This is achieved by showing participants the project scenario, the required equipment, and the card linked to a web resource. Participants are guided through a step-by-step instructional process resembling a wizard-based approach. This step aims to determine if the support mechanism effectively assists users in successfully assembling the prototype.

- **Transferring knowledge**
  Another critical objective is to examine how much knowledge gained from this exercise can be transferred to another project under different conditions. This part of the process involves evaluating whether the skills and understanding acquired during the initial project assembly can be effectively utilized when faced with a different context and requirements.

In summary, this comprehensive process involves integrating tangible elements, digital instructions, and step-by-step guidance to test users' ability to assemble a project, evaluate the suitability of the support provided, and gauge the transferability of acquired knowledge to different contexts, with additional insights gained from follow-up interviews.

6.4.1 | Insights Gained from the Study

We found that participants could assemble the prototype following the presented instructions (11/12 participants completed the project). We have also found that several strict conditions are required for their success:

- hardware and software behave exactly as expected,
- tutorials follow precisely what is shown in the needed interfaces, and
- everything works as expected, with no unexpected hiccups.

The study indicates that participants find the process enjoyable when everything goes smoothly. However, they become quickly frustrated when faced with obstacles. Constructing a personalized

IoT project might involve using unstable technology, resulting in frustration and even potential abandonment of the project by end-user developers. When encountering situations they struggle to comprehend fully, participants expressed a preference for reaching out to someone they trust rather than seeking help through public online forums. This hesitance to engage with online communities highlights the desire for instant access to an expert, which ultimately restricts or even renders the creation of personalized IoT projects impractical for end users.

The study's results corroborate that it is possible to bridge the gap between an idea with a high level of abstraction and a functioning prototype, although the process is quite specific. Participants have not demonstrated they could build bespoke projects without a tailored tutorial. Given the diversity of options for creating bespoke IoT projects, it is highly improbable to find specific tutorials for all the possible combinations of hardware and software.

Previous research has focused on supporting end-user developers in defining and ideating an IoT project. Our study extends into the realisation of a project and demonstrates that end-user developers can accomplish it, although they need to be strongly supported.

Using an in-the-wild approach, asking participants to develop the project at home with their technology could have provided better insights. Future research could also test end-user developers' ability to combine different tutorials (e.g., hardware and software).

6.5 | Summary

This chapter's user study aims to evaluate a method of aiding end-user developers in crafting personalised IoT projects, specifically focusing on the challenges of *building electronic circuits* and *programming computers*. The central hypothesis of this thesis is that, through adequate support, it is indeed feasible for end-user developers to create bespoke IoT projects successfully. The study's findings confirm the feasibility of end-user developers successfully creating bespoke IoT projects with the right kind of support. However, it highlights the complexity involved and emphasises the need for targeted assistance, such as step-by-step guidance and real-time troubleshooting tools. The results suggest potential directions for further exploration, such as improving real-time support mechanisms or developing more intuitive troubleshooting tools.

It has become evident that end-user developers face difficulties when encountering unexpected technical problems, such as troubleshooting faulty circuits or debugging code. These challenges underscore the importance of providing adequate guidance when things do not proceed as planned. The subsequent chapter introduces a user study that addresses this issue by evaluating a method of assisting end-user developers in diagnosing and rectifying errors when faced with unexpected setbacks.

Chapter 7

Guided Troubleshooting

User Study on IoT Problem-Solving Protocol

> *Argh! It doesn't work!"*
> *I had been working on a circuit for two days.*
> *And I had just soldered it.*
> *Now, I was programming it.*
> *But suddenly, it didn't work anymore.*
> *And didn't understand why.*
>
> **— Oyvind Dahl**
> *Build Electronic Circuits Newsletter* [41]

IN THE PRECEDING CHAPTERS, we examined the experience of end-user developers in IoT project development and explored strategies to support their ideation and prototype realisation. Following the comprehensive long-term study presented in Chapter 4, which helped identify essential skills such as electronic circuit construction and programming for IoT project creation, Chapter 5 introduced a user study oriented to test how a deck of conceptual cards enables end-user developers to envision better and map ideas for their projects. Subsequently, in Chapter 6, we expanded the TILES IoT Toolkit to assist end-user developers in turning abstract ideas into functional prototypes. Our observations revealed that end users derive enjoyment when their projects progress smoothly but quickly become frustrated when faced with obstacles. Notably, we recognised that the technology can be unstable, often leading to malfunctions that develop unexpectedly over time. Such instances leave end-user developers perplexed and frustrated.

Our fourth research question, RQ4, inquires into the specific forms of support necessary for these developers:

> RQ4 | What forms of support do end-user developers require to efficiently identify and resolve issues in their IoT projects?

Exploring the landscape of support mechanisms available for IoT troubleshooting reveals a critical challenge: resources, though plentiful, are often disorganised and dispersed across the Web. This disarray significantly obstructs end-user developers from accessing and effectively using the support they need.

Cognitive Load Theory, as introduced by Sweller et al. [165], posits that structured information presentation enhances understanding and retention and is essential in complex troubleshooting scenarios, which are commonplace in IoT projects. Addressing this crucial gap, our study assesses the impact of a troubleshooting guide tailored to assist end-user developers in identifying and resolving issues. We hypothesise that providing sequential guidance for isolating problems will significantly improve developers' ability to troubleshoot effectively.

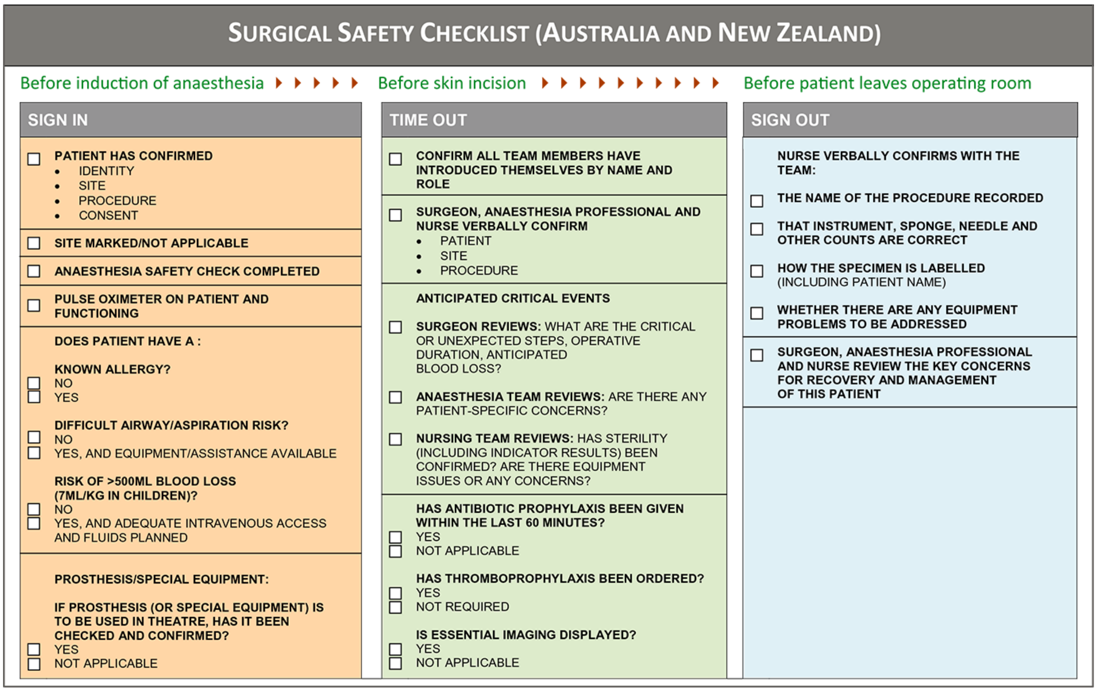
We hypothesise that a carefully crafted guide, providing sequential steps to isolate problems, will enhance developers' ability to identify and resolve issues, thereby streamlining the development process. This hypothesis rests on the idea that structured methods can demystify troubleshooting, making it more accessible for end-user developers. By linking our study's goals to the fragmented current support systems, we illustrate how structured methods can improve the coherence and usability of IoT troubleshooting. Although our findings support that such protocols can drive meaningful improvements, they do so within specific limitations. This framing does not detract from the value of our results but instead sets a practical scope for applying and exploring guided processes in IoT development.

We conducted a user study to test our hypothesis that a problem-solving method helps end-user developers isolate and address IoT project issues. The study aims to validate guided strategies using tailored instructional materials like step-by-step cards and website guidelines to provide accessible troubleshooting support. These tools are designed to reduce cognitive load and improve problem-solving efficiency. The following sections detail the study's methodology, outcomes, and analysis, providing evidence for the value of structured problem-solving approaches in enhancing end-user development in IoT.

After these introductory paragraphs, this chapter unfolds as follows: Section 7.1 presents the user study design and Section 7.2 describes the method for executing it. Section 7.3 presents our methodology for data analysis and Section 7.4 introduces a comprehensive analysis of the collected data. Looking at these insights, Section 7.5 discusses our findings in identifying issues in IoT projects and the associated challenges. Culminating our exploration, Section 7.6 recapitulates the key takeaways from this chapter, providing a concise summary of our investigation.

7.1 | Designing the User Study

This section details our approach to a user study to evaluate support tools for non-experts troubleshooting bespoke Internet of Things (IoT) systems. Our goal is to assess the effectiveness of a structured checklist protocol, akin to those used in medical settings, in enhancing non-experts' ability to identify and resolve IoT system issues.


SURGICAL SAFETY CHECKLIST (AUSTRALIA AND NEW ZEALAND)

Before induction of anaesthesia ▸▸▸▸▸ Before skin incision ▸▸▸▸▸▸▸▸▸▸ Before patient leaves operating room

SIGN IN

☐ PATIENT HAS CONFIRMED
- IDENTITY
- SITE
- PROCEDURE
- CONSENT

☐ SITE MARKED/NOT APPLICABLE

☐ ANAESTHESIA SAFETY CHECK COMPLETED

☐ PULSE OXIMETER ON PATIENT AND FUNCTIONING

DOES PATIENT HAVE A :

KNOWN ALLERGY?
☐ NO
☐ YES

DIFFICULT AIRWAY/ASPIRATION RISK?
☐ NO
☐ YES, AND EQUIPMENT/ASSISTANCE AVAILABLE

RISK OF >500ML BLOOD LOSS (7ML/KG IN CHILDREN)?
☐ NO
☐ YES, AND ADEQUATE INTRAVENOUS ACCESS AND FLUIDS PLANNED

PROSTHESIS/SPECIAL EQUIPMENT:

IF PROSTHESIS (OR SPECIAL EQUIPMENT) IS TO BE USED IN THEATRE, HAS IT BEEN CHECKED AND CONFIRMED?
☐ YES
☐ NOT APPLICABLE

TIME OUT

☐ CONFIRM ALL TEAM MEMBERS HAVE INTRODUCED THEMSELVES BY NAME AND ROLE

☐ SURGEON, ANAESTHESIA PROFESSIONAL AND NURSE VERBALLY CONFIRM
- PATIENT
- SITE
- PROCEDURE

ANTICIPATED CRITICAL EVENTS

☐ SURGEON REVIEWS: WHAT ARE THE CRITICAL OR UNEXPECTED STEPS, OPERATIVE DURATION, ANTICIPATED BLOOD LOSS?

☐ ANAESTHESIA TEAM REVIEWS: ARE THERE ANY PATIENT-SPECIFIC CONCERNS?

☐ NURSING TEAM REVIEWS: HAS STERILITY (INCLUDING INDICATOR RESULTS) BEEN CONFIRMED? ARE THERE EQUIPMENT ISSUES OR ANY CONCERNS?

HAS ANTIBIOTIC PROPHYLAXIS BEEN GIVEN WITHIN THE LAST 60 MINUTES?
☐ YES
☐ NOT APPLICABLE

HAS THROMBOPROPHYLAXIS BEEN ORDERED?
☐ YES
☐ NOT REQUIRED

IS ESSENTIAL IMAGING DISPLAYED?
☐ YES
☐ NOT APPLICABLE

SIGN OUT

NURSE VERBALLY CONFIRMS WITH THE TEAM:

☐ THE NAME OF THE PROCEDURE RECORDED

☐ THAT INSTRUMENT, SPONGE, NEEDLE AND OTHER COUNTS ARE CORRECT

☐ HOW THE SPECIMEN IS LABELLED (INCLUDING PATIENT NAME)

☐ WHETHER THERE ARE ANY EQUIPMENT PROBLEMS TO BE ADDRESSED

☐ SURGEON, ANAESTHESIA PROFESSIONAL AND NURSE REVIEW THE KEY CONCERNS FOR RECOVERY AND MANAGEMENT OF THIS PATIENT

FIGURE 7.1: The Australia and New Zealand edition of the Surgical Safety Checklist. Reproduced from *Royal Australasian College of Surgeons*, https://www.surgeons.org.

Drawing inspiration from the Surgical Safety Checklist used in healthcare (Figure 7.1), we have adapted this idea to suit the IoT context. In surgery, checklists ensure vital safety steps are followed. Similarly, our study uses a card-based protocol, guiding non-experts through troubleshooting. The analogy here emphasises the importance of structured, step-by-step approaches in complex tasks.

Our user study combines a card-based protocol inspired by medical safety procedures with QR code links to an instructional website. Recognising that end-user developers might lack the technical depth to follow these steps independently, we supplement the cards with an instructional website. This website provides detailed instructions and enables easy access to necessary code, facilitating a more efficient and accurate troubleshooting process. This method provides a systematic and structured approach for non-experts in troubleshooting bespoke IoT systems. We evaluate its effectiveness in helping participants pinpoint and address errors in IoT projects.

We have labelled the user study as *Break This Glass*, a concept inspired by physical objects like fire extinguisher cabinets and emergency alarm systems. In those contexts, *Break This Glass* involves breaking a glass cover to access emergency tools or trigger alarms. In our case, we use cards and instructional materials to provide end-user developers with a protocol for specific actions required in exceptional situations, such as system failures or any scenario where intervention is needed to rectify a system malfunction.

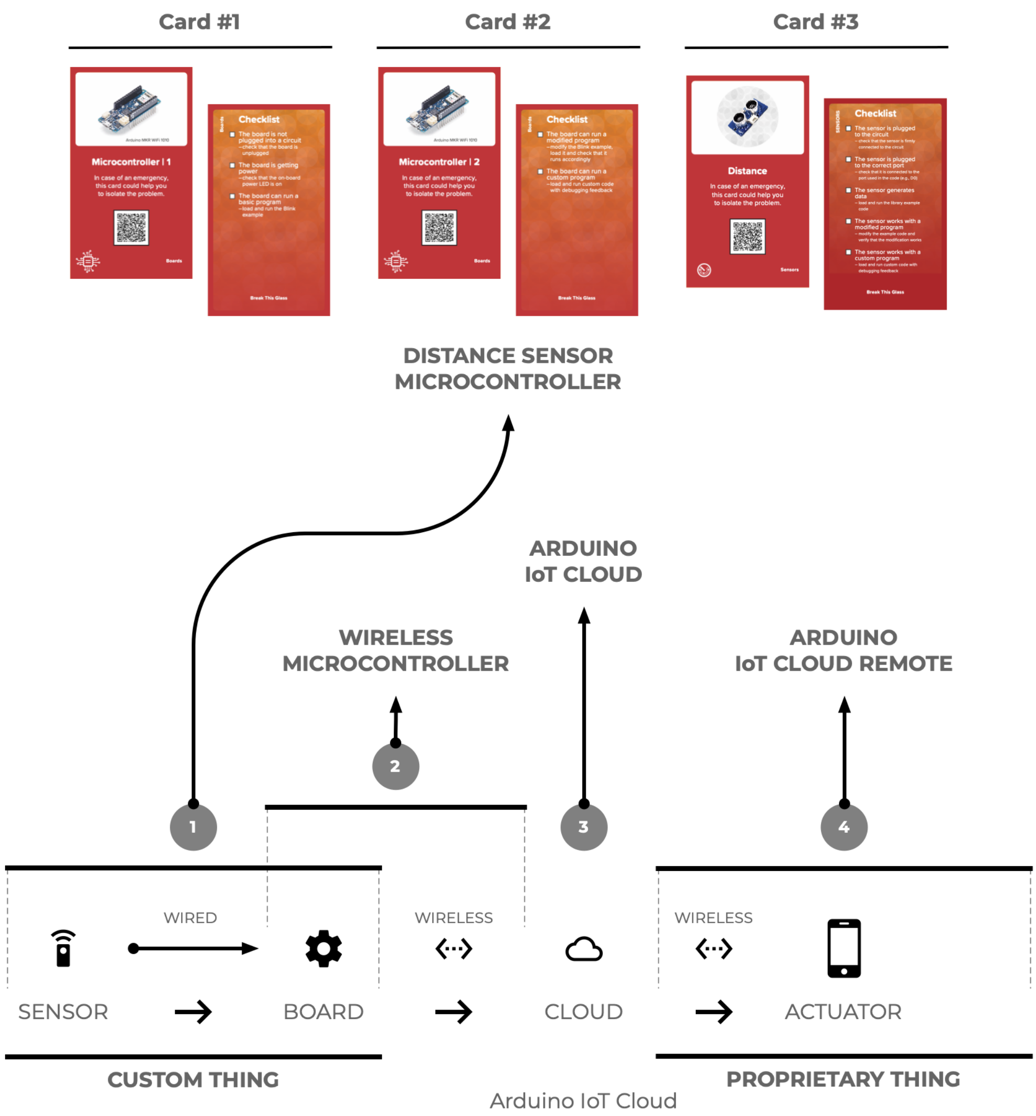


FIGURE 7.2: Project structure showing the three cards presented to participants.

FIGURE 7.3: Microcontroller cards used in the user study: Microcontroller 1 & 2.

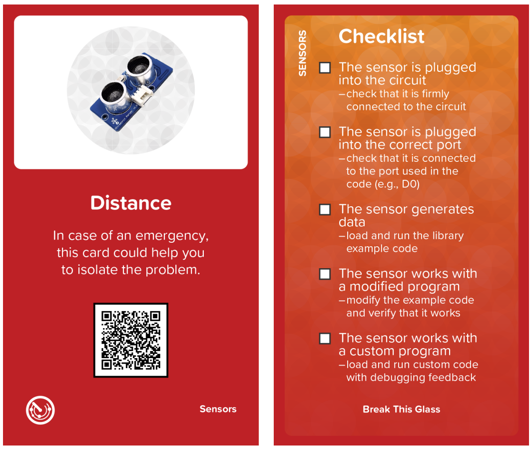


FIGURE 7.4: Sensor card used in the user study: Distance.

7.1.1 | Preparation

Our initial step, in preparation for the user study, involved creating a diagrammatic view of the IoT project used in the previous study (Figure 7.2), which laid the foundation for our troubleshooting approach. Inspired by the Surgical Safety Checklist, we developed a series of cards to guide participants through different aspects of troubleshooting within the project. We refined this to three key cards, which are central to our study (Figure 7.3 and Figure 7.4).

Each card is designed to guide participants through the troubleshooting process sequentially, featuring tick boxes that allow for systematic progress tracking. Furthermore, these cards include a QR code linking to an instructional website (Figure 7.5), thus creating a comprehensive hybrid tool that bolsters the problem-solving capabilities of end-user developers within a custom IoT system. This website provides detailed instructions, aiding participants in identifying and understanding the causes of malfunctions. By integrating tangible and digital elements, this approach significantly enhances the efficiency of the troubleshooting process.

7.2 | Study Method

In our study, we employed the aforementioned bespoke cards alongside the specially designed website to explore whether end-user developers could effectively identify and resolve issues within a custom IoT project. Participants were engaged with a project that had been deliberately modified to introduce a software mistake: a mismatch between the physical connection of the sensor to the microcontroller (using port D0) and the software configuration, which erroneously used port D7, following the default setting in the Ultrasonic library. This misconfiguration resulted in system malfunction. Tasked with utilising the guidance provided by the physical cards and the instructional website, participants aimed to diagnose and correct this discrepancy, testing their ability to address and solve the problem effectively.

The phases and procedures are outlined in Subsection 7.2.1, the participant invitation process is described in Subsection 7.2.2, and the Subsection 7.2.3 details the structure and relationship

FIGURE 7.5: Initial page of the instructional web resource, accessed by users through the QR code on the Microcontroller 1 card.

between the different components of data collection, and Subsection 7.2.4 outlines the ethical approval and consent procedures.

7.2.1 | Parts and Procedure

In the initial phase of our user study, participants were greeted and welcomed, setting a cordial tone for the engagement. Following the introduction, we moved into the different parts of the study, as presented in the following sections.

Participants were introduced to the materials through a brief verbal presentation delivered at the beginning of the session, outlining the objective of the activity: to identify and resolve a fault in a malfunctioning IoT prototype. The physical setup was presented alongside the instructional cards, which included a step-by-step diagnostic protocol with checkboxes and the corresponding support website. Participants were instructed to follow the provided instructions, combining physical inspection with digital guidance. Verbal confirmation was used to ensure each participant understood how to proceed before beginning. No additional orientation or training was provided.

First part: survey

Participants were presented with a Qualtrics Survey that included questions for recording consent, ensuring ethical compliance and transparency in data collection.

The survey is divided into two sections: demographics and knowledge/experience, as detailed in Appendix D. The demographic section collects basic information such as age and education level. In the knowledge and experience section, we explore the participants' familiarity with digital technologies, including the Internet of Things (IoT), and their awareness of community resources and support mechanisms. The survey is almost the same as used in the previous study (presented in Chapter 6), with an extra question designed to gauge participants' problem-solving skills and ability to identify malfunctions in electrical, electronic, or digital devices: *"How obvious is it for you to spot the cause of an electrical, electronic or digital device malfunction?"*

Second part: use of instructional cards and website

After securing consent after the survey, we commenced recording to chronicle the study proceedings. Participants were acquainted with the experimental arrangement, including a custom circuit and specialised cards integral to troubleshooting. The utility of these cards was meticulously explained, underscoring their connection to online tutorials that serve as digital aids.

As the study progressed into its second phase, we presented a comprehensive diagram of the custom IoT project. It was accompanied by three physical cards with QR codes linking to digital instructions. This innovative approach effectively merges tangible guidance with digital resources, streamlining the error identification process. Through a wizard-style sequence on our instructional website, participants systematically reviewed each project component to detect malfunctions. This structured, user-friendly methodology was designed to enhance the troubleshooting process. The task offered insight into how effectively the provided materials and instructions supported participants in identifying and resolving issues. Furthermore, insights were gained into participants' strategies for overcoming challenges, shedding light on the support system's success in navigating unexpected issues.

Completion of the diagnostic task was defined as reaching the final stage of the instructional material, successfully identifying the fault in the system, and restoring the prototype to a functioning state. The error was deliberately embedded and had a known resolution path. All participants ultimately completed the task; however, a few required facilitator assistance to locate the problem, as they had difficulty identifying it independently. No partial success criteria or performance thresholds were defined, as the primary aim was to observe how participants navigated the diagnostic process using the provided support materials.

This phase prepared participants for the tasks ahead and established a foundation for the subsequent semi-structured interview, designed to elicit insightful feedback on their experience.

Third part: semi-structured interview

The final component of the study involved a semi-structured interview, detailed in Appendix C, where participants were asked a series of open-ended questions to capture their perspectives on the project. This interview sought to gather participants' reflections post-completion, encouraging them to describe their experience in their own words. Key areas of focus included their perception of the time investment required, the learning curve they experienced while engaging with the project, and their overall journey. By discussing both their failures and successes, we aimed to identify the critical bottlenecks in the process and explore potential solutions that could more effectively support end-user developers in similar IoT projects.

7.2.2 | Inviting Participants

Building on the experience gained from the user study detailed in Chapter 6, we developed a series of posters to recruit participants for a new study. These posters were strategically placed across the University of Waikato campus, especially in areas not predominantly associated with technical disciplines like the School of Computing and Mathematical Sciences or the School of Engineering. Our goal was to engage a diverse group of participants, particularly those who might not have a strong technical background, aligning with our focus on end-user developers. Each poster included a QR code linking to a Google Forms page for expressing interest in the study.

In response to the posters, 22 individuals completed the interest form. Additionally, we contacted 27 people from our previous study's respondent pool, including the 12 who had participated in it. Ultimately, 12 participants were involved in the study between November 2022 and July 2023.

Four out of the twelve participants had previously participated in the study presented in Chapter 6, rendering them more cognisant of the project and somewhat acquainted with the associated technology. Nevertheless, the temporal span between their involvement in these two studies ranged from a minimum of seven months to a maximum of one year and six months. Given that end-user developers may not engage with this technology regularly, it is plausible that their recollection of it is somewhat limited.

7.2.3 | Data Collection Structure

The study consisted of three distinct components: an initial survey, a diagnostic task involving instructional materials, and a concluding semi-structured interview. These components were conducted within a single session, but played different methodological roles. The survey was primarily used to identify potential outliers in the participant cohort and was not intended to inform or interpret the interview results. In contrast, the diagnostic task directly influenced the subsequent interview by providing a shared experiential context; this enabled the facilitator to tailor questions in response to each participant's specific actions or omissions during the task. Data from the survey and the interviews were analysed independently, contributing to parallel interpretative strands within the study.

7.2.4 | Ethical Considerations

The study received ethical approval from the School of Computing and Mathematical Sciences Ethics Committee at the University of Waikato. The approval letter is included in Appendix A. Informed consent was obtained from all participants before the commencement of the session, which included the survey, diagnostic activity, and post-task interview. Participants were informed that video and audio recordings would be made for research purposes. No specific ethical concerns were identified regarding participant vulnerability, data sensitivity, or prior relationships.

7.3 | Methodology for Data Analysis

Completing our user study yielded a substantial collection of data, encompassing survey responses, detailed video recordings totalling 13 hours, 21 minutes, and 56 seconds, and responses to three Likert scale questions from the latter part of the semi-structured interviews. These recordings meticulously documented participants' engagement with the instructional materials and their reactions during the interviews, providing a foundation for quantitative and qualitative analysis. This comprehensive data set highlights the participants' experiences and viewpoints and includes insights from the Qualtrics survey and a nuanced examination of the video and interview responses. Our analytical approach combined statistical tools for survey data with thematic analysis for the video content and interview feedback. This integrated methodology ensures a thorough understanding of participant interactions and experiences. The subsequent sections detail the preparatory steps and the analytical framework used, setting the stage for an exhaustive and multi-dimensional analysis of the collected data.

7.3.1 | Preparing for Quantitative Analysis

As a result of the user study, we have two distinct data sets for quantitative analysis: data derived from the preliminary survey and data from the final questions in the semi-structured interviews.

The preliminary survey, akin to the approach in our preceding study, was partitioned into two segments: a) demographics; and b) knowledge and experience (Appendix D). The *Demographics* section aimed to ensure a heterogeneous participant distribution. Ensuring a fair distribution of participants across various demographics, particularly age and educational backgrounds, before commencing a deeper qualitative analysis was essential to prevent the over-representation of specific groups, such as an abundance of 18-24-year-olds with high school education. In the event of such skewed representation, we would have rolled back and added more participants to the user study.

In the *Knowledge and Experience* segment, the survey sought to gauge participants' familiarity with digital technologies. This assessment encompassed their interactions with digital devices

TABLE 7.1: Themes and codes.

Themes	Quotations	Codes (Quotations)
Challenges	365	Confusion (112), Uncertainty (106), Frustration (35), Lack of knowledge (9), Self-doubt (8), ...
Keywords	189	Technology (28), Interest (11), Success (11), Testing (11), Confirmation (8)...
Achievement	173	Curiosity (18), Clarity (14), Confidence (10), Efficiency (10), Satisfaction (7), ...
Learning and knowledge acquisition	128	Problem-solving (51), Decision-making (8), Trial and error (8), Experimentation (5), Observation (5), ...
Technology-related challenges	131	Technical difficulties (17), Technical troubleshooting (14), Troubleshooting (12), Technical knowledge (7), Technical (6), ...
Cognitive processes	59	Attention to detail (12), Self-reflection (5), Analytical thinking (4), Comprehension (3), Categorization (2), ...
Perception	59	Questioning (6), Surprise (6), Doubt (4), Expectation (4), Regret (3), ...
Instructions	43	Instructions (27), Technical instructions (3), Following instructions (2), Adherence to instructions (1), Awareness of the importance of instructions (1), ...
Coding skills	36	Coding (3), Task completion (2), Time estimation (2), Time management (2), Basic functions (1), ...
Seeking information	25	Desire for clarity (2), Need for clarification (2), Assistance (1), Desire for control (1), Desire for explanation (1), ...

and services and their awareness of public digital resources like internet forums, which could be pivotal during the project's assembly phase. The survey clarified to participants that 'digital technologies' encompass electronic tools, systems, devices, and resources functioning with digital data, primarily binary 0s and 1s, including but not limited to websites, mobile phones, digital televisions, and music players. This section was designed to understand participants' interaction levels with digital technologies, focusing on familiarity and satisfaction rather than technical expertise.

In the concluding phase of the semi-structured interviews, participants responded to three key questions using Likert scales:

- Evaluation of the process in terms of difficulty.
- Frequency of encountering uncertain situations.
- Regularity of experiencing clarity in the action path.

This structured approach enabled the quantification of data. Such visual representations will facilitate deeper insights into participants' perceptions of the process, enriching our understanding of their profiles and experiences.

7.3.2 | Preparing for Qualitative Analysis

Following the methodology of our prior study (Chapter 6), we recorded videos post-survey with participant consent, totalling 13 hours, 21 minutes, and 56 seconds. These recordings detailed participant interactions with the study protocol and interviews.

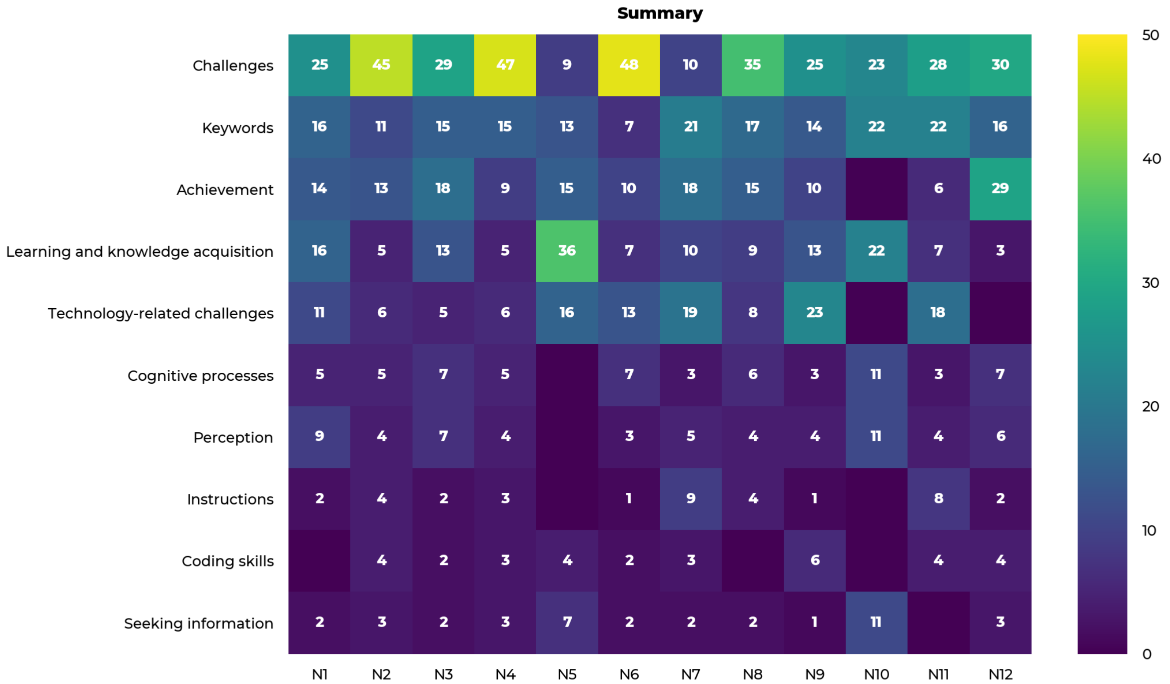


FIGURE 7.6: Percentage distribution of quotations by participants and themes.

We applied thematic analysis to the video data for an in-depth understanding of participant experiences. The recordings were transcribed into a 79,166-word dataset using AssemblyAI[1], capturing all participant-researcher dialogues.

For further analysis, we used ATLAS.ti[2], employing its AI Coding feature to initially code the data. This process identified 183 codes, organized into ten thematic categories (Table 7.1), and annotated 514 quotations from the 12 participants.

Mirroring our approach in Chapter 6, we created a heat map to display the distribution of quotations across codes, ensuring even spread and identifying any coding errors. The heat map (Figure 7.6) shows participant quotes across ten themes, with rows for themes and columns for participants. Each cell reflects the percentage of quotes per participant per theme, totalling 100% across all columns. This visualisation affirms the integrity of our coding process, demonstrating a fair distribution of quotations across both codes and participants. Despite certain themes, such as *Keywords*, accruing more quotations, this does not signify a concentration that could suggest bias. The equitable spread of quotations ensures that no single area disproportionately influences the analysis, reinforcing the objectivity of our methodological approach.

Our analysis is divided into two parts based on the distribution of quotations across themes. The first part focuses on the top three themes – *Challenges*, *Keywords*, and *Achievement*, accounting for 58% of the quotations, which are analysed individually. The remaining 42%, comprising themes

[1] https://www.assemblyai.com/

[2] https://atlasti.com/

like *Learning and Knowledge Acquisition*, *Technology-Related Challenges*, and others, are grouped into conceptual categories: *Educational Processes*, *Technological Competence*, and *Information Utilisation* for collective analysis.

This final list outlines the themes and categories applied in analysing participants' comments from the user study.

- **Theme #1: Challenges**
- **Theme #2: Keywords**
- **Theme #3: Achievements**
- **Conceptual Category #1: Educational processes**
  Themes: *Learning and knowledge acquisition*, *Cognitive processes*, and *Perception*.
- **Conceptual Category #2: Technological competence**
  Themes: *Technology-related challenges*, and *Coding skills*.
- **Conceptual Category #3: Information utilisation**
  Themes: *Seeking information*, and *Instructions*.

The **Theme #1: Challenges** accounts for 29% of participant quotations (Figure 7.6), primarily featuring emotional responses like *Confusion* (112 quotations), *Uncertainty* (106 quotations), and *Frustration* (35 quotations), which together constitute approximately 69% of this theme's quotes. These responses highlight a common experience among participants. The remaining 30% of quotes in this theme fall into three categories: *Other emotional responses*, covering additional emotional reactions to challenges; *Cognitive limitations*, reflecting issues in understanding or processing information; and *Situational difficulties*, on external challenges unrelated to personal or emotional factors.

Following, the **Theme #2: Keywords** emerges as the second most significant, comprising 189 quotations across various codes such as *Technology*, *Interest*, and *Success*, and accounting for an average of 16% of quotations per participant (Figure 7.6). These keywords are conceptually categorised into five groups: Cognitive and Learning Aspects (12 codes, 40 quotations), Communication and Interaction (10 codes, 16 quotations), Emotional and Psychological Factors (15 codes, 35 quotations), Methodology and Process (12 codes, 34 quotations), and Technical Aspects and Tools (13 codes, 56 quotations).

The **Theme #3: Achievements** encompasses 132 quotations, accounting for 13% of participant quotations. It includes 73 unique codes, with the most frequent being *Curiosity* (18 quotations), *Clarity* (14), *Confidence* (10), *Efficiency* (10), and others with fewer quotations. The analysis priorities codes with higher quotations, such as *Curiosity*, *Clarity*, *Confidence*, and *Efficiency*. The remaining codes are grouped into Personal Growth and Motivation (30 codes, 46 quotations), Outcome and Process Evaluation (17 codes, 31 quotations), and Engagement and Social Dynamics (22 codes, 44 quotations).

Finally, the **Conceptual Category #1: Educational processes** encompasses themes such as *Learning and knowledge acquisition*, *Cognitive processes*, and *Perception*. The **Conceptual Category #2: Technological competence**, combines *Technology-related challenges* and *Coding skills* and the final category, **Conceptual Category #3: Information utilisation**, includes *Seeking information* and *Instructions*.

The following section presents insights from our user study, focusing on user engagement, challenges, and learning behaviours in IoT projects. These findings are crucial for developing more effective support and educational resources for end-user developers.

7.4 | Analysis

This section presents key insights from our user study on a toolkit – cards and online material – designed to assist end-user developers in IoT projects. Insights include user engagement, challenges, and learning behaviours derived from surveys, observations, and analysis of recordings. These findings highlight difficulties faced by end users in troubleshooting IoT malfunctions and identify knowledge gaps in the IoT field, aligning with our goal of providing necessary support for IoT system creation and use. Moreover, while illustrating the efficacy of our intervention in a particularly narrow scenario, these results underscore the inherent challenges of formulating a universal approach. This information is crucial for developing focused support tools and educational resources to improve IoT system efficacy and user-friendliness.

In the following sections, Subsection 7.4.1 presents the demographic from the user study. Subsection 7.4.2 shows the background of the participants in the user study. Subsection 7.4.3 presents user engagement and experience insights. Subsection 7.4.4 highlights challenges users faced in navigating and troubleshooting the IoT system, utilising data from process observations and transcribed recordings. Subsection 7.4.5 discusses how users learned and adapted, drawing on findings from the preliminary survey, process observations, and transcribed recordings. Subsection 7.4.6 focuses on key insights from the transcribed recordings, particularly on participant feedback.

7.4.1 | Demographics

The demographics of our user study, coming out from the preliminary survey, indicate a diverse distribution of participants' ages. Specifically, the age distribution was as follows: one participant was under 18, four fell within the 18-24 age range, four were in the 25-34 age range, two were in the 35-44 age range, and one participant was 65 or older (Figure 7.7). Notably, a conspicuous absence of participants fell within the 45-54 or 55-64 age brackets, indicating a skew toward individuals under 45 in this sample group.

The distribution of participants in our user study reveals a diverse range of educational backgrounds (Figure 7.8). Specifically, we found four participants held a high school degree or equivalent (e.g., GED), making it the most prevalent educational qualification within our sample group. Three participants each possessed a Bachelor's degree (e.g., BA or BS) and a Doctorate (e.g.,

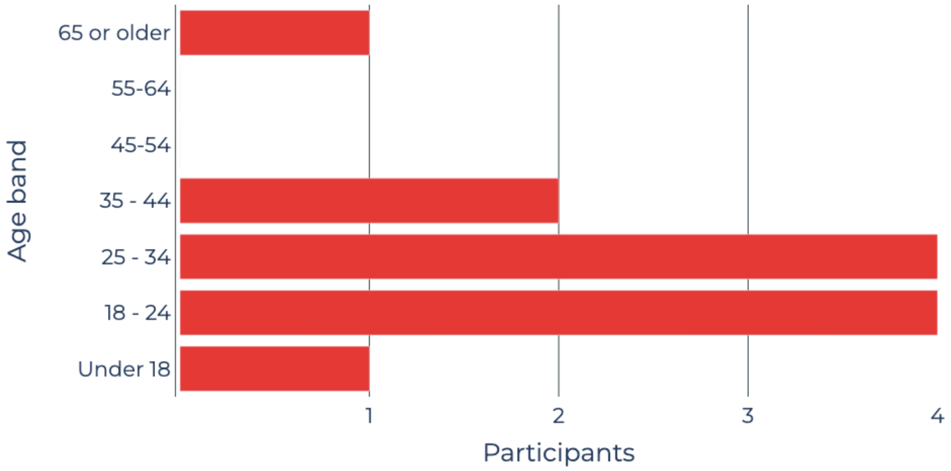


FIGURE 7.7: Age of the participants.

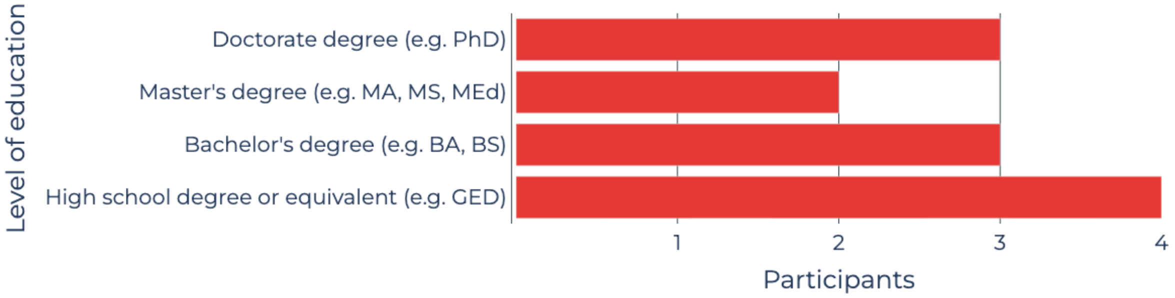


FIGURE 7.8: Highest degree of education completed.

PhD), showcasing a mix of undergraduate and advanced academic backgrounds. Additionally, two participants held a Master's degree (e.g., MA, MS, MEd), indicating a level of postgraduate education within the cohort. These findings highlight the variety of educational experiences among our study's participants, which can be critical in understanding their perspectives and interactions with the material presented in the study.

Our participants' diverse ages and educational backgrounds provide a broad spectrum of perspectives and learning approaches, which are crucial in understanding how end-user developers interact with IoT systems. This diversity helps ensure that the insights gleaned from our study represent a wide range of user experiences and challenges, thereby enriching our understanding of how different groups might approach problem-solving in the context of IoT projects.

7.4.2 | Background Knowledge of Participants

This section introduces participants' backgrounds based on survey results, highlighting their IoT knowledge and experience. This context aids in interpreting their interaction, troubleshooting challenges, and learning outcomes, informing our analysis of engagement and feedback from testing cards, website navigation, and interviews.

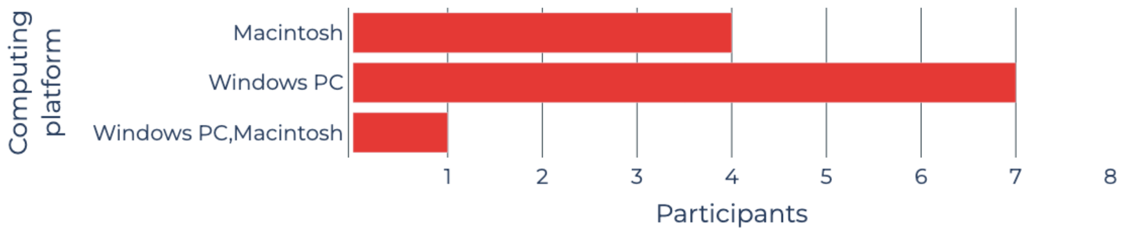


FIGURE 7.9: Computing platforms commonly used.

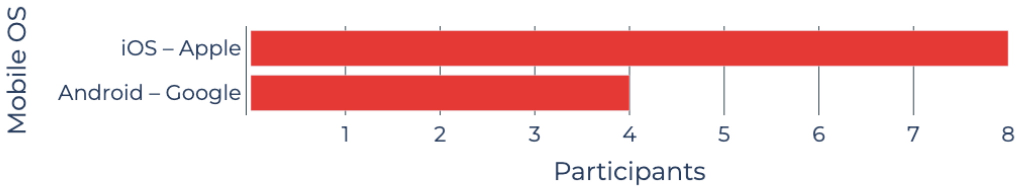


FIGURE 7.10: Mobile operating systems used.

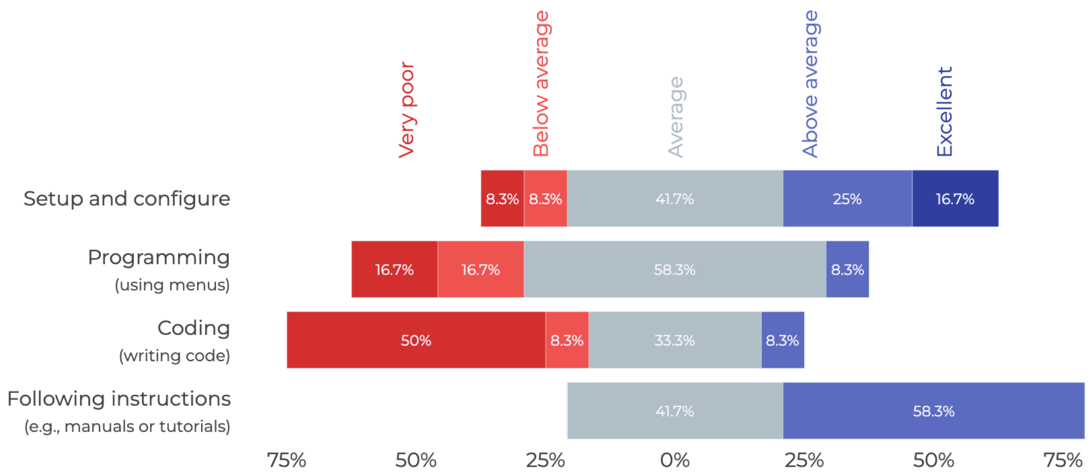


FIGURE 7.11: Knowledge. How would you qualify your digital technologies knowledge?

Computers and mobile devices

The survey began by asking participants about their computing preferences: 7 chose Windows PC, 4 Macintosh, and 1 used both, aligning with concerns about the dominant use of Windows despite the study being conducted on Macintosh (Figure 7.9). The use of the cross-platform Firefox browser mitigated potential issues. Mobile OS preferences showed 4 for Android and 8 for iOS (Figure 7.10), noting an interesting contrast in Apple device usage between mobile and computers.

Digital technologies knowledge

Participants assessed their digital technology skills in the survey's Knowledge section (Figure 7.11). For Setup and Configure, 41.67% rated themselves as *Average*, 25% as *Above average*,

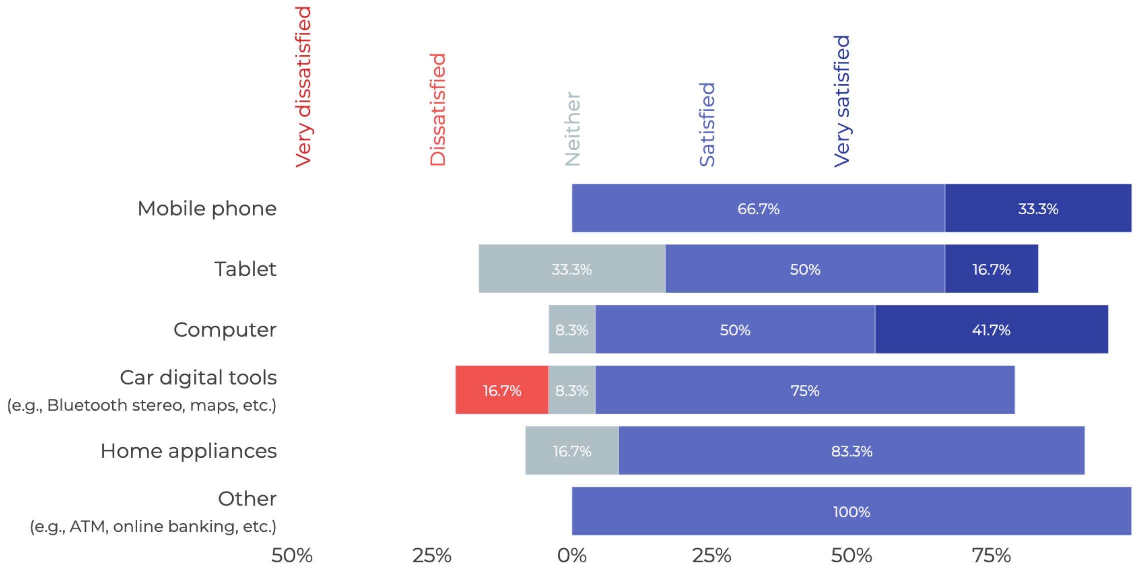


FIGURE 7.12: Experience. How would you consider your level of satisfaction with digital technologies?

and 16.7% as *Excellent*, with a small number feeling *Below average* or *Very poor* (8.33% each). In menu-driven programming, 58.33% felt *Average*, while 16.7% each were *Below average* and *Very poor*, and 8.33% *Above average*. For coding skills, 50% considered themselves *Very poor*, 33.33% *Average*, and a minor fraction either *Below average* or *Above average* (8.33% each). Conversely, in Following Instructions, 58.33% rated themselves *Above average* and 41.67% *Average*.

This section reveals most participants view their digital technology and menu-driven programming knowledge as *Average*, but half rate their coding skills as *Very poor*, suggesting a gap in coding proficiency. Yet, a strong ability to follow instructions was noted, with a majority rating themselves *Above average*, highlighting varied skill levels among participants and informing the analysis of user study outcomes.

Satisfaction with digital technologies

The Experience section of the survey asked participants about their satisfaction with digital technologies (Figure 7.12). In the Experience section, participants were asked about their satisfaction with digital technologies, including mobile phones, tablets, computers, car digital tools, home appliances, and other technologies like ATMs and online banking (Figure 7.12) – most reported satisfaction across these categories. Specifically, mobile phones saw 8 out of 12 participants satisfied, with the remainder highly satisfied. For tablets, half were satisfied, a third neutral, and the rest highly satisfied. Computers had an even split of satisfaction among 6 participants, with 41.6% highly satisfied and one neutral. Car digital tools had 75% of respondents highly satisfied, albeit two expressed dissatisfaction. Home appliances were positively received by most (10 participants), with two neutral. Notably, satisfaction with ATMs and online banking was unanimous.

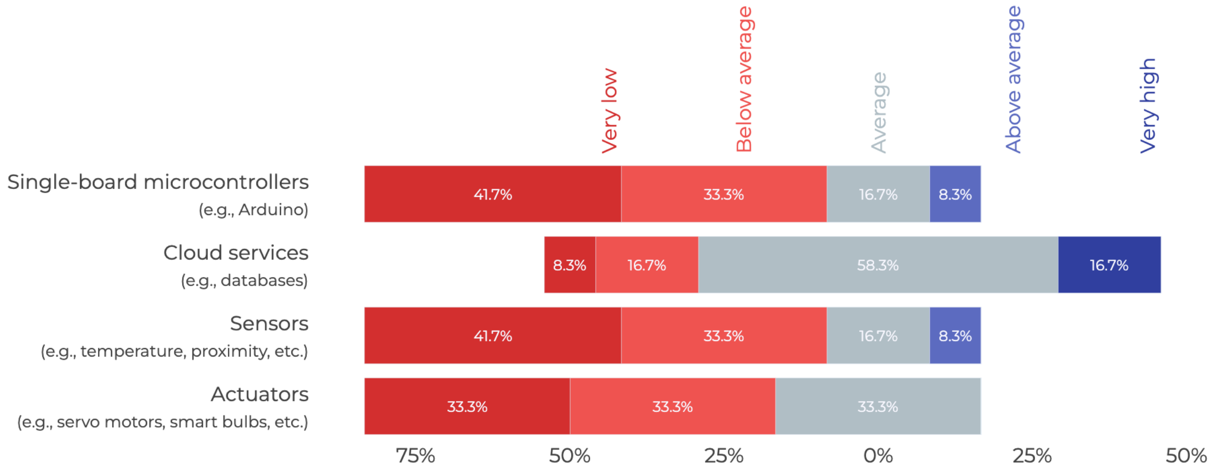


FIGURE 7.13: Internet of Things (IoT). How would you rate your knowledge of IoT technologies?

These results suggest participants' high satisfaction level with digital technologies, indicating general comfort and familiarity. High satisfaction with home appliances and car tools highlights their perceived effectiveness and ease of use. However, neutrality and dissatisfaction, especially with car tools, suggest areas for further user experience research and improvement, underscoring the significant, generally positive impact of digital technology in daily life.

Knowledge of Internet of Things technology

In the IoT section of the survey (Figure 7.13), participants assessed their knowledge of IoT technologies, such as Arduino, cloud services, and various sensors and actuators, to measure their familiarity with key IoT components. This evaluation aimed to identify needed support types and ensure a balanced representation of technological expertise among end-user developers.

Survey results showed varied understanding of IoT technologies. For Arduino and similar microcontrollers, 41.6% reported very low knowledge, 33.3% below average, with only a few claiming average or above average knowledge. Cloud services saw a more positive response, with 58.3% rating their knowledge as average, a few as below average or very low, and two as very high. Knowledge of sensors paralleled that of microcontrollers, with many participants indicating low familiarity. Actuators saw a more evenly distributed knowledge level, with participants split across very low, below average, and average categories.

The survey indicates diverse familiarity with IoT technologies among participants, with a notable gap in understanding foundational components like microcontrollers and sensors. Conversely, cloud services appear more familiar, suggesting different support requirements for participants in IoT projects.

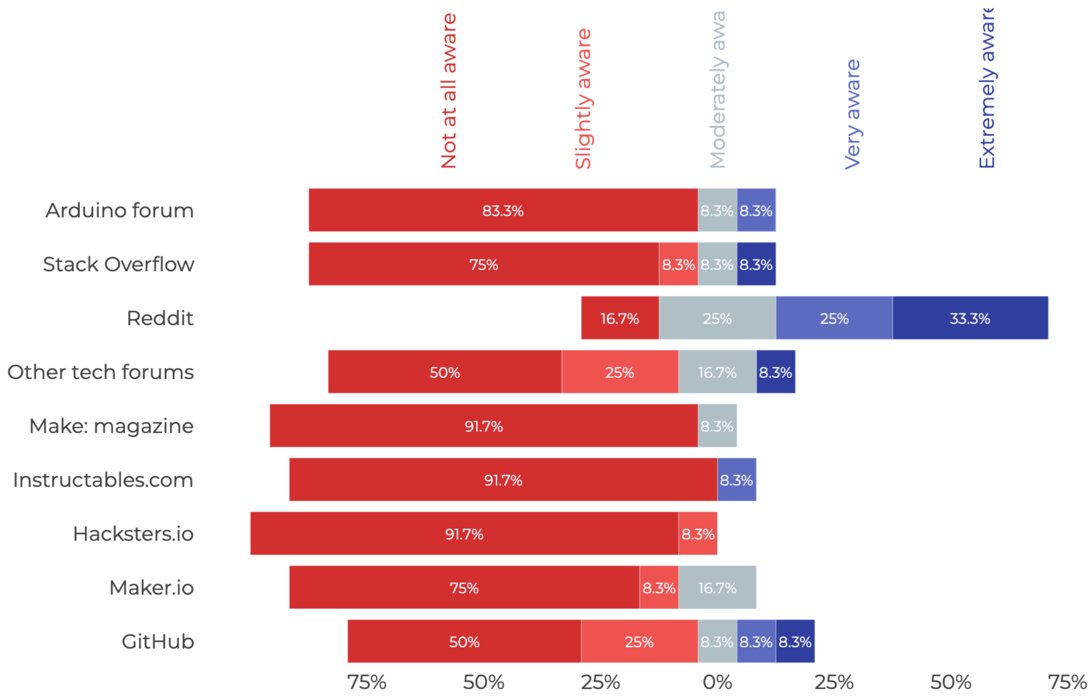


FIGURE 7.14: Community. How would you rate your awareness of the following community-based resources?

Community

The survey explored participants' familiarity with community-based resources critical for IoT projects, including the *Arduino Forum*, *Stack Overflow*, *Reddit*, and various online platforms and publications like *Make:* magazine and *GitHub* (Figure 7.14). This aimed to assess their awareness of support and troubleshooting resources.

Findings show a general lack of awareness: 83.3% were unfamiliar with the *Arduino Forum*, 75% with *Stack Overflow*, and over 90% with *Make:* magazine, *Instructables.com*, and *Hacksters.io*. *Reddit* saw more balanced awareness, with a mix of participants across awareness levels, and *GitHub* had a slightly better spread of awareness, yet 50% remained unaware.

These results highlight a significant gap in familiarity with essential IoT community resources, suggesting participants may lack knowledge crucial for troubleshooting and learning. The variance in awareness of platforms like *Reddit* and *GitHub* suggests some exposure to digital platforms. Still, overall, there is a clear need for educational initiatives to improve engagement with community-based IoT resources.

Help and assistance

The survey assessed how participants seek online help for digital technology issues beyond Google searches, exploring methods like forum posts, chats with experienced individuals, and social media inquiries (Figure 7.15). This aimed to identify preferred methods for resolving digital technology challenges.

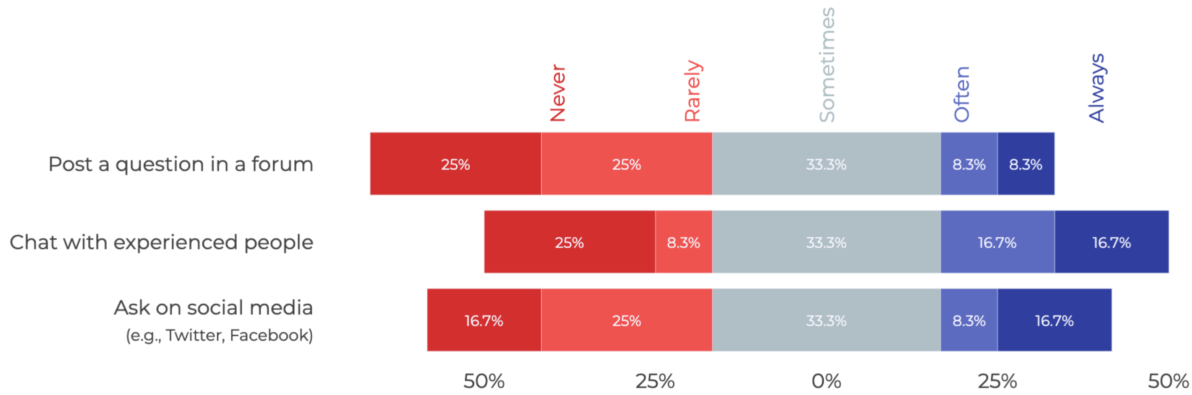


FIGURE 7.15: Help and assistance. How frequently do you ask for help online? (beyond "asking Google")

Responses to posting in forums showed 33.3% sometimes do so, 25% rarely, another 25% never, with smaller percentages often or always engaging. For seeking help through chats, 33.3% sometimes engage, 25% never, and the rest vary between often and always, with a small portion rarely using this method. Social media usage for help was mixed, with 33.3% occasionally, 25% rarely, and differing levels of regular use.

Participants exhibited varied preferences in seeking online assistance, with a notable number occasionally turning to forums or social media. This indicates a spectrum of approaches to obtaining help, from formal forums to casual chats and social media platforms, highlighting the need for a multi-faceted support system to effectively address the diverse needs of end-user developers in IoT projects.

Identifying problems

The survey assessed participants' skills in identifying malfunctions across various device categories, including household and smart appliances, audiovisual systems, remotely controlled devices, computers, and wireless devices (Figure 7.16).

- **Household Appliances**: 41.7% reported moderate ease, 25% found it slightly easy, another 25% very easy, and 8.3% extremely easy.
- **Smart Appliances**: Half experienced moderate ease, a third found it slightly or very easy (16.7% each), and 8.3% extremely easy.
- **Audiovisual Systems**: A third found it slightly easy, 25% moderately easy, and one-third experienced it as very or extremely easy (16.7% each), with 8.3% finding it not at all easy.
- **Remotely Controlled Devices**: A third stated it was slightly easy, half reported moderate to very easy (25% each), with small groups experiencing extreme difficulty or ease (8.3% each).

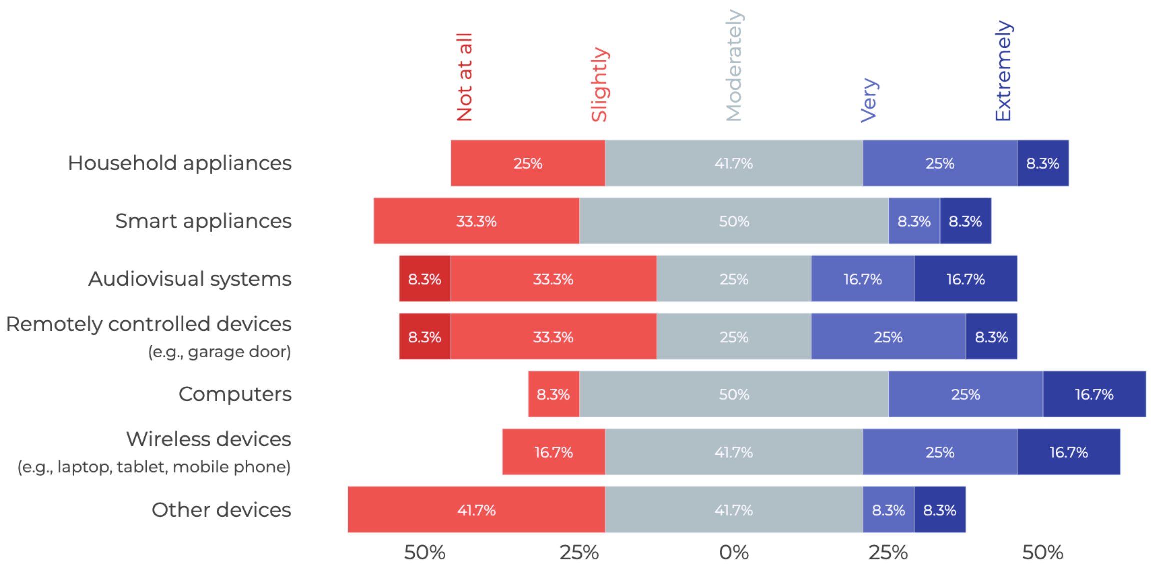


FIGURE 7.16: Identifying problems. How easily can you spot the cause of an electrical, electronic or digital device malfunction?

- **Computers**: Half found diagnosing moderately easy, a quarter very easy, 16.7% extremely easy, and 8.3% slightly easy.
- **Wireless Devices**: 41.7% stated moderate ease, 25% very easy, and a third found it either extremely difficult or easy (16.7% each).
- **Other Devices**: Over 80% found it slightly or moderately easy (41.7% each), and 16.7% experienced it as very or extremely easy (8.3% each).

The results indicate varied confidence and ability in diagnosing malfunctions, with higher ease reported in household and smart appliances, and a broad capability range across other categories.

Following a process

The survey questioned participants about their troubleshooting routines across various device types, including electrical and electronic devices, computers (hardware and software), mobile phones, tablets, network connectivity, printers, and others (Figure 7.17).

- **Electrical Devices**: 41.7% sometimes, 25% rarely, and 16.7% often follow a routine, with 8.3% always and never doing so.
- **Electronic Devices**: Half sometimes, 25% often, and 16.7% always follow a routine, with 8.3% rarely and none never.
- **Computer Hardware**: Practices varied; 50% rarely, 16.7% sometimes or often, and 8.3% always or never follow a routine.

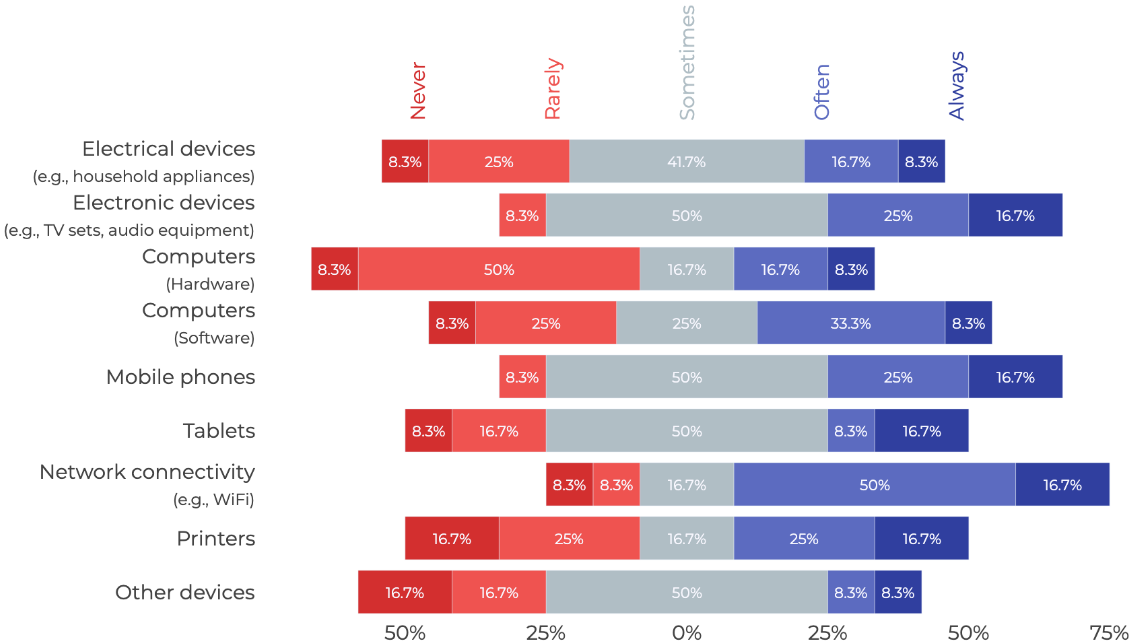


FIGURE 7.17: Process. Do you have a set of steps you always follow to find the cause of a problem or malfunction?

- **Computer Software**: A third often, 25% sometimes or rarely, and 8.3% always or never use a routine.
- **Mobile Phones**: Half sometimes, a quarter often, 16.7% always, and 8.3% rarely follow a routine.
- **Tablets**: Half sometimes, 16.7% rarely or always, 8.3% often, and another 8.3% never follow a routine.
- **Network Connectivity**: Half often, 16.7% sometimes or always, and 8.3% rarely or never follow a routine.
- **Printers**: 25% rarely, 16.7% sometimes, often, or always, and 16.7% never follow a routine.
- **Other Devices**: Half sometimes, 16.7% rarely or always, 8.3% often, and 16.7% never follow a routine.

The findings reveal diverse approaches to troubleshooting, with notable habits like often following a routine for network connectivity issues and varied practices for other devices, reflecting the participants' flexibility and adaptability in addressing technical problems.

7.4.3 | User Engagement and Experience

In this subsection, we present how participants engaged with and experienced the toolkit of cards and online instructions. The insights presented here come from the recordings, our observations during the process and the final semi-structured interview with participants.

Emotional and psychological factors

This section addresses the emotional and psychological impact of interacting with technology, covering emotional responses, motivations, and impacts, including *Success*, *Difficulty*, and *Progress*. Key insights include:

- **Hierarchy** (Participant N1): Highlights emotional relief from structured approaches, reducing frustration and uncertainty.
- **Success** (Participants N3 and N11): Describes the validation and achievement felt upon completing tasks, underscoring how success boosts confidence and satisfaction.

These examples underscore the significant role emotional and psychological factors play in technology interaction, highlighting the importance of structured problem-solving, as facilitated by the resources created for this user study and the positive effects of completing technical tasks.

Hardware vs. Software

Participants in our study showed a clear preference for interacting with hardware over software. They found hardware tasks, like connecting components and using LED indicators, enjoyable and straightforward, compared to assembling LEGO™ blocks. This hands-on aspect boosted their confidence, making them comfortable in a familiar setting.

In contrast, many software tasks, especially coding in C++, were challenging. Participants described coding as an intimidating and foreign language, difficult to understand and apply despite its English-like syntax. This led to frustration and inadequacy, as they struggled with the seemingly familiar yet fundamentally different coding tasks.

Participants' reflections underscored these differences:

- *"...when the code wasn't working, that was pretty stressful... But got over it, so it's fine."* | PARTICIPANT N2
- *"...I didn't work with any software."* | PARTICIPANT N4
- *"That's my first time doing this..."* | PARTICIPANT N8
- *"...in the coding there was a lot... It perhaps assumed a higher level of computing familiarity."* | PARTICIPANT N9

After the experiment, participants felt more confident but acknowledged the need for practice to fully grasp and remember the process, as indicated in follow-up interviews.

7.4.4 | Challenges in Navigation and Troubleshooting

This section examines participants' difficulties navigating instructional materials and resolving issues while running the experiment. It aims to illuminate the complexities of understanding and applying instructions, pinpointing where users struggled and why. By analysing these difficulties, we seek to uncover insights that can guide the improvement of instructional design, making it more intuitive and accessible for end users.

Confusion

These insights result from analysing recordings coded as *Confusion* within the *Challenges* theme, highlighting areas where participants encountered difficulties. Participants faced confusion due to unfamiliar tasks, unclear objectives, and technical terminology while working on the malfunctioning IoT project. Key instances include:

- **Encountering unfamiliar complexity**: Participant N3 felt overwhelmed by the sudden complexity, indicating a steep learning curve for those without technical backgrounds.
- **Uncertainty in purpose and process**: Participant N8 lacked clarity on the device's purpose and necessary steps, highlighting the need for clearer guidance.
- **Challenges in understanding technical terminology**: Participant N12 struggled with technical jargon, pointing to the importance of clear instructions and visual aids.

These experiences emphasise the need for assistance tools designed from the end-user perspective, with intuitive interfaces, simplified language, and visual aids to reduce confusion and improve problem-solving in IoT development.

Uncertainty

Exploring the theme of *Uncertainty* revealed participants' struggles with ambiguous situations and unclear decision-making in IoT projects. Key areas include:

- **Uncertainty in problem identification:** Participant N7 faced difficulties identifying and resolving issues due to unclear device functionality.
- **Uncertainty in response to technical issues:** Participant N9 experienced indecision when encountering non-functional components, highlighting the need for clearer expectations.
- **Uncertainty in executing steps:** Participant N12 was unsure about specific actions, such as uploading, indicating a need for more explicit guidance.

These examples show that to improve end-user development in IoT, assisting tools must offer clear instructions, feedback, and troubleshooting tips to minimise uncertainty and enhance problem-solving.

Frustration

Exploring *Challenges: Frustration*, we find it significantly affects end-user developers' experiences, indicating the emotional impact and problem-solving barriers frustration introduces in malfunctioning IoT projects. Key instances include:

- **Overwhelming technical challenges:** Participant N6's frustration with unresolved technical issues highlights the emotional strain of continuous problems.
- **Frustration with instructions and understanding:** Participant N7's irritation with confusing instructions points to clarity and logical guidance.
- **Obstacles in following instructions:** Participant N11's frustration from not progressing despite following instructions shows the need for more precise, actionable steps.

These examples emphasise the importance of designing assisting tools that are functional and considerate of the user's emotional journey, suggesting that clearer instructions, intuitive guides, and adaptable tools can reduce frustration and improve the IoT development experience for non-experts.

Emotional Responses, Cognitive Limitations, and Situational Difficulties

In our analysis, we created three categories, *Emotional Responses*, *Cognitive Limitations*, and *Situational Difficulties*, to capture the diverse challenges faced by end-user developers in IoT projects. However, we report these insights collectively to offer a unified view of how these interrelated factors affect user experience and problem-solving in IoT development.

Our unified report delves into the intertwined nature of Emotional Responses, Cognitive Limitations, and Situational Difficulties, as revealed through participant quotations. Collectively, these categories paint a comprehensive picture of the multifaceted challenges end-user developers face in IoT projects.

- **Emotional Responses** surface through stress and a sense of being lost as participants grapple with technical challenges and a lack of foundational knowledge. For instance, the stress from not knowing how to resolve a problem (Participant N2) and the overwhelming feeling of navigating an unfamiliar territory without sufficient knowledge (Participant N3) underscore the emotional toll on users.
- **Cognitive Limitations** highlight the struggle with understanding complex concepts and retaining crucial information. Participant N2's confusion about *"looping through the ports"* and Participant N9's attempt to make sense of logs without adequate knowledge illustrates the cognitive barriers that hinder effective problem-solving.
- **Situational Difficulties** reflect the external challenges impacting the problem-solving process, including the lack of experience and attention to detail. Participant N2's oversight in copying code without understanding its details and Participant N3's acknowledgement of their lack of experience reveal how situational factors compound cognitive and emotional challenges.

These insights underscore the need for IoT development tools and environments to be designed with a holistic understanding of end-user developers' experiences, addressing the emotional, cognitive, and situational dimensions to facilitate a more effective and satisfying engagement with technology.

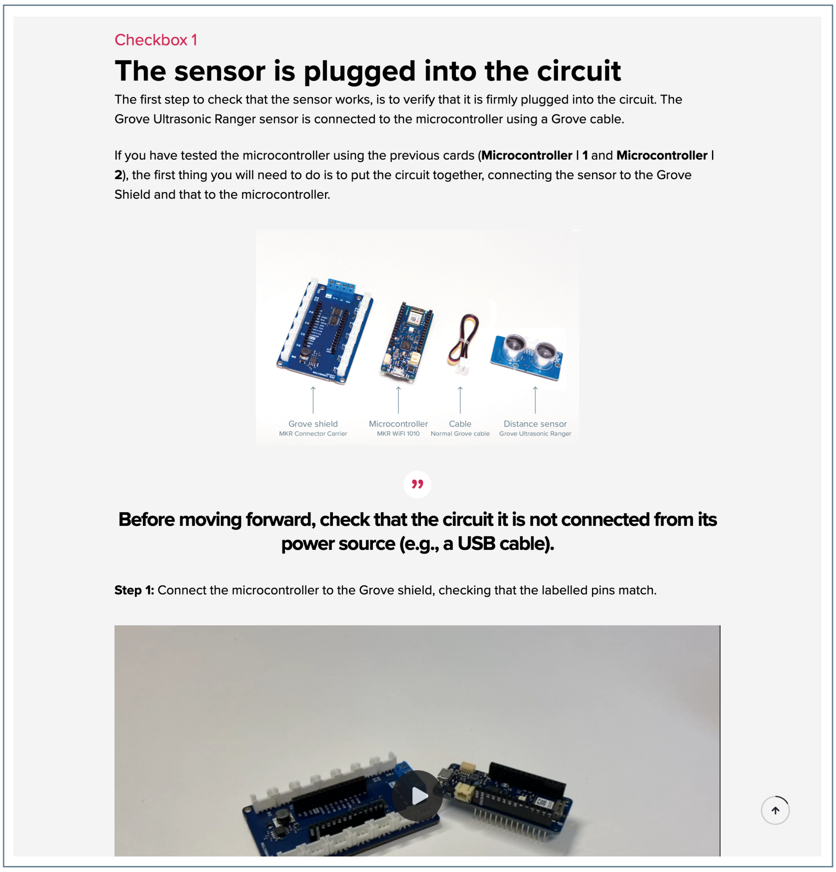


FIGURE 7.18: Section of the instructional website containing a safety warning overlooked by participants: *"Before moving forward, check that the circuit is not connected from its power source (e.g., a USB cable)"*.

Technical aspects and tools

This section examines participants' interactions with technology's practical elements, including hardware and software, and highlights *Technology*, *Programming*, and *Hardware* among key codes.

– **Hardware Understanding** (Participant N2): Discusses gaining a basic grasp of the hardware-software relationship, acknowledging the complexity of the subject.

– **Hands-on Hardware Experience** (Participant N7): Describes detailed interaction with a microcontroller, illustrating the tangible aspects of engaging with hardware.

– **From Basics to Custom Programming** (Participant N11): Shows the transition from basic examples to creating custom programs, emphasising the application of technical knowledge.

These insights reveal how participants navigate technical challenges and apply their understanding to develop functional solutions, underscoring the importance of hands-on experience and hardware and software knowledge integration.

Images vs. Text

After disassembling the circuit earlier, participants proceeded to the *Distance* card to test the malfunctioning sensor. The instructions, including images, text, and a video, aimed to guide

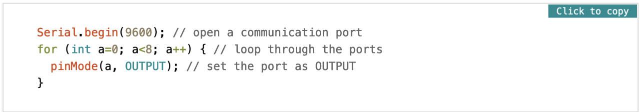


FIGURE 7.19: Instructional website's click-to-copy feature.

them in testing and reconnecting the sensor. Despite the prominent display of the instruction to first unplug the device from power (in bold and larger font), many overlooked this crucial step (Figure 7.18).

Upon realising the oversight, participants expressed concern over potentially damaging the components. However, they acknowledged the importance of such precautions after learning about the low risk due to the circuit's low voltage and the potential for a short circuit. This experience highlighted the need for clearer instructions, possibly ensuring devices are unplugged from power before proceeding with the task.

7.4.5 | Learning and Adaptation in User Behaviour

This section investigates how participants learned from their experiences and adapted their behaviour throughout the user study.

Cognitive and learning aspects

This segment focuses on the cognitive processes of grasping and applying technology, covering problem-solving, memory, and learning strategies. Key codes include *Interest*, *Understanding*, *Learning*, among others, reflecting the varied mental strategies participants employed.

– **Visualisation** (Participant N1): Demonstrates using visualisation to simplify complex systems, aiding understanding and learning.

– **Discovery** (Participant N1): Emphasises the *"light bulb"* moment, showcasing the role of hands-on experience and feedback in grasping new concepts.

– **Learning** (Participant N2): Highlights beginning stages of learning, acknowledging a basic understanding and the process of learning about the hardware-software interplay.

These examples underscore the significance of cognitive strategies like visualisation and discovery in learning and the recognition of learning as a layered process, providing insights into how participants navigate and absorb new technological concepts.

Copy and paste

Observations from Chapter 6 user study showed participants struggled to copy code manually, often missing characters. To address this, we introduced a *click to copy* feature, which allowed users to copy code by clicking a button (Figure 7.19). This change simplified code copying but did not clarify where to paste the code or help understand code syntax. Participants were confused about code structure, especially the use and nesting of curly brackets. Despite using the *Verify and Save* option in the Arduino Editor, which checks syntax, incorrect pasting locations led to error messages that participants found confusing (Figure 7.20). Compiler error messages

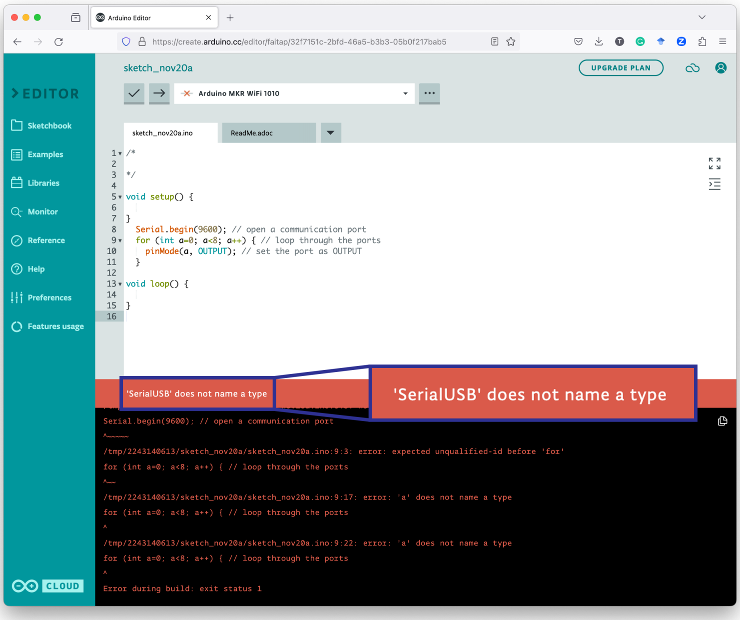


FIGURE 7.20: Error on the Arduino Editor. The message "*'Serial USB' does not name a type*" is difficult to interpret by an end-user developer.

have long challenged novice programmers, being difficult to understand and acting as barriers to learning [15, 130]. These cryptic messages, dating back to research from the 1960s, often frustrate and hinder progress [93]. End-user developers, like programming students, face similar difficulties, with error messages designed for experts not aiding in error correction for novices. This underscores the need for clearer, more accessible feedback in programming tools.

7.4.6 | Insights from Participant Feedback

This section presents insights from participant feedback, highlighting their experiences and perceptions throughout the study.

Communication and interaction

This section highlights the role of clear communication and interaction in using technology, emphasising the need for detailed explanations, effective feedback, and comparative analysis to facilitate understanding and engagement. Key codes include *Explanation*, *Feedback*, and *Communication*.

- **Explanation** (Participant N1): Points out the necessity of detailed explanations for deeper understanding and engagement.
- **Feedback** (Participant N7): Shows how feedback enhances comprehension and achievement.
- **Comparison** (Participant N9): Discusses the importance of comparing new information with known knowledge for effective learning.

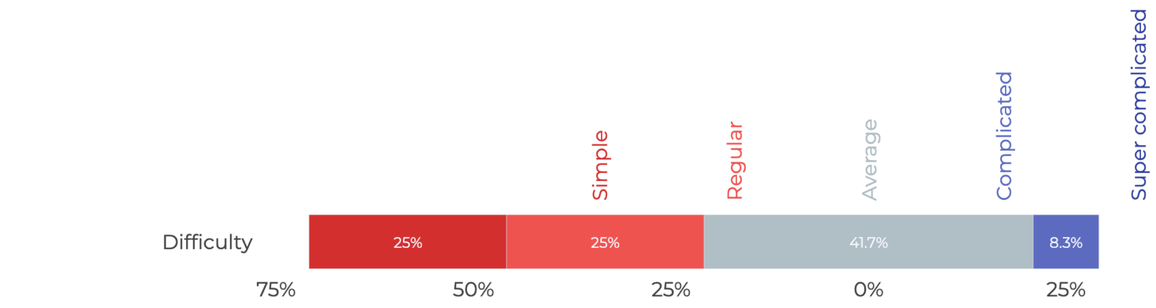


FIGURE 7.21: Process Evaluation: Assessing Difficulty. How do you evaluate the process in terms of difficulty?

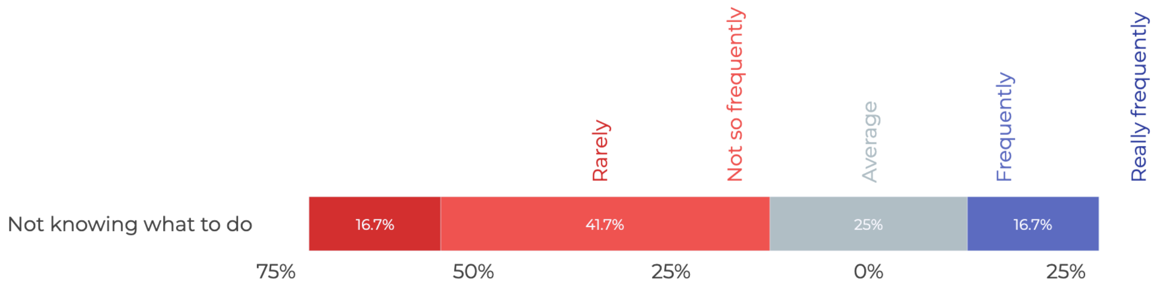


FIGURE 7.22: Navigational Challenges: Frequency of Uncertainty. How frequently did you encounter moments where you did not know what to do?

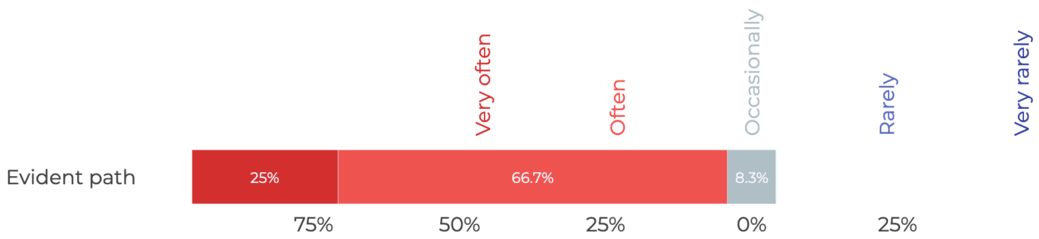


FIGURE 7.23: Clarity of Path: Occurrence of Evident Actions. How often did you experience moments where the path of actions was evident to you?

These insights underline the importance of clear communication and effective interaction in navigating and comprehending complex technological information.

Quantitative analysis of final questions

The study concluded with three Likert scale questions on participants' experience (Appendix C). Results showed that most found the process straightforward (Figure 7.21), rarely felt uncertain about the next steps (Figure 7.22), and saw a clear path of action (Figure 7.23). These outcomes suggest participants felt guided, though some uncertainty and frustration remained.

7.5 | Discussion

At the beginning of this chapter, we stated the research question that was guiding the user study we were going to embark on:

> **RQ4** | What forms of support do end-user developers require to efficiently identify and resolve issues in their IoT projects?

Our hypothesis centred around the idea that structured support through instructional cards linked to online resources could enhance the efficiency of end-user developers in troubleshooting and resolving issues within bespoke IoT projects. To test the validity of this hypothesis, we conducted a user study, presenting participants with a malfunctioning IoT project along with the devised support materials to observe their troubleshooting efficacy.

The study yielded a wealth of data, offering insights into how end-user developers employ such support methods to diagnose and address issues in malfunctioning IoT devices. In the ensuing paragraphs, we aim to bridge our findings with the initial research question, delineating the practical implications of our study and the potential avenues it opens for future research in the field. This approach not only validates our hypothesis but also underscores the critical role of accessible and structured support in the realm of IoT development, particularly for those with varying levels of expertise.

7.5.1 | Insights from the User Study

Our study implemented a guided process, enhanced by the strategic use of instructional cards and web resources, to evaluate how end-user developers tackle troubleshooting tasks in IoT settings. This systematic approach allowed us to capture a comprehensive snapshot of their problem-solving journey – highlighting the protocol's effectiveness in guiding users through complex troubleshooting scenarios. The nuanced feedback and observed interactions with these tools shed light on their strengths and areas for improvement. Through an in-depth analysis of moments where the protocol led to successful issue resolution and instances where users encountered hurdles, we aim to dissect the dynamic interplay between the designed support system and the users' practical needs. This evaluation serves as a foundation for refining our approach to developing user support tools, ensuring they are intuitive and closely aligned with real-world troubleshooting challenges.

Highlights of success

One notable success observed in our study was the enhanced problem-solving efficiency among end-user developers, attributable to the troubleshooting guide supported by instructional cards and web resources. This approach provided clear, step-by-step guidance that significantly reduced the cognitive load, allowing users to focus on diagnosing and resolving issues more efficiently. The protocol facilitated a smoother troubleshooting experience by breaking down complex problems into manageable tasks, further supported by the targeted information and instructions available through the web resources. This structure helped users navigate the troubleshooting process with greater ease and fostered a deeper understanding of the underlying principles of IoT systems.

Building on this foundation of enhanced problem-solving efficiency, we also witnessed increased user engagement and confidence. The structured approach empowered users with the tools and knowledge to tackle challenges independently. As participants progressed through the troubleshooting tasks, their growing familiarity with the process and the success they experienced in resolving issues contributed to a noticeable boost in their confidence levels. This increased confidence, in turn, led to higher levels of engagement with the tasks at hand. Users became more invested in troubleshooting, demonstrating a willingness to explore and experiment with solutions. This positive feedback loop between efficiency and engagement underscores the importance of designing support systems that address users' immediate troubleshooting needs and contribute to their long-term development and confidence in working with IoT technologies.

Areas for improvement

While the guided process significantly enhanced problem-solving efficiency and user confidence, our study also identified key areas where the support system could be further optimised.

While participants navigated the web resources relatively easily during straightforward troubleshooting, their effectiveness waned when confronted with a deliberately inserted anomaly – switching "D0" for "D7" in the system configuration. This subtle yet critical alteration was designed to test the protocol's capacity to guide users through unexpected problems. The challenge arose not from navigating the resources but from the participants' ability to detect and correct this specific, artificially introduced error. Despite the web resources' aim to offer comprehensive support, this scenario highlighted a need for enhancing the tools' ability to direct users' attention to potential, non-obvious issues within the system. It underscores the importance of designing user-friendly support tools and adept at guiding users through the nuanced detection of subtle errors that may not be immediately apparent.

A significant observation from our study was the tendency for participants to treat the instructional cards and web resources as separate rather than interconnected elements. Once participants navigated to the web resources, the cards were often set aside and not referred back to, breaking the intended continuous loop of guidance. This practice suggests that while the cards served as an effective summary or initial road map, their utility diminished once users engaged deeply with the online content. This separation indicates a missed opportunity for the cards to reinforce learning and support throughout the web-based troubleshooting process. Consequently, this points to an area for improvement in ensuring that the cards and web resources function more cohesively. Designing these tools to more explicitly encourage ongoing reference between the two, particularly by making the cards useful even after users become familiar with the process, could strengthen the support protocol's overall effectiveness.

Additionally, the study brought challenges related to the depth and complexity of content provided through the web resources. While designed to be comprehensive, some users felt overwhelmed by the detailed technical information, which occasionally detracted from the troubleshooting process rather than facilitating it. This outcome points to the delicate balance in tailoring content to meet varied user expertise levels. Ensuring that the information is accessible and manageable for users with diverse backgrounds and skills is crucial. It suggests a potential refinement of content presentation, possibly through adaptive content delivery or more targeted, user-specific pathways, to better meet individual needs and enhance the overall effectiveness of the support tools.

In summarising the insights derived from our user study, it is evident that while the guided process, comprising instructional cards and web resources, offers a solid foundation for enhancing troubleshooting efficiency, significant opportunities remain to refine these tools for broader applicability and effectiveness. The observed challenges, such as the disconnect between using cards and web resources and the difficulties in addressing complex or non-obvious system anomalies, underscore the necessity for a more integrated and intuitive support system. Moving forward, it is crucial to focus on creating support mechanisms that facilitate immediate problem resolution and contribute to the development of end-user developers' expertise in IoT. By prioritising user-centric design and adaptive learning materials, future iterations of support tools can address the diverse needs and challenges developers face in this dynamic field, ultimately fostering a more capable and confident community of IoT practitioners.

7.5.2 | Limitations and Further Research

The study's reliance on twelve participants limits the representation of diverse experiences and perspectives. Future research will benefit from a larger sample size to develop more robust IoT development tools and guidelines, ensuring a broader spectrum of user insights.

A particular challenge of our research was its bespoke design for a specific IoT project, which provided deep, contextual insights but made it difficult to generalise our findings. The protocol, developed with custom-designed instructional cards and web resources, was tailored to meet the unique challenges of this project. This specificity makes it difficult to claim that the protocol, as is, would be universally effective across all IoT troubleshooting scenarios. Recognising this, there is a clear need for future studies to examine the adaptability and scalability of such support protocols in a wider variety of IoT projects. This approach will help to validate their applicability and effectiveness in diverse environments and troubleshooting contexts.

Moreover, conducting the study in a controlled laboratory setting, while beneficial for close observation, might not fully reflect the complexities of real-world behaviour. Future research should, therefore, consider *in-the-wild* studies, where *"the importance placed on the setting and context, conducting research in every day and naturalistic environments"* [149].

Additionally, the potential for integrating AI tools like *ChatGPT* into IoT troubleshooting signifies an exciting frontier for research. Investigating how such technologies can aid end-user developers in identifying and rectifying IoT project errors offers promising avenues to enhance support mechanisms and tool effectiveness in real-world scenarios.

7.6 | Summary

Reflecting on the journey in this chapter, we revisited the fragmented landscape of IoT troubleshooting support. We sought to re-imagine it through a troubleshooting guide, blending instructional cards with web resources. This blend allows for both hands-on guidance and accessible online troubleshooting support. This synthesis aimed to bridge the scattered resources and streamline the troubleshooting experience. Our hypothesis posited that this approach could significantly clarify the often opaque process of IoT troubleshooting, giving end-user developers a clearer path through potential issues. The user study tested this hypothesis, providing insights into the interplay between user needs and the structured support offered.

Our exploration underscores a vital truth: while our protocol offers an organised pathway to assist end-user developers, the vast diversity within IoT systems presents a formidable challenge in anticipating every potential issue. This reality brings us to an essential conclusion – the most effective strategy may not lie solely in attempting to cover all bases but rather in empowering developers to build their expertise. By providing clear, step-by-step guidance, our approach helps developers gain the confidence and foundational skills needed to navigate unexpected challenges and deepen their expertise. Gaining deep, versatile knowledge emerges as a crucial strategy for mastering the complex and continually evolving landscape of IoT troubleshooting, turning technical challenges in IoT troubleshooting into opportunities for skill development and creative problem-solving.

With this final user study, we conclude the *Nucleus* part of the thesis and transition into the *Resolution* phase, where the upcoming chapters will synthesise our findings and outline future directions.

III Resolution

Chapter 8

Discussion

> *Moreover, if we move in the direction of making machines which learn and whose behaviour is modified by experience, we must face the fact that every degree of independence we give the machine is a degree of possible defiance of our wishes.*
>
> **— Norbert Wiener**
> "The Machine Age" [184]

AS WE DISCUSS OUR FINDINGS, this chapter revisits the key insights from our user studies in the realm of IoT for end-user developers. We will discuss how our findings highlight the importance of tailored support for non-expert developers and the challenges complex IoT systems pose. We compare our initial proposals and the developed conceptual framework with the outcomes of the user studies detailed in Chapter 4, Chapter 5, Chapter 6, and Chapter 7. This evaluation explores how our findings align with or differ from our hypotheses and considers the broader implications, limitations, and future research directions.

The structure of the remainder of this chapter is as follows. Section 8.1: Interpretation of Results summarises our main findings, interprets and analyses the results within the context of our research questions, and discusses any unexpected outcomes and their possible explanations. Section 8.2: Comparison with Previous Research compares these results with the existing literature and previous studies, identifying similarities, differences, and emerging trends. It also deliberates on how our findings contribute to or diverge from the established body of knowledge. In Section 8.3: Methodological Reflections, we reflect on the research design, including data collection and sample size limitations, and propose improvements for future work. In Section 8.4: Contributions to the Field, we articulate the contributions of this study to the field, underscoring the novel aspects and innovations introduced, and deliberate on the significance of our research in advancing current knowledge. Section 8.5: Future Research Directions proposes potential avenues for future research, identifies unanswered questions and areas ripe for further exploration, and examines how subsequent studies can constructively build upon the groundwork laid by this work. Section 8.6: Limitations explicitly acknowledges the limitations of our study, discusses their potential impact on the results, and considers alternative approaches or methodologies that might mitigate these constraints. Finally, Section 8.7: Summary summarises the key points presented in the chapter, reinforces the importance of our research within the broader field, and concludes with a concise summary of the primary contributions and implications.

8.1 | Interpretation of Results

Part II – Nucleus of this thesis presents four user studies in Chapters 4 to 7. This section presents an interpretation of the results, chapter by chapter, closing with a thematic synthesis. Starting with an interpretation of Chapter 4, which details a long-term study observing an end-user developer creating an IoT project aiming to comprehend the time investment, project phases, and encountered obstacles, we continue with an analysis of the results from Chapter 5, that discusses a user study where participants use a custom-made deck of conceptual cards to ideate technology-driven solutions. We then move forward to interpret the results from Chapter 6, which presents a user study focusing on the support required by end-user developers to overcome challenges in building electronic circuits and programming computers, and an analysis of the results from Chapter 7, that presents a user study evaluating a proposal to help end-user developers effectively identify and potentially solve issues in their IoT projects. We finalise the section with a thematic synthesis of end-user IoT development.

8.1.1 | Findings from Observing an End-User Developer

In Chapter 4, we present a long-term study observing an end-user developer's journey in creating an IoT project. This study closely aligns with the first scenario described in Chapter 2, where Jasmine Kirstel seeks to create an artefact to track and monitor her desk usage, capturing data such as the height of the desk's tabletop at regular intervals. This investigation is directed towards answering the first research question, as presented in Chapter 1:

> **RQ1** | What barriers do end-user developers face in creating bespoke IoT projects?

We hypothesised that, while not impossible, the complexity of completing a functional IoT project was often a significant challenge for an end-user developer. To test this hypothesis, we embarked on a case study with specific goals:

- To understand the time frame required to create a custom IoT project to an acceptable functional state.
- To comprehend the different phases a non-expert had to pass through to complete their plan.
- To identify obstacles that could have stopped or delayed the project's progress.

This chapter's findings illuminate end-user developers' practical challenges and learning experiences as they navigate the intricate process of creating a functional, customised IoT solution.

Understanding time

The study's first goal was to determine the time required for an end-user developer to achieve a functional IoT solution. Over 344 days, the participant (P2) worked for 148 days, totalling 578 hours (approximately 4 hours per day). Despite the lack of a strict deadline, the study hypothesised that the complexity of assembling an IoT project might be too daunting for a non-technical user. The results indicate that while the participant completed the project within this timeframe, the average non-expert might abandon the project without significant motivation, given the substantial time investment and intermittent work patterns observed.

Understanding phases

The second goal of our study was to understand the different phases a non-expert user undergoes to complete a bespoke IoT solution. Through extensive diary analysis and visualisations, 15 distinct phases of the project were identified, varying in complexity and duration. Initially, the phases were sequential but became more intertwined and complex as the project progressed. Various factors, such as meetings or receipt of components, marked the start or end of phases, but no uniform pattern across all phases was found. Some phases concluded without clear documentation, while others had specific goals achieved, as noted in the diary. This indicates the nonlinear and varied nature of project development.

Understanding obstacles

The third goal was to identify obstacles in IoT projects. Through diary analysis and interviews with two developers of varying expertise, obstacles were identified and categorised into declarative (*"knowledge of"*) and procedural (*"knowledge how"*) knowledge, with the latter aligning with *Threshold Skills*. Seven key skills were outlined: technical vocabulary, methodologies, understanding various technologies and formats, and practical skills like circuit building and programming. Mastery of these skills is crucial, as their complexity can be a blockage. Expertise level affected problem identification and resolution time, suggesting an interdisciplinary approach is necessary, particularly for non-experts.

These findings helped us establish fundamental design principles for IoT based on the participants' behaviours and actions while they engaged in creating the IoT project. They guided the creation of the following user studies: beginning with concept refinement and ideation in Chapter 5; continuing with the exploration of support strategies for overcoming technical challenges in Chapter 6; and concluding with an evaluation of troubleshooting interventions in Chapter 7. Beyond these functional insights, the study also illuminated deeper issues surrounding user persistence and motivation. The observed pattern of intermittent engagement and reliance on external prompts suggests a threshold beyond which frustration or confusion can erode commitment, particularly when expectations about project complexity are unmet or poorly calibrated.

8.1.2 | Findings from Concept Refinement and Ideation

Chapter 5 focused on the crucial phase of concept refinement and ideation, aiming to explore how end-user developers can be effectively supported in generating and developing their ideas. The main objective was to assess the effectiveness of physical cards in facilitating the ideation process for non-experts. This chapter addresses Research Question 2;

> **RQ2** | In what ways can targeted support during the ideation phase improve end-user developers' ability to conceptualise unique technology-driven solutions?

Having created a bespoke deck of cards, the user study aimed to test them to understand how they could help users better define their ideas for a technology-driven project. Using these cards, end users map their ideas, bridging the gap between an abstract concept and something more concrete and achievable.

The key findings from these sessions indicate that physical cards significantly enhanced the ideation process for end users. The tactile nature of the cards, combined with the visual and interactive elements, helped users better articulate and refine their ideas. The pre-conceived cards provided seamless access to additional information and resources, which supported users in understanding complex concepts without overwhelming them.

Using these tools increased user enthusiasm for creating projects. Participants reported that the physical cards made the ideation process more intuitive and enjoyable. Their tactile nature encouraged active engagement and clearer visualisation of concepts, leading to more precise and feasible project ideas. The physicality of the cards also helped sustain users' interest throughout the sessions.

The findings from Chapter 5 underscore the importance of user-friendly, accessible tools in IoT development. Enhancing the ideation process with these tools can lead to feasible IoT projects, promoting broader adoption and effective use of IoT technologies by non-experts. Incorporating physical, interactive tools and supportive digital resources in the ideation phase is crucial. These insights align with the research objectives, providing a solid foundation for future studies to empower end-user developers. While participants credited the cards for enhancing ideation, their progress was often scaffolded by facilitator interventions, which prompted reflection, clarified ambiguities, and sustained engagement. This underscores the tacit but critical role of facilitation in shaping the user experience, even when the tool itself appears to be the primary driver of insight.

8.1.3 | Findings from Project Enabler

In Chapter 6, we present a detailed user study that probes the efficacy of support systems available to end-user developers as they construct functional prototypes for custom IoT projects. This chapter is specifically designed to address the third research question posited in this thesis;

> **RQ3** | What essential support strategies enhance end-user developers' success in building and programming IoT projects?

The study was designed to achieve three critical objectives: to assess developers' assembly capabilities using the enhanced *TILES IoT Toolkit* cards, to evaluate the effectiveness of the tailored step-by-step instructional support, and to explore the transferability of acquired knowledge to new project scenarios. Follow-up discussions revealed varying levels of participant confidence in applying learned concepts to unfamiliar setups, highlighting both progress and limitations in independent skill application. By integrating tangible tools with digital instructions, the investigation provided a comprehensive environment to evaluate users' assembly proficiency and the impact of the support strategies on the adaptability of the skills developed.

Building on these objectives, we explored participants' experiences and found that 11 out of 12 successfully assembled the prototype under strict hardware, software behaviour, and tutorial accuracy conditions. The journey revealed smooth progress brought enjoyment and satisfaction, but obstacles quickly led to frustration, potentially causing project abandonment. A preference for personal assistance over online forums emerged, highlighting the need for direct expert guidance. Despite the challenges, our hypothesis held that end-user developers, with proper support, can create bespoke IoT projects. The complexity of such projects underscores the need for tailored assistance, given the diversity of IoT projects and the lack of specialised guides.

The insights from this chapter demonstrate that end-user developers can successfully complete projects with significant support and suggest future research approaches, such as in-the-wild studies and combining various tutorials to enhance the developer's journey. These findings highlight developers' emotional and cognitive challenges, emphasising the need for practical, empathetic, and adaptive guidance. This sets the stage for our next research phase, presented in Chapter 7, where we evaluated methods to help developers diagnose and fix errors when dealing with real-world problems. The study also revealed a mismatch between users' expectations and the actual effort required to complete tasks. Participants frequently misjudged the difficulty of the work—some found it unexpectedly straightforward, while others found it more demanding than anticipated. This discrepancy underscores the importance of calibrating users' expectations early and providing conceptual scaffolding that helps them understand the required skills and their relevance.

8.1.4 | Findings from Guided Troubleshooting

Chapter 7 builds on the findings presented in Chapter 6, which provided key insights into the challenges encountered by end-user developers in bespoke IoT project creation. These findings highlighted areas where users struggled most, particularly in identifying and resolving technical issues during project development. As a result, Chapter 7 offers a focused investigation into the specific support needs that emerged from these challenges, aiming to decipher the forms of assistance necessary for effective troubleshooting.

The inquiry is guided by the following research question, which is designed to deepen understanding of the assistance required for efficient problem-solving:

RQ4 | What forms of support do end-user developers require to efficiently identify and resolve issues in their IoT projects?

Our research has unearthed a wealth of insights regarding the multifaceted hurdles developers encounter. Emotional and cognitive challenges, educational processes, information utilisation, and technological competence form the cornerstone of our understanding of the support needs and strategies that can enhance the efficiency and efficacy of end-user developers in IoT project development.

Participants faced significant emotional and cognitive challenges, including confusion, uncertainty, and frustration, highlighting the emotionally charged nature of IoT development for non-experts. These findings emphasise the need for intuitive, user-friendly tools and empathetic design to reduce stress and confusion. The investigation in Chapter 7 also revealed diverse cognitive approaches and problem-solving strategies, indicating the necessity for adaptable support systems that cater to various learning styles and promote proactive engagement with technology. Clear, actionable instructions and resources are essential for effective technology use and project progress. Additionally, hands-on experience and practical skills emerged as crucial for overcoming technology-related challenges. Furthermore, findings from the user study in Chapter 7 advocate experiential learning and structured methodologies to cultivate these competencies among end-user developers, reinforcing the value of competence-building approaches in IoT project development.

Despite confirming the hypothesis that end-user developers can navigate IoT challenges with adequate support, the study highlighted the intricacies and specific types of assistance required. The complexities of technological projects, especially for non-experts, can be daunting, and even minor inconsistencies can severely disrupt progress. Furthermore, the study highlighted developers' difficulties when encountering unforeseen issues, emphasising the need for comprehensive, adaptive, and empathetic guidance.

In synthesising the findings from Chapter 7, it is clear that while progress is possible with the proper support, the journey of an end-user developer in IoT is fraught with emotional, cognitive, and technical hurdles. Addressing these challenges through user-centric design, adaptive learning resources, and hands-on experience will be crucial in empowering these individuals to successfully navigate and complete their IoT projects. Although participants had access to explanatory materials and structured documentation, they engaged with them only superficially. Instead, they preferred direct manipulation, verbal guidance, or immediate feedback, revealing a consistent disengagement from abstract resources and highlighting the value of embedded, experiential support mechanisms over traditional didactic formats. As we move forward, the insights from this chapter will undoubtedly shape the trajectory of future research and development in this evolving field.

8.1.5 | A Thematic Synthesis of End-user IoT Development

Across the studies in Chapters 4 to 7, a set of recurring tensions emerged that cut across the boundaries of task, tool, and participant. A critical theme was the *Threshold of Frustration:* participants frequently disengaged when systems behaved unpredictably or when outcomes contradicted their intuitive expectations. In these moments, technical obstacles became affective thresholds. Without strong personal motivation, even minor complications proved sufficient to derail progress. This threshold highlights the fragility of engagement in non-expert contexts and the need for tools and environments that actively mitigate discouragement.

Closely linked to this is the theme of *Perceived Agency and Feedback*. Participants' sense of competence was strongly tied to moments of clear, causal response, such as when a sensor reading triggered an immediate visual cue. These moments, though technically trivial, served as powerful affirmations of progress. They reinforced a perception of intelligibility and control that often outweighed more abstract understandings of system logic. Such feedback loops were central to sustaining momentum, suggesting that fine-grained interactional cues may be as crucial as overarching tool functionality.

A third insight concerns *Tacit Dependence and Missing Vocabulary*. Participants routinely relied on informal support networks—friends, forums, or facilitators—not merely for troubleshooting but to articulate problems they lacked the language to describe. This unacknowledged dependency often masked deeper conceptual gaps, which only surfaced at points of failure. These moments revealed the extent to which successful IoT development rests not only on procedural know-how, but also on possessing a minimal conceptual lexicon—something most participants lacked and could not easily acquire through documentation alone.

Fourth, there was a recurrent *Expectation vs. Effort Misalignment*. Participants frequently misjudged the complexity of IoT tasks, with some underestimating the cognitive and technical demands, and others overestimating the challenge and anticipating failure. These mismatches shaped their attitudes towards the tools and their competence. More importantly, they underscore the pedagogical imperative for toolkit designers: to calibrate onboarding experiences that accurately signal the level of effort required and pre-empt demotivating surprises.

Ultimately, the theme of the *Invisible Role of Facilitation* is a common thread that runs through all studies. Although artefacts such as card decks or toolkits were nominally under evaluation, participants' trajectories were often steered—subtly or decisively—by the presence of the researcher. Facilitation, while not formally part of the intervention, proved crucial in maintaining momentum, framing tasks, and interpreting breakdowns. Yet it remained largely unacknowledged, with participants attributing progress to the artefacts rather than the human support. This highlights a structural blind spot in toolkit evaluation: the tendency to undertheorise the role of facilitation in shaping user experience.

Together, these themes offer a more textured understanding of end-user development—one in which cognitive scaffolding, emotional regulation, and social support are interdependent. They suggest that empowerment is not a property of tools alone, but of relational ecologies that blend materials, feedback, language, and care. Future work must account for these dynamics if it is to design genuinely inclusive and supportive environments for non-expert IoT creators.

8.2 | Comparison with Previous Research

This section compares the results of this research with existing literature and previous studies in the field of IoT development for end-user developers. By reviewing key thematic areas such as end-user development, ideation tools, support systems, and gamification, this comparison highlights both similarities and divergences between existing research and emerging trends in the field. The scenarios presented in Chapter 2 provide simulated contexts that reveal nuanced challenges and opportunities in IoT development, offering points of comparison and contrast with the literature. This comprehensive review provides a detailed perspective on how various methodologies and tools have been employed to facilitate IoT development for non-experts, underscoring gaps in current research. By identifying these gaps, we aim to show how our findings contribute to advancing the field, particularly by empowering end users to create bespoke IoT solutions that integrate seamlessly into their everyday lives and positioning this research within the broader academic landscape.

We begin with Subsection 8.2.1: Development and Support Systems, providing a comprehensive overview of End-User Development, including research on toolkits and supporting systems. Next, in Subsection 8.2.2: Creative Processes and Tools, we discuss tools designed for the ideation phase of a project and examine various creative processes. Finally, Subsection 8.2.3: Motivation and Broader Context addresses the literature on motivational aspects and explores the broader context of supporting end-user developers in the IoT.

8.2.1 | Development and Support Systems

This subsection compares our research in the *Development and Support Systems* domain for Internet of Things (IoT) applications against existing literature. It is divided into two primary subtopics: **End-User Development** and **Toolkits and Support Systems**. The *End-User Development* section examines how our findings diverge from earlier research, particularly regarding the evolving role and capabilities of non-technical users in IoT development. The *Toolkits and Support Systems* subtopic contrasts our work with existing toolkits designed to support IoT development, focusing on the functionalities, usability, and support mechanisms of contemporary toolkits.

End-User Development

The literature on End-User Development (EUD) spans several decades, encompassing many studies and insights. Key themes and findings pertinent to this thesis highlight the importance of user-friendly, accessible tools in supporting end-user developers. Enhancing the ideation process can lead to more innovative and feasible IoT projects, contributing to the broader adoption and effective use of IoT technologies by non-experts.

Research highlights significant cognitive and technical challenges end-user developers face in programming, often struggling with the abstract concepts and detailed intricacies required to create functional software. Costabile et al. [34] explains how end-users inadvertently take on the role of developers when creating and customising software solutions, often resulting in high error rates and identifying a communication gap between end-users and professional developers. Paternò [142] explores the motivations and foundational concepts of EUD, offering a comprehensive review of current approaches and technologies, emphasising the need for ongoing evolution in this field and highlighting the diverse methodologies supporting end-user customisation and the creation of digital artefacts. Blackwell [17] underscores the importance of cognitive models that support non-experts in comprehending programming tasks. A systematic mapping study by Barricelli et al. [11] examined the fields of EUD, End-User Programming (EUP), and End-User Software Engineering (EUSE), revealing a lack of integration among these fields and a shortage of comprehensive frameworks to assist novices. Various tools and frameworks, including visual programming environments, modular toolkits, and drag-and-drop interfaces, have been developed to lower these barriers and simplify development processes, making programming more accessible for non-experts. Addressing these gaps can lead to more effective and inclusive tools for end-user development, helping to cater to the significant differences in skills, motivations, and contexts among end-user developers.

The literature also emphasises the complexities and challenges of IoT-specific projects. M. Burnett and Kulesza [22] stresses the need for user-friendly interfaces to mitigate end users' difficulties with IoT systems, proposing strategies to simplify interactions and improve usability. Dibitonto et al. [60] discusses the evolving IoT landscape and its implications for end-user development, highlighting the need for more accessible programming tools to bridge the gap between advanced IoT technologies and non-experts. Corno, De Russis, and Sáenz [32] identifies technical and usability barriers that hinder effective end-user programming and interaction, advocating for holistic approaches that consider both technological aspects and user experience.

Our research identifies similar challenges in IoT development for non-experts, particularly in creating digital circuits with sensors and actuators and programming their functionality. We emphasise that IoT development's complexity extends beyond programming, including understanding physical components and their interactions, necessitating interdisciplinary knowledge spanning software, hardware, and network communication. While existing tools simplify certain development aspects, they often fail to support seamless integration with physical components—an essential requirement highlighted across the three scenarios in Chapter 2. Our findings reveal that non-experts require more intuitive and comprehensive tools that effectively combine hardware and software elements. Current tools frequently necessitate additional support, indicating a significant gap in toolkits that do not fully meet the needs of IoT end-user developers.

Toolkits and Support Systems

In the fast-evolving Internet of Things (IoT) landscape, innovative tools and support systems are crucial for advancing technology and enhancing user experiences. With IoT's increasing integration into daily life and industry, comprehensive methodologies and resources are essential for design, prototyping, and user engagement. This section explores key contributions from existing literature on IoT toolkits and support systems, which are vital for these advancements. The following paragraphs will present three critical areas: **IoT Ideation and Design Tools**, which examines frameworks and methodologies for fostering creativity in IoT design, highlighting ideation decks and card-based systems for structured brainstorming and problem-solving; **Prototyping and Development in IoT**, focusing on tools and platforms that support iterative creation, emphasising rapid prototyping techniques, modular toolkits, and integrated development environments for easing development processes; and **User-Centred Design and UX in IoT**, which underscores the importance of user-focused IoT systems, addressing the challenges of creating intuitive interfaces and services that cater to diverse needs, while advocating for multidisciplinary approaches for effective and inclusive IoT applications.

The domain of **IoT Ideation and Design Tools** encompasses various frameworks and methodologies aimed at fostering creativity and streamlining the design process for IoT applications. The literature highlights the critical role of toolkits and design systems in enabling multidisciplinary teams to conceptualise and develop innovative IoT solutions. Notable contributions include the *MappingTheIoT Toolkit* by Vitali and Arquilla [177]. This open-source resource supports IoT products' structured and creative design across various phases, from initial research to refined development. This toolkit facilitates collaboration among diverse teams and aligns with established design models like the Double Diamond (Figure 5.1), making it a valuable reference for methodological IoT design tools. Our findings from Chapter 4 provide additional insights into the initial stages of IoT project conception, offering an extended view on how end-user developers engage with ideation tools and highlighting their need for structured support in creative ideation and problem definition. The empirical evidence presented in Chapter 5 demonstrates the practical application and effectiveness of ideation tools in refining technological concepts in real-world scenarios, emphasising the importance of integrating user-centric feedback and iterative refinement into the design of these tools.

Further contributions to the field include the *Tiles Cards* by Mora et al. [128], which support non-experts in generating ideas for augmented objects through a set of 110 design cards and workshop techniques, the *IoT Design Kit* by De Roeck et al. [48], offering modular and flexible tools for IoT product development, and the *Karakuri card deck* by Muñoz et al. [133], which facilitates IoT ideation in the context of manufacturing. These resources emphasise creative thinking and critical evaluation, facilitating the transition from theoretical concepts to practical applications.

Our findings reveal these tools' practical challenges and applications, illustrating how they bridge the gap between theoretical frameworks and practical implementation. The analysis in Chapter 5 underscores the value of a more hands-on, adaptable approach to ideation, critical for effectively supporting IoT design's complex and evolving landscape. Extending the *TILES toolkit*[128], the user study presented in Chapter 6, bridges the gap between ideation and prototyping, aiming at covering more areas of the whole process of an IoT project.

The realm of **Prototyping and Development in IoT** is integral to transforming conceptual ideas into tangible applications. The existing literature highlights various tools and frameworks that simplify IoT prototyping and development processes. For instance, the *"Game of Internet of Things"* (IoTgo) toolkit, as described by Rizvi [148], adopts a playful approach to assist non-experts, particularly teenagers, in creating IoT smart devices. This toolkit facilitates exploration, ideation, programming, and prototyping, making IoT development more approachable by bridging the technical skill gap. Similarly, the Tiles inventor toolbox by Mora et al. [127] provides a robust platform for non-experts to develop interactive objects through physical input/output primitives and extensible hardware modules. The RapIoT software toolkit, detailed by F. Gianni et al. [82], abstracts communication complexities, allowing developers to focus on application logic, thereby enabling collaborative applications across multiple smart objects.

Contrasting these frameworks with our findings, Chapter 5 provides practical insights into the transition from ideation to prototyping, underscoring the critical need for support systems that help users refine and develop their concepts into tangible prototypes. While ideation tools are crucial, our research highlights that transitioning to physical development requires tools that effectively bridge the gap between abstract ideas and practical implementation. Chapter 6 further explores the tools and support systems that facilitate the assembly of IoT projects, emphasising the necessity for modular and accessible tools that support integration stages, ensuring users can successfully bring their projects to fruition. Additionally, our findings from Chapter 7 underscore the importance of guided troubleshooting during development, revealing how users identify and resolve issues. This emphasises the need for development tools that incorporate effective troubleshooting mechanisms, an aspect often under-explored in existing literature, which tends to focus primarily on the initial stages of development. Our research advocates for a comprehensive approach to IoT prototyping and development, integrating practical support throughout the lifecycle to ensure users are well-equipped to handle challenges and refine their solutions, leading to more robust and user-friendly IoT applications.

The literature points to the necessity of modular and adaptive prototyping frameworks, as illustrated by F. Gianni et al. [81], which support the integration and augmentation of everyday objects, making IoT prototyping accessible and straightforward. However, our research in Chapter 7 provides a critical perspective on iterative problem-solving, advocating for development tools that support IoT solutions' creation and iterative refinement. This approach ensures that the tools we develop are user-friendly, flexible, and comprehensive enough to bridge the gap between ideation and practical implementation, ultimately leading to the development of complex, interactive IoT applications that are accessible to non-experts and effective in real-world scenarios. This holistic view facilitates initial development and provides ongoing support for troubleshooting and refinement, thereby ensuring the successful realisation of IoT projects in diverse and practical contexts.

In Chapter 2, particularly in *Scenario B: Family-focused Data* and *Scenario C: Citizen Science,* we highlighted the challenges of scaling IoT systems to accommodate group dynamics and larger community settings. As the complexity of coordination and group dynamics increases in communal contexts, these scenarios reveal gaps in existing IoT systems, particularly their limited capacity to support collaborative and community-based projects' unique coordination and technological needs. This contrast underscores the need for future research that explores sociological and technological support mechanisms tailored for larger-scale, communal IoT implementations. By addressing these gaps, future frameworks could facilitate more effective IoT adoption in real-world scenarios, ultimately creating adaptable and supportive tools across both individual and collective use cases.

The sub-topic of **User-Centred Design and UX in IoT** addresses the intricate challenges and methodologies associated with creating intuitive and user-friendly IoT interfaces and services. The existing literature underscores the complexity of this task and the necessity for multidisciplinary approaches to cater to diverse user needs. For example, F. V. Gianni [83] highlights how IoT development often requires users to adapt to machine-centric languages and complex processes, posing significant barriers for non-experts. F. V. Gianni's work centres on simplifying the development process using user-centred design and HCI principles, particularly in the context of smart city learning – where citizens engage with city data and services to foster awareness and lifelong learning. Gianni aims to reduce barriers and enhance user engagement through iterative prototyping and design science methodologies. Complementing this, Paternò and Santoro [144] discusses the challenges of end-user development (EUD) in IoT, emphasising the need for environments that support dynamic interactions among users, devices, and contexts. Their work outlines a design space for EUD, advocating for tools that democratise IoT development by making it accessible to non-programmers, which is further explored through the IoT Design Deck by Dibitonto et al. [61], facilitating multidisciplinary collaboration in creating connected products with a service-oriented approach.

In contrast, our research from Chapter 4 provides an in-depth perspective over time on the user experiences of end-user developers, revealing the practical challenges they face when interacting with IoT systems. This chapter underscores the critical need for designing tools that facilitate user interaction and adapt to evolving requirements and constraints, emphasising a user-centric approach. Additionally, Chapter 6 explores user needs during project assembly, highlighting the importance of developing tools that guide users through the complexities of assembling IoT projects, enhancing the overall user experience.

Dibitonto et al. [60] propose a co-design method to address the new paradigm of human-computer interaction introduced by IoT, advocating for seamless user experiences in IoT products. Their method promotes collaboration among multidisciplinary teams and encourages prototyping and user testing to create intuitive, user-centric IoT solutions. Unlike this approach, our work focuses on novice developers who typically work in isolation, without access to multidisciplinary teams or extensive resources. The scenarios presented in Chapter 2, particularly *Scenario A: Individual* (Subsection 2.1.1) and *Scenario B: Family-focused Data* (Subsection 2.1.2), involve individuals or small family groups who rely on self-guided learning and minimal external support. In these cases, the co-design approach proposed by Dibitonto et al. [60], which is tailored for professional and collaborative environments, would not be practical. However, this approach may be more applicable in contexts where participants have specific expertise and work within a collaborative framework, such as in *Scenario C: Citizen Science* (Subsection 2.1.3). In contrast, our investigation focuses on understanding how to empower independent developers, helping them independently navigate the complexities of IoT development.

As detailed in Chapter 7, our findings emphasise the importance of user-friendly troubleshooting tools and protocols to ensure effective problem resolution and seamless user interaction with IoT applications. Together, these insights advocate for a comprehensive approach to user-centred design in IoT, integrating practical perspectives to develop technologically advanced and engaging systems for users, thereby enhancing usability across various contexts.

8.2.2 | Creative Processes and Tools

This section presents a comparative analysis of the methodologies and instruments employed in our research against existing literature, organised into two key subtopics: **Ideation and Conceptual Tools** and **Creative Processes and Exploration**. The first subtopic, *Ideation and Conceptual Tools*, compares the ideation processes and conceptual tools utilised in our study with those documented in prior research. It highlights the evolution of ideation frameworks and the effectiveness of tools like brainstorming cards, structured ideation techniques, and digital platforms in fostering creativity and innovation within the IoT domain. We examine how these tools facilitate the generation of novel ideas and compare their impact on the creative process across various contexts and user groups. The second subtopic, *Creative Processes and Exploration*, investigates how the creative processes and exploratory methods applied in our research align with or differ from those identified in previous studies. It critically analyses methodologies such as design thinking, rapid prototyping, and iterative exploration, discussing their roles in encouraging experimentation and innovation. By contrasting our approaches with established practices, this subsection offers a comprehensive understanding of the dynamic interplay between creative processes and tool usage in IoT design and development.

Ideation and Conceptual Tools

The ideation process is pivotal for driving innovation and creative problem-solving within HCI and IoT. This subsection examines the landscape of ideation and conceptual tools, particularly their application in IoT development and design. We examined three primary areas: **Card-Based Ideation Tools**, exploring their role in fostering creativity and structured brainstorming through engaging, tactile methods; **Evaluation and Development of Ideation Tools**, assessing the effectiveness of various tools in aiding designers and end users in conceptualising connected products; and **IoT-Specific Ideation Tools**, focusing on tools tailored to the unique challenges of IoT product development, aiding in the transition from abstract concepts to tangible prototypes. This exploration aims to provide a comprehensive understanding of the current landscape, highlighting how the innovative approaches in this thesis, including our bespoke ideation deck for end-user developers discussed in Chapter 5: Concept Refinement and Ideation, align with and contribute to these established domains.

The work presented in Chapter 5 aligns closely with established principles of card-based ideation tools, as highlighted in the literature. We introduced a structured, card-based system aimed at reducing ambiguity and assisting end-user developers in defining and refining their project ideas. This approach is consistent with the concept of *Ideation Decks*, which facilitate tackling specific design problems within a broader context through parallel design practices and iterative explorations Golembewski and Selby [85]. The use of conceptual cards in our study supports creative dialogue. It provides a tangible framework for idea generation, echoing the findings of

Hornecker [91] on the utility of card games in promoting creative discussions.

Furthermore, the *Envisioning Cards* toolkit, discussed by Friedman and Hendry [75], emphasises integrating human values in the design process, which our work also aims to achieve by encouraging a user-centred approach to mobile app development. Our contribution leverages the benefits of card-based tools for ideation, focusing on practical applications in mobile technology and underscoring their role in bridging the gap between abstract concepts and tangible project outcomes. This is particularly evident in the comprehensive design of the deck, which categorises cards into themes like technology components and user interactions, enabling users to explore various project configurations visually and tangibly.

Real-world contexts, such as those illustrated by *Scenario B: Family-focused Data* in Chapter 2, reveal how diverse learning styles and user interactions shape the ideation process in family-oriented or small-group IoT projects. While existing ideation toolkits can be applied in this context, family-oriented projects often require flexible and adaptable tools that can accommodate the varied creative processes of different users—such as accommodating individual preferences, learning speeds, or levels of familiarity with technology. This need for flexibility resonates with findings in both our study and the broader literature, which stress the importance of adaptable frameworks in group-based settings.

In evaluating and developing ideation tools, our work provides a detailed account of the iterative design process used to create the conceptual card deck, aligning with the methodologies discussed by De Roeck et al. [47], who emphasise the integration of physical and digital elements in product design. Chapter 5 identifies challenges in the ideation phase for end-user developers and introduces a structured tool to address them, validating its effectiveness through a user study. This approach mirrors the development and testing of the *Mapping The IoT Toolkit*, presented by Vitali and Arquilla [177], which also underscores the importance of iterative refinement and user feedback. Our user study focuses on evaluating the efficacy of the card deck through empirical data collection, including participant feedback and observations. The insights gathered validate the hypothesis that structured support enhances the ideation process, providing valuable directions for future refinement. While the work in Chapter 5 focuses on mobile applications, its principles and methodologies are relevant to IoT product ideation. The *IoT Design Deck*, discussed by Dibitonto et al. [61], and the IoT Design Kit by De Roeck et al. [48], emphasise the need for tools that support the complex, multidisciplinary nature of IoT design.

Creative Processes and Exploration

This section presents a comparative analysis of the "Creative Processes and Exploration" involved in IoT development for end-user developers, integrating findings from this research with existing literature. A significant focus is on the facilitation of creative ideation and concept development. Existing studies, such as those by Golembewski and Selby [85] and Friedman and Hendry [75], demonstrate the effectiveness of ideation decks and Envisioning Cards in aiding non-experts to generate and refine ideas. These tools offer a structured and engaging approach to brainstorming and problem-solving, which is crucial for transforming abstract ideas into tangible project plans. The findings from Chapter 5, where a bespoke card deck was developed to support end-user developers in mobile application ideation, illustrate how such structured systems can significantly reduce ambiguity and enhance the creative process. This deck, similar to the ideation tools discussed in the literature, highlights the importance of a tactile, visual approach in facilitating concept development and underscores the value of engaging methodologies in fostering creativity.

The simplification of prototyping and development processes is another critical aspect, with existing research by Rizvi [148] and Mora et al. [127] showcasing the role of accessible and user-friendly toolkits in bridging the gap between high-level concepts and functional IoT prototypes. These studies resonate with the outcomes in Chapter 6, which discusses the extended *TILES IoT Toolkit* designed to assist non-expert developers in IoT project assembly and programming. The toolkit's approach, encompassing detailed instructions and visual aids, aligns with the iterative design and feedback processes emphasised by Vitali and Arquilla [177] in their work on the Mapping The IoT Toolkit. This alignment highlights the effectiveness of providing structured and supportive environments for end-user developers, making IoT prototyping more accessible and manageable.

Furthermore, the necessity for guided troubleshooting and intuitive support systems is underscored in both the thesis findings and the existing literature. Studies by Dibitonto et al. [60] and De Roeck et al. [47] emphasise the importance of integrating effective troubleshooting mechanisms into ideation tools to assist users in resolving issues during development. This is corroborated by the findings in Chapter 7, where a structured troubleshooting protocol was developed to help non-experts navigate technical challenges. The practical implications of these studies and the research presented in the thesis highlight the necessity for clear, step-by-step guidance to maintain user engagement and ensure successful project outcomes. The thesis and existing literature demonstrate the potential for empowering end-user developers to innovate effectively within the IoT domain by providing comprehensive support systems that facilitate problem-solving and iterative development.

8.2.3 | Motivation and Broader Context

This section explores the pivotal role of gamification and motivational strategies in IoT development for end-user developers, comparing our findings with existing literature to evaluate their effectiveness. In the *Gamification and Motivation* sub-section, we assess how gamified elements and motivational frameworks employed in this research align with or diverge from past studies, highlighting their impact on user engagement and creativity. The subsequent sub-section, *Broader Context and Additional Insights*, situates the findings within a wider landscape, discussing the broader implications and novel insights that extend beyond the immediate scope of the study. This comprehensive analysis not only underscores the relevance of motivational strategies in fostering user involvement but also contextualises the study's contributions within the broader field, providing a deeper understanding of the challenges and opportunities in IoT development for non-expert users.

Gamification and Motivation

Gamification and motivational strategies play a critical role in enhancing the engagement and creativity of end-user developers, particularly in the context of IoT project development. Existing literature, such as that by Deterding [56], highlights the effectiveness of gamification in fostering motivation and user participation through elements like points, badges, and leader-boards. In Chapter 4, the long-term study reveals the challenges non-expert developers face in maintaining motivation throughout the IoT project lifecycle. The findings underscore the importance of incorporating gamified elements to sustain user engagement and reduce the perceived complexity of IoT projects. Similarly, Hornecker [91] discusses how gamified card-based tools can transform abstract ideation into structured and enjoyable activities, which aligns with the methods employed in Chapter 5 to facilitate concept refinement and ideation through a bespoke card deck designed to support creative processes.

In Chapter 6 and Chapter 7, gamification is a key strategy to simplify prototyping and troubleshooting for end-user developers. Chapter 6 introduces an extended version of the *TILES IoT Toolkit*, which incorporates gamified elements such as QR-coded instructional cards that link to interactive online resources. This approach echoes the principles discussed by Mora et al. [128], who emphasise the role of gamification in making IoT development more accessible and engaging for non-experts. Furthermore, Chapter 7 focuses on guided troubleshooting, highlighting the effectiveness of gamified support systems in helping users navigate technical challenges. This is consistent with the findings of Rizvi [148], who advocate for the use of gamified toolkits like *IoTgo* to support end-user development in IoT. By integrating gamification and motivational strategies, these chapters enhance user experience and align with broader research, indicating that such approaches are crucial for successfully empowering non-expert users to engage with complex IoT technologies.

The three scenarios in Chapter 2 illustrate distinct motivational contexts—individual, family, and community settings—each shaped by unique social dynamics. In multi-person scenarios, social factors such as peer support, shared goals, and accountability significantly influence a project's progression. For example, individual projects (Scenario A) are driven primarily by personal goals like health or productivity. On the other hand, family-focused projects (Scenario B) are motivated by shared interests, benefiting from collaborative challenges and collective rewards that foster family bonds and mutual engagement. In community settings (Scenario C), motivations often arise from communal goals or local concerns, with public recognition and shared achievements playing a central role. This diversity underscores the need for motivational strategies that align with the specific social context, contrasting with the individual-centred approaches more predominant in existing literature. Recognizing these nuances is crucial for designing support systems that engage users effectively across diverse IoT project scales, ensuring that motivational strategies are contextually relevant and resonant.

Broader Context and Additional Insights

The findings from Chapters 4 to 7 provide valuable insights into the broader context of IoT development for end-user developers, particularly concerning motivational strategies. These include tangible elements like cards, interactive tools, and subtle aspects of gamification, such as progression systems and celebratory feedback (e.g., *"Time to celebrate!"*) when milestones are reached. The practical challenges identified in Chapter 4, such as sustaining engagement and overcoming technical barriers, underscore the need for dynamic and supportive environments. This is consistent with the insights provided by Paternò and Santoro [144], who emphasise the importance of designing inclusive systems that cater to a wide range of user experiences and capabilities. By aligning the study's findings with the principles of user empowerment discussed by Blackwell [17], the thesis highlights the critical role of tailored, user-centric approaches in fostering motivation and engagement among non-expert developers. The structured card deck introduced in Chapter 5 provides a practical example of how gamified tools can enhance creative processes, reflecting broader trends in HCI research that advocate for the integration of playful elements to facilitate learning and ideation, as discussed by Vitali and Arquilla [177].

Furthermore, the iterative development and evaluation of the *TILES IoT Toolkit* in Chapter 6 align with the broader discussions on the importance of providing robust support systems for end-user developers. F. V. Gianni [83] emphasise the need for comprehensive toolkits that simplify technical processes and enhance user motivation through gamified interactions. The study's focus on guided troubleshooting in Chapter 7 extends the discourse by highlighting the necessity of clear, user-friendly protocols to support problem-solving, a point underscored by Desolda et al. [53]. This chapter's findings resonate with the broader context provided by Gennari et al. [79], who discuss integrating motivational strategies into support tools to maintain user engagement and enhance the learning experience. Overall, our work integrates these broader insights by demonstrating how gamification and motivational strategies can empower end-user developers, bridging gaps in existing research and providing practical solutions for real-world applications in IoT development.

In conclusion, the comparison with previous research underscores the significant strides made in this study towards understanding and supporting end-user developers in IoT projects. By integrating gamification and motivational strategies, the research addresses the immediate challenges of non-expert developers and contributes to a broader discourse on enhancing user engagement and creativity. The findings highlight the critical need for comprehensive, user-friendly toolkits that bridge the gap between conceptual ideation and practical implementation, echoing the themes identified in previous studies. Moving forward, this research lays the groundwork for developing more integrated and accessible support systems that empower diverse users to engage with complex IoT technologies effectively. This holistic approach, which combines theoretical insights with practical applications, provides a robust foundation for future advancements in IoT development for end-user developers.

8.3 | Methodological Reflections

The methodology utilised in the user studies presented in Chapter 4, Chapter 5, Chapter 6, and Chapter 7 is multifaceted and encompasses various stages, participants, and data collection methods. This presents strengths, weaknesses, and limitations, which we will detail below. We also present suggestions for improvements that can be made in the following studies in the area.

8.3.1 | Strengths

The methodology employed in the user studies exhibits several commendable strengths, as outlined below:

- **Variety of methods:** The combination of a long-term study, surveys, prototyping, and semi-structured interviews offers a rich, multi-dimensional understanding of the participants' experiences and challenges.
- **Iterative design:** In some cases (e.g., the long-term study presented in Chapter 4), the methodology allows for a preliminary study to inform the main study, a good practice in design-oriented research. This iterative approach can help refine the research tools and questions based on early findings.

8.3.2 | Weaknesses

Despite the methodological strengths, we identified certain weaknesses, which are detailed below:

- **Participant selection:** Due to limited access to many interested potential participants, selecting participants was convenient rather than systematic. While this approach is common in exploratory studies, it may not provide a representative sample of end-user developers.
- **Limited sample size:** The studies involve a small number of participants. While this allows for in-depth analysis, it may not capture the full range of experiences and challenges end-user developers face.
- **Complexity in documentation:** The methods require participants to engage with various tools and *ad hoc* documentation, which might be challenging for some as they do not possess any previous experience, and the new environment is intimidating. This lack of familiarity can lead to difficulties in understanding and using the tools effectively, potentially impacting the quality and authenticity of the data collected. Additionally, participants might opt to say they understand things they do not to avoid feeling dumb in the context of the study, further complicating the documentation process and skewing the study's results.

8.3.3 | Limitations

Our research also presented limitations that should be acknowledged, as outlined in the following points:

- **Potential bias:** Participants, aware of being observed as part of the study, might alter behaviours—such as engagement levels or task persistence—which could affect the authenticity of the data collected.

- **Technical support:** The level of technical support provided during the study may not reflect the limited assistance available in real-world settings, potentially influencing participants' success in completing tasks and their overall experience.

- **Individual-oriented Studies:** A key limitation of our research is its individual-oriented focus, which may overlook the complex social dynamics—such as collaboration, peer influence, and shared accountability—present in family or community IoT projects, as outlined in Chapter 2. These dynamics can introduce unique challenges and motivations that are essential for understanding multi-person settings in IoT development.

8.3.4 | Suggestions for Improvements

To address the identified weaknesses and limitations, the following suggestions are proposed for future research enhancements:

- **Increase sample size and diversity:** Future research should aim for a larger and more diverse participant pool. Expanding beyond the university community and including participants with varying levels of experience and diverse backgrounds would enhance generalisability and provide broader insights into the needs of different user demographics.

- **Longitudinal follow-up:** Conduct follow-up studies with the same participants over an extended period to understand the long-term impact and sustained use of the tools and methods introduced, examining how they adapt to or continue using these tools in their ongoing IoT projects.

- **Controlled vs. natural setting:** Compare findings from controlled environments (such as a lab) with in-the-wild studies in natural settings (e.g., participants' homes or workplaces) to assess how different environments might influence the IoT development process and the practical applicability of the tools and methods.

- **Real-world context:** Conduct studies with actual end-user developers working on real projects, as this could offer authentic insights into the challenges, problem-solving strategies, and successes that emerge in real-world IoT development scenarios.

- **Blind trials:** To reduce potential bias from participants' awareness of being studied, consider implementing blind trials where participants are unaware of the specific research focus or believe they are part of a different study, thereby fostering more natural behaviours and reactions.

- **Group-driven projects:** Future research could benefit from examining group-driven projects, as these studies would provide deeper insights into the motivational, collaborative, and organisational challenges unique to multi-person IoT development settings—as the scenarios presented in Chapter 2. This focus could reveal how social dynamics such as peer influence, collective decision-making, and shared accountability impact the project outcomes.

While the methodology used in the user studies has its strengths in providing a detailed, nuanced understanding of the end-user developers' journey, there are notable limitations, particularly concerning sample size and diversity. Future research should aim to address these limitations while building on the strengths of the multi-method approach to provide even more comprehensive insights into the challenges and needs of end-user developers in IoT.

8.4 | Contributions to the Field

The research presented in Chapters 4 to 7 advances the field of IoT development for end-user developers by addressing critical challenges and introducing innovative tools and methodologies. Chapter 4 highlights the practical challenges end-user developers face, such as the complexity of integrating hardware and software components and the need for sustained motivation. By focusing on these real-world obstacles, the study provides valuable insights into the specific needs of non-expert developers. It underscores the importance of creating intuitive and accessible tools that facilitate the entire development lifecycle from ideation to implementation.

A notable aspect of this research involves the introduction of gamified ideation, prototyping and troubleshooting processes, designed to enhance user engagement and creativity. Chapter 5 introduces a bespoke card deck to assist end-user developers in the ideation phase, making abstract concepts tangible and easier to explore. This gamified approach fosters creativity and reduces the cognitive load associated with the initial stages of IoT project development. Furthermore, the iterative development of the extended version of the *TILES IoT Toolkit*, as detailed in Chapter 6, incorporates gamified elements to simplify the prototyping process, thus bridging the gap between high-level ideas and functional prototypes. These innovations demonstrate a significant shift towards user-friendly design in IoT development tools, highlighting the potential for gamification to lower barriers to entry for non-experts.

The significance of our research in advancing current knowledge is evident in its comprehensive approach to problem-solving in the IoT domain. Chapter 7 focuses on guided troubleshooting, presenting a protocol that integrates clear, step-by-step instructions to assist non-expert developers in resolving technical issues. This approach is critical in ensuring the successful deployment of IoT projects, as it addresses one of the major pain points for end-user developers: the complexity of troubleshooting and debugging. By providing a structured support system that combines theoretical insights with practical applications, this research not only enhances the usability of IoT tools but also contributes to a deeper understanding of how to support non-expert users in complex technological environments effectively.

The research in this thesis contributes significantly to the field by offering innovative solutions that address the unique challenges end-user developers face in IoT. Developing gamified and user-centric tools empowers non-experts to engage more deeply with IoT technologies. It sets a new standard for future research and development in this area. By advancing our understanding of how to support and motivate end-user developers effectively, this study provides a robust foundation for creating more inclusive and accessible IoT ecosystems, ultimately driving innovation and enabling a broader range of users to participate in the digital transformation.

8.5 | Future Research Directions

Future research in IoT development for end-user developers could build upon the foundational insights presented in this thesis by exploring new avenues that enhance the accessibility and effectiveness of support systems for non-experts. One promising area for future investigation is integrating artificial intelligence (AI) into IoT development tools. By leveraging AI, future tools could provide more personalised and adaptive support, tailoring guidance and troubleshooting assistance to individual users' unique needs and skill levels. For instance, AI assistance can enhance the ideation process, helping users brainstorm and refine innovative ideas by suggesting potential solutions and improvements based on a vast repository of knowledge and patterns. Additionally, AI could offer copilot support for developing and troubleshooting IoT software, assisting users in real time with code suggestions, error detection, and problem resolution. This could significantly reduce the learning curve associated with IoT development and make advanced technological capabilities more accessible to a broader audience.

A critical area requiring further exploration is the long-term impact of gamification on user engagement and learning outcomes in IoT development. While Chapter 5 and Chapter 6 demonstrate the immediate benefits of gamified tools in enhancing creativity and motivation, it is essential to conduct follow-up studies to assess the sustained impact of these strategies over time. As end-user developers continue using the gamified environment, they keep learning and becoming more efficient and effective in creating and implementing new solutions. This ongoing engagement can lead to more profound skill development, increased confidence in tackling complex IoT projects, and better integrating innovative approaches into their workflows. Future research could investigate how ongoing engagement with gamified tools influences the development of technical skills, problem-solving abilities, and overall project success rates among end-user developers. Additionally, exploring the balance between fun and functional elements in these tools could provide deeper insights into optimising their design for maximum educational benefit.

Another area ripe for future research is the development of comprehensive frameworks that integrate hardware, software, and network components into a seamless user experience for IoT development. As highlighted in Chapter 7, non-expert developers often struggle with the complexity of managing these diverse elements. Future studies could focus on creating unified platforms that streamline the integration process, enabling end users to conceptualise, prototype, and implement IoT solutions more easily. Investigating the role of augmented and virtual reality in providing immersive, hands-on learning experiences could also open new possibilities for enhancing the practical understanding of IoT systems among non-experts.

Future research should also address the socio-technical aspects of IoT development, particularly in terms of privacy, security, and ethical implications. While the current research has made significant strides in making IoT technologies more accessible, there is a need to explore how these technologies can be designed to safeguard user data and ensure ethical use. Studies that examine user perceptions of privacy and security in IoT development and how these concerns influence the adoption and use of IoT tools could provide valuable insights for creating more trusted and secure IoT ecosystems. By building upon the current work, future research can contribute to developing more inclusive, ethical, and secure IoT environments that empower all users to participate confidently in the digital landscape.

8.6 | Limitations

Despite the significant contributions made by this research, it is important to acknowledge several limitations that may have influenced the findings presented in this thesis. One major limitation is the reliance on a relatively small sample size for user studies, which may not fully capture the diversity of end-user experiences and perspectives in IoT development. Future research should aim to include a more diverse participant pool to ensure that the findings are representative of a broader user base and to explore how different user demographics interact with IoT tools.

Another limitation lies in the focus on specific types of IoT projects and the tools developed for them (e.g., a standing desk using a distance sensor to detect its position). The research in Chapter 6 and Chapter 7 was centred around the extended version of the *TILES IoT Toolkit* and a bespoke set of cards correspondingly, which were tailored to certain aspects of IoT development. While these tools were effective within the scope of the study, they may not address all the challenges end-user developers face in different contexts or with varying project requirements. The specificity of the tools and the contexts in which they were tested might have influenced the outcomes, potentially overlooking broader challenges and opportunities in the IoT landscape. Exploring a wider range of tools and project types could provide a more comprehensive understanding of the needs and challenges of end-user developers.

The studies also primarily focused on the initial stages of IoT project development, such as ideation, prototyping, and troubleshooting as detailed in Chapter 5, Chapter 6 and Chapter 7. This emphasis means less attention was given to later stages, such as maintenance, and scaling of IoT solutions. These stages present unique challenges and opportunities for end-user developers, and further research is needed to address these areas comprehensively. Additionally, while valuable, the studies' reliance on user feedback and observational data may introduce biases that affect the interpretation of results.

Future research could adopt a more holistic approach to address these limitations by incorporating studies that track end-user developers throughout the IoT project lifecycle. This would provide deeper insights into how users' needs and challenges evolve and how they interact with IoT tools at different stages of development. By acknowledging and addressing these limitations, future studies can build on the current work to develop more effective and inclusive support systems for end-user developers in the rapidly evolving field of IoT.

8.7 | Summary

In this chapter, we have critically analysed the findings from our user studies on IoT development for end-user developers, comparing them with existing literature and exploring their broader implications. We began by summarising the primary outcomes of our research, highlighting how our approach has provided valuable insights into the needs and challenges of non-expert developers. We identified critical gaps in existing support systems for non-expert developers, particularly in the areas of building and troubleshooting. We proposed tools and methodologies to address these gaps, including tangible components like information cards connected to online resources and guided, step-by-step protocols discussed in Chapters 4 through 7. The broader implications of these findings suggest that by integrating user-centric and gamified approaches, future IoT systems can become not only more accessible but also more engaging, fostering more significant innovation and collaboration across different user groups.

The comparative analysis with existing research in Section 8.2: Comparison with Previous Research reinforced the unique contributions of this study. Our work extends the existing body of knowledge by thoroughly examining how gamification and motivational strategies can enhance user engagement and creativity in IoT projects. We demonstrated that while current tools often simplify individual aspects of IoT development, a more integrated approach is needed to support non-expert developers comprehensively.

In the broader context, our research highlights the critical need for innovative and accessible IoT development tools that cater to a diverse user base. The novel aspects of our study, including the bespoke card deck and the extended *TILES IoT Toolkit*, provide new pathways for end-user developers to engage with IoT technologies effectively. These contributions significantly promote a more inclusive approach to IoT development, empowering a more comprehensive range of users to participate in the digital transformation. The insights from this research pave the way for future studies to explore and refine these tools, ensuring that IoT technologies become more accessible and user-friendly for non-experts.

The main contribution of this research is its comprehensive approach to supporting end-user developers through interactive and user-centric design principles. This study addresses non-experts' immediate challenges by incorporating hands-on tools like conceptual cards and focusing on user-friendly design. It sets the stage for future innovations in IoT development. The implications of these findings extend beyond the immediate scope of this research, offering valuable guidance for developing more effective and inclusive support systems that can drive the broader adoption and success of IoT technologies in various contexts.

Chapter 9

Conclusion

> *Objectivity is the subject's delusion*
> *that observing can be made without him.*
>
> **— Heinz von Foerster**
> as cited in Schaefer, *The Certainty of Uncertainty* [159]

THIS CHAPTER SUMMARISES THE THESIS by presenting overarching insights and emphasising the significance of the work. It reviews the research questions, synthesises key findings, and reiterates contributions to the field. It also discusses the broader technological and societal implications of our research.

The chapter unfolds the following sections: Section 9.1: Summary of Key Findings provides an overview of the essential contributions from each chapter, illustrating how our research addresses the key questions and enhances the landscape of IoT for end-user developers. Section 9.2: Contribution to Knowledge presents the main contributions of this research to supporting end-user developers in IoT projects within the Human-Computer Interaction (HCI) domain. We discuss key advancements, including empirical and theoretical insights, practical contributions, and the introduction of new tools and methodologies. Section 9.3: Limitations and Further Research outline directions for further research that emerge from this study's findings. We identify key areas that require more in-depth exploration to better support end-user developers in IoT projects and suggest specific topics for future investigation. Section 9.4: Final Remarks presents our final remarks.

9.1 | Summary of Key Findings

This section aims to articulate the primary insights and findings from each chapter of this thesis. The overarching goal of this research was to explore the complexities of IoT development for end-user developers and to identify effective support strategies to facilitate the realisation of bespoke projects.

In the beginning, this research aimed to address the following central question:

> **Q** | Why is it challenging for end-user developers to develop simple IoT systems despite the availability and affordability of technology?

Supporting this central question were several sub-questions:

> **RQ1** | What barriers do end-user developers face in creating bespoke IoT projects?
>
> **RQ2** | In what ways can targeted support during the ideation phase improve end-user developers' ability to conceptualise unique technology-driven solutions?
>
> **RQ3** | What essential support strategies enhance end-user developers' success in building and programming IoT projects?
>
> **RQ4** | What forms of support do end-user developers require to efficiently identify and resolve issues in their IoT projects?

Having restated the primary research question and its supporting sub-questions, we now summarise each chapter's key insights and findings. These summaries highlight the main contributions and outcomes of the research, providing a comprehensive overview of how each chapter addresses the research questions and advances our understanding of IoT development for end-user developers.

9.1.1 | Summary of Each Chapter

This section provides a concise summary of the key insights and findings from each thesis chapter. The following paragraphs present an overview of the research context, methodologies, and significant contributions detailed in Chapters 1 through 8, highlighting the primary conclusions drawn from each study segment.

- **Chapter 1: Introduction**
  The introduction presented a comprehensive overview of the IoT evolution and its relevance to end-user developers. It established the research context, motivation, and hypothesis, emphasising the need to simplify IoT development to reach a broader group of end-user developers.

- **Chapter 2: IoT Across Scenarios**
  The chapter explored diverse scenarios, illustrated potential IoT applications, and identified common technical and user-centric requirements. It highlighted the need for a plug-and-play approach, ease of configuration, and reliable interconnectivity.

- **Chapter 3: Supporting End-User Developers**
  A thorough review of existing literature and toolkits revealed gaps in current IoT support tools

for end-user developers. This chapter underscored the necessity for more comprehensive, user-friendly resources to help in IoT development.

- **Chapter 4: Observing an End-User Developer**
  The long-term study presented in Chapter 4 provided an in-depth case study of an end-user developer's journey in creating a bespoke IoT project. Key skills required, such as electronic circuit construction and coding, were identified, along with the significant challenges faced.

- **Chapter 5: Concept Refinement and Ideation**
  This chapter introduced a novel deck of conceptual cards designed to aid in the ideation phase of a technology-driven project. The user study in Chapter 5 demonstrated that systematic support significantly enhances the ability of end-user developers to refine and conceptualise their project ideas.

- **Chapter 6: Project Enabler**
  By extending the *TILES IoT Toolkit* with detailed instructions and QR code links, this chapter demonstrated that enhanced toolkits effectively support the practical implementation of bespoke IoT projects. Participants in the study presented in Chapter 6 successfully used the provided toolkit to build functional prototypes.

- **Chapter 7: Guided Troubleshooting**
  The systematic troubleshooting protocol evaluated in this chapter improved the efficiency of end-user developers' problem-solving. The user study in Chapter 7 confirmed that step-by-step guidance is crucial for end-user developers when identifying and resolving issues.

- **Chapter 8: Discussion**
  A critical evaluation of the findings from all the user studies provided a synthesis of end-user IoT development compared with existing literature. This chapter highlighted the multi-faceted challenges and the importance of user-centric design in support tools.

This thesis significantly contributes to IoT and end-user development by identifying key challenges, proposing effective support strategies and validating them through empirical studies. The research addresses the initial hypothesis and provides practical implications for enhancing end-user developers' experiences and outcomes in IoT projects.

The findings of this research underscore the importance of comprehensive, systematic support for end-user developers in the IoT domain. By addressing the identified challenges and implementing the proposed strategies, the IoT community can empower more individuals to develop their IoT solutions successfully. These key findings provide a foundation for the subsequent discussion on our contribution to knowledge, future research avenues and final remarks, as presented in the following sections of this chapter.

9.2 | Contribution to Knowledge

This section outlines the contributions of this research to supporting end-user developers in IoT projects within the broader context of Human-Computer Interaction (HCI). Identifying these contributions is crucial for advancing the field, as it highlights the novel insights and practical applications that can drive future research and innovation. The following paragraphs explore the key areas of contribution, culminating in a summary of the main advancements in IoT and end-user development.

9.2.1 | Advancements in IoT for End-user Developers

This thesis advances the understanding of IoT development tailored to end-user developers by addressing their unique challenges and proposing effective support mechanisms. Through a series of user studies, the research identifies critical skills required and obstacles non-experts encounter when developing bespoke IoT projects. Focusing on the entire development lifecycle—from ideation to troubleshooting—this research provides a holistic view of the end-user developer's journey, highlighting key areas where support is most needed.

One of the major contributions is the introduction of tools and methodologies specifically designed to simplify the IoT development process for non-experts. Developing and evaluating the conceptual card deck for ideation, enhanced TILES IoT Toolkit, and systematic troubleshooting protocol exemplify how tailored support can empower end-user developers. The conceptual card deck simplifies brainstorming by providing visual and tangible aids that help users systematically organise and refine their ideas. This tool addresses the initial conceptual challenges of non-experts, enabling them to envision feasible projects without needing extensive technical knowledge.

Additionally, enhancing the TILES IoT Toolkit and introducing a systematic troubleshooting protocol significantly improve practical support for end-user developers. The enhanced toolkit includes detailed instructions and QR code links to online resources, bridging the gap between abstract concepts and real-world implementation. This makes the assembly and programming of IoT projects more straightforward. The troubleshooting protocol offers a step-by-step approach to identifying and resolving issues, reducing the frustration often associated with technical problems. These tools empower non-experts to successfully navigate the complexities of IoT development, from initial concept to functional prototype, making IoT technology more inclusive and user-friendly.

While these tools have been tested as proofs of concept in a lab environment, there is significant potential to develop them further into fully-fledged products. Future work could focus on refining these tools based on user feedback, enhancing their usability, and integrating them into comprehensive IoT development platforms. This ongoing development would ensure that the tools meet the initial design goals and evolve to address emerging needs and challenges in the rapidly changing landscape of IoT technology.

9.2.2 | Empirical and Theoretical Insights

In this section, we discuss the key insights gained from the user studies presented in Chapters 4 to 7, categorising them into empirical and theoretical contributions. The empirical insights draw from hands-on observations and user experiences, providing concrete evidence of the challenges and successes encountered by end-user developers in IoT projects. The theoretical insights build on these findings to propose new models and frameworks that advance our understanding of IoT development for non-experts. Together, these insights highlight the practical implications of our research and contribute to the broader academic discourse in Human-Computer Interaction (HCI) and IoT.

Empirical insights

The four user studies detailed in Chapter 4: Observing an End-User Developer, Chapter 5: Concept Refinement and Ideation, Chapter 6: Project Enabler, and Chapter 7: Guided Troubleshooting provide valuable empirical insights. The following paragraphs present these insights, organised according to the findings from each respective chapter.

- **Chapter 4: Observing an End-User Developer**
  The long-term study provided an in-depth look at the experiences of an end-user developer over several months while creating a bespoke IoT project. Key insights included identifying critical skills for successful IoT development, such as electronic circuit construction and programming. The study highlighted the significant time investment and varied project phases that end-user developers navigate. It revealed the nonlinear nature of project development, with phases often intertwined and influenced by external factors such as meetings or component availability. This study underscored the challenges of maintaining motivation and overcoming technical hurdles without systematic support, pointing to the need for more accessible tools and resources.

- **Chapter 5: Concept Refinement and Ideation**
  This chapter focused on the ideation phase, evaluating the effectiveness of a novel deck of conceptual cards designed to aid end-user developers. The user study showed that these cards significantly improved participants' ability to organise and refine their project ideas. The visual and tactile nature of the cards facilitated brainstorming and conceptualisation, making the initial stages of IoT development more approachable for non-experts. Participants reported increased clarity and confidence in their project concepts, demonstrating the value of systematic ideation tools in reducing conceptual vagueness and enhancing creativity.

- **Chapter 6: Project Enabler**
  In this chapter, the study explored the support needed for building and programming IoT projects. Extending the TILES IoT Toolkit with detailed instructions and QR code links, the study found that these enhancements effectively supported end-user developers in assembling and programming their IoT projects. Participants were able to successfully build functional prototypes, demonstrating the practicality of the enhanced toolkit. The study also highlighted the importance of clear, step-by-step instructions and accessible online resources in bridging the gap between high-level concepts and hands-on implementation. However, it also noted the technical challenges and the need for further refinement to ensure robustness and user-friendliness.

- **Chapter 7: Guided Troubleshooting**
  This chapter evaluated a systematic troubleshooting protocol designed to assist end-user developers in diagnosing and resolving issues in their IoT projects. The user study showed that the step-by-step approach significantly improved participants' problem-solving efficiency and reduced frustration. The systematic protocol provided a clear pathway for identifying and addressing technical issues, often a major hurdle for non-experts. Participants appreciated the guidance, which helped them maintain progress and avoid common pitfalls. The study underscored the importance of systematic troubleshooting support to enhance IoT projects' overall development experience and success rate.

These empirical insights collectively demonstrate the critical need for systematic, accessible, and user-friendly tools to support end-user developers in the various stages of IoT project development. The findings highlight areas where non-experts struggle and provide evidence-based solutions to make IoT development more manageable and less daunting for end-user developers.

Theoretical insights

The theoretical insights from this research advance our understanding of IoT development for end-user developers within the broader context of Human-Computer Interaction (HCI). Building on the empirical insights, this section contextualises the cognitive and technical challenges non-experts encounter within relevant theoretical frameworks. It identifies key learning barriers and transition points throughout the IoT development lifecycle, offering suggestions for future research to better support non-experts in overcoming these challenges.

- **Cognitive Load, Complexity, and Tangible Tools** The study presented in Chapter 4 highlights how non-expert developers experience cognitive overload at specific stages of the IoT development lifecycle, particularly when transitioning from one phase to another (e.g., from connecting sensors to processing data). Using tangible tools, such as the conceptual card deck and the extended IoT toolkit, is critical in reducing cognitive load by externalising the problem-solving process. This aligns with broader cognitive load theories in complex problem-solving and reinforces the need for accessible, physical tools that help non-experts manage technical challenges.

- **Learning Barriers, Threshold Concepts, and Modular Systems** The difficulties encountered by non-expert developers at key transition points suggest the presence of threshold concepts – critical, often challenging ideas that must be grasped to make further progress. For example, understanding cloud integration or data processing represents a significant conceptual hurdle. Modular systems, such as the Grove system used in this research, help lower these barriers by simplifying complex technical tasks into manageable components. This contributes to a broader understanding of learning barriers in IoT and technical education, highlighting how modularity in design can reduce cognitive complexity.

- **Navigating the Development Cycle and Feedback Loops** The long-term study in Chapter 4 provides a comprehensive view of how non-experts approach the IoT development lifecycle. By observing the entire process, we identified specific stages (e.g., ideation, prototyping, troubleshooting) that place heightened cognitive and technical demands on users. Feedback loops embedded in troubleshooting frameworks, particularly during the troubleshooting phase, offer critical support by providing real-time guidance and iterative feedback. These insights can inform the design of interventions to support non-experts, reduce the learning curve, and facilitate smoother transitions between stages.

- **Theoretical Implications for HCI and IoT Research** The findings from this research have broader theoretical implications for HCI and IoT. The study underscores the necessity of designing user-centric tools that cater to the unique needs of end-user developers, challenging traditional assumptions about user expertise and technical proficiency. By highlighting the interplay between cognitive processes, technical skills, and the use of tangible tools, modular systems, and feedback loops, this research advocates for an interdisciplinary approach. Combining insights from HCI, education, software engineering, and design is crucial for developing comprehensive solutions that address the multifaceted challenges end-user developers face in the IoT domain.

In summary, the theoretical insights from this research provide a foundational understanding of how tangible tools, modular systems, and feedback loops can effectively support end-user developers in IoT development. These insights highlight key strategies for addressing the cognitive and technical challenges non-experts face, offering pathways for more accessible and supportive IoT environments. By integrating these approaches, the research contributes to the theoretical discourse in HCI and IoT, underscoring the importance of user-centred, structured support systems that empower non-expert developers throughout the development lifecycle.

9.2.3 | Practical and Methodological Contributions

Considering the four user studies presented in this thesis, several practical and methodological contributions emerge from this research. The following paragraphs outline these contributions and discuss their impact on developing better context environments for IoT end-user developers.

Practical Contributions

The practical contributions of this research offer valuable insights for IoT tool designers, developers, and the broader community. This research addresses critical gaps in end-user IoT development by exploring innovative approaches such as linking physical cards to web resources through QR codes, employing operation room-inspired protocols for guided troubleshooting, and utilising physical cards for ideation and definition. These contributions underscore the importance of creating user-friendly, accessible tools that empower non-experts to engage with IoT technology effectively. The findings from this research can inform the design of more effective and user-friendly IoT development tools and support systems, ultimately fostering a more inclusive and widespread adoption of IoT solutions.

The key practical contributions of this research are as follows: the use of physical cards for ideation and definition, QR codes to link physical cards to web resources, a protocol inspired by operation rooms to guide troubleshooting, the development of user-friendly IoT prototypes, the integration of tangible elements, and comprehensive documentation and educational resources. These contributions collectively aim to enhance the accessibility and usability of IoT development tools for end-user developers.

- **Physical Cards for Ideation and Definition**
  Refining vague ideas and clearly defining a technology-based project can be daunting for an end-user developer. Using familiar elements such as gaming cards, as presented in Chapter 5, provides an effective approach to help them consolidate their ideas.
- **Integration of Tangible Elements with Digital Resources**
  The integration of tangible elements, such as conceptual cards, with QR codes linking to digital resources, as illustrated in Chapter 6 and Chapter 7, significantly enhances end-user engagement and comprehension in IoT development. These physical tools help users visualise and organise their projects while the connected web resources provide more detailed technical guidance at each step. This approach leverages the strengths of physical props while overcoming their limitations by offering access to more comprehensive information, making the development process more intuitive and accessible for non-experts to move from ideas to working prototypes.
- **A Protocol Inspired by Operation Rooms to Guide Troubleshooting**
  Adapting a known method from operation rooms to guide troubleshooting, as presented in Chapter 7, is a novel approach to learning how to guide the troubleshooting of technology-driven projects. This can be expanded beyond the context of this thesis, which was limited to the Internet of Things, to broader scenarios where a troubleshooting protocol can be applied.
- **Development of User-Friendly IoT Prototypes**
  Creating and testing prototypes that can be easily assembled and used by non-experts significantly contributes to practical IoT development. This includes creating clear, step-by-step guides and providing necessary components ready for use without extensive technical knowledge.
- **Documentation and Educational Resources**
  Providing comprehensive documentation and educational resources tailored to non-experts is crucial. This includes tutorials, instructional videos, and user-friendly manuals that simplify complex concepts and procedures.

Methodological Contributions

This research introduces several methodologies and approaches that advance the study of IoT and end-user development. Firstly, the long-term study methodology provides valuable insights into the evolving challenges and needs of end-user developers over time, particularly in the context of IoT projects that involve both hardware and software components. Secondly, the application of hybrid tangible-digital tools combines physical elements like conceptual cards with digital resources accessed via QR codes, offering non-expert users intuitive support for transitioning from ideation to prototyping. Finally, the focus on real-world, non-expert IoT troubleshooting addresses a critical gap by developing accessible tools that assist users in diagnosing and solving technical issues during IoT development. These methodologies collectively contribute to creating more accessible and supportive IoT development environments for non-expert users.

- **Long-term Study in the Context of IoT Projects**
  The use of long-term studies in the context of non-expert IoT development offers valuable insights into how users' challenges and needs evolve over time. Although long-term studies are common in HCI, applying them specifically to IoT development sheds light on how non-expert users interact with hardware, software, and networking components across different project stages.
- **Application of Hybrid Tangible-Digital Tools to IoT**
  While tangible-digital hybrid tools are not new, their application to IoT development for non-expert users is a notable contribution. The combination of physical tools (such as conceptual cards) with digital resources (accessed via QR codes) specifically addresses the challenges of guiding users from ideation to prototyping in the complex domain of IoT.
- **Focus on Real-World, Non-Expert IoT Troubleshooting**
  The emphasis on troubleshooting tools tailored to non-expert IoT developers provides a novel focus. Existing research on troubleshooting rarely targets IoT development for non-technical users. This contribution lies in the creation of accessible, real-time troubleshooting tools that address the specific difficulties faced by non-experts during IoT development.

This research's practical and methodological contributions provide significant advancements in IoT and end-user development. By applying long-term studies, hybrid tangible-digital tools, and focusing on real-world troubleshooting for non-experts, this work addresses critical gaps and lays the groundwork for future research. These approaches ensure a deeper, more practical understanding of end-user needs and experiences, ultimately contributing to the creation of more accessible and user-friendly IoT tools and systems. Collectively, these contributions support the broader adoption and effective use of IoT technologies by non-experts.

9.2.4 | Addressing Research Gaps

This section aims to identify the specific gaps in the existing literature that this research addresses and explain how the findings fill these gaps, contributing to a more comprehensive understanding of IoT development for end-user developers.

Identifying Research Gaps

The existing literature on end-user development for the IoT presents several gaps. As highlighted in Chapter 3, one of the most prominent issues is the lack of comprehensive tools that effectively support non-expert users through the entire lifecycle of IoT projects, from ideation to prototyping and maintenance. While several toolkits exist, many assume a level of technical knowledge that non-expert developers typically lack, creating significant barriers to entry. Furthermore, the literature reveals inadequate support during the prototyping phase, where non-experts struggle with integrating hardware, software, and network components. As described in the reviewed studies, this phase is critical but often lacks the necessary debugging and iterative testing resources. Another identified gap is the insufficient involvement of end users in the testing and validation stages of IoT toolkits, which limits their effectiveness in addressing real-world challenges faced by non-professionals. Moreover, the reviewed research underscores the lack of detailed guidance and instructional materials for non-experts, making navigating common IoT development issues difficult. Addressing these gaps is crucial for empowering non-expert developers and enhancing the accessibility of IoT technologies.

Addressing the Gaps

Our research, presented in Chapters 4 to 7, directly tackles these gaps by introducing and evaluating several user-centred tools and methods tailored for non-expert IoT developers. Chapter 4 addresses the challenges faced during the entire IoT lifecycle by conducting a long-term user study to identify the specific barriers that non-expert developers encounter, from ideation to maintenance. This study informs the development of more comprehensive support tools that provide more explicit guidance and structured learning throughout the development process.

In Chapter 5, we focus on the ideation phase by testing conceptual ideation cards specifically designed to help non-expert users generate and refine project ideas. These cards help bridge the gap in technical knowledge, providing structure during the ideation phase and offering an intuitive way to think about projects' technical feasibility. The cards were developed to ensure that even users with minimal technical skills could start with clear, actionable concepts.

Chapter 6 extends this work by extending the TILES IoT Toolkit, focusing on the prototyping phase. The toolkit offers structured guidance on building functional prototypes and incorporates QR codes that link to instructional resources. This approach directly addresses the gap in support for integrating hardware, software, and network components, simplifying the prototyping process for non-experts by providing just-in-time learning and concrete examples.

Finally, Chapter 7 addresses the gap in troubleshooting support by focusing on tools and methods that assist non-experts in identifying and solving technical problems during IoT development. The user study presented in this chapter highlights the need for real-time feedback and intuitive error-handling mechanisms and explores how end-users interact with tools designed to help them debug and troubleshoot issues. This chapter fills the gap by emphasising the importance of tailored troubleshooting resources that are easy to understand and apply for non-expert developers.

Through these studies, our research provides a comprehensive response to the gaps identified in the literature by offering practical tools and insights that support non-expert developers throughout the entire IoT development lifecycle.

9.2.5 | Synthesis of Contributions

This research addresses several gaps in the literature on IoT development for end-user developers. One major gap is the lack of long-term studies exploring the experiences and challenges end-user developers face over extended periods. Existing research often focuses on short-term interactions and immediate outcomes, leaving a significant gap in understanding these users' sustained engagement and evolving needs. Additionally, practical support systems, such as integrating QR codes and physical cards, are underexplored. These tools can bridge the gap between physical and digital resources, yet their potential remains untapped. Furthermore, while gamification elements are recognised for their motivational benefits, their application typically does not span the entire lifecycle of an IoT project, from ideation through to building and ongoing support.

The findings of this research fill these gaps by introducing innovative methodologies and practical tools that enhance the usability and accessibility of IoT development for non-experts. By adopting a user-centred design approach, this research ensures that the development process aligns with the actual needs and limitations of end-users, incorporating their feedback through surveys, interviews, and user studies. Using long-term case studies offers a comprehensive view of end-user developers' experiences and challenges as they evolve, revealing insights that short-term studies often overlook. Additionally, integrating QR codes and physical cards facilitates seamless transitions between physical and digital resources, enhancing the ideation and development phases. Including gamification elements throughout the project lifecycle further engages and motivates users, making the development process more interactive and rewarding.

Overall, this research contributes to a more comprehensive understanding of IoT development for end-user developers by addressing previously unexplored areas and providing practical solutions to common challenges. By filling these gaps, it advances the field. It supports the broader adoption and effective use of IoT technologies by non-experts, ultimately fostering a more inclusive and innovative technological landscape.

9.3 | Limitations and Further Research

As discussed in Chapter 8, this research has several limitations, including the small sample size, the focus on specific types of IoT projects, and the emphasis on the early stages of IoT development. These limitations not only highlight areas for improvement but also suggest directions for future research. To address these gaps, future studies should aim to expand the participant pool, explore a broader range of IoT project types, and investigate the later stages of IoT development, such as maintenance and scaling. Additionally, adopting a long-term approach that follows end-user developers throughout the entire project lifecycle could provide deeper insights into their evolving needs and challenges.

Building on this study's findings, several avenues for further research can advance the field of IoT development for end-user developers:

- **In-the-wild user studies:** Conducting studies in natural settings where end-user developers engage with IoT tools can provide more realistic and comprehensive insights. This approach, complementing the controlled studies presented in this thesis, could reveal unforeseen challenges and opportunities, ultimately leading to more effective IoT tools and support systems.

- **Expanding research to diverse user demographics:** Future research should include a more diverse user base, incorporating different age groups, cultural backgrounds, and technical skill levels. Understanding how various demographics interact with IoT technologies will contribute to creating more inclusive and accessible solutions that cater to the specific needs of a wider audience.

– **Longitudinal studies on learning curves:** Tracking end-user developers over extended periods, from novice to more experienced stages, could provide valuable insights into the effectiveness of current training tools and resources. This could inform the development of more effective educational materials and support mechanisms that promote continuous learning and skill progression.

– **Integration of advanced technologies, such as AI:** Exploring the integration of AI-powered tools into IoT development could significantly enhance the user experience. AI-driven features like real-time feedback, intelligent assistance, and automation can simplify complex development processes, making IoT development more efficient and user-friendly for non-experts.

– **Cross-disciplinary collaborations:** Fostering collaborations between HCI researchers, designers, engineers, and social scientists can help address the multifaceted challenges end-user developers face in IoT. By integrating diverse perspectives and expertise, more holistic and innovative solutions can be developed to meet end users' needs better.

Pursuing these further research directions will enable the field of IoT development for end users to evolve, helping to ensure that IoT technologies are accessible, usable, and effective for non-expert developers.

9.4 | Final Remarks

This section summarises the research's overall contributions, reflects on the journey, and provides a closing perspective on the work's significance and future potential.

9.4.1 | Summary of Contributions

This research has made several key contributions to the field of IoT development for end-user developers. It has introduced innovative tools such as QR codes and physical cards to facilitate ideation and development, and integrated web-supported protocols to enhance user experience. Methodologically, the research has employed different approaches, including user-centred design, long-term case studies, and mixed-methods research. These contributions address significant gaps in the existing literature and provide a more comprehensive understanding of the challenges and opportunities in IoT development for non-experts.

9.4.2 | Broader Implications and Future Potential

While this research has contributed valuable insights into developing IoT tools for end users, many challenges remain unresolved. Technology vendors like Arduino continue to release products that promise easy, plug-and-play IoT solutions for everyone. For instance, Arduino recently introduced a new kit with the bold claim: *"Hey, creating an IoT device shouldn't be rocket science. We believe technology is for everyone. That's why we've developed the whole new, beginner-friendly Plug and Make Kit – the easiest way to get started with Arduino!"* [6]. However, despite marketing promises

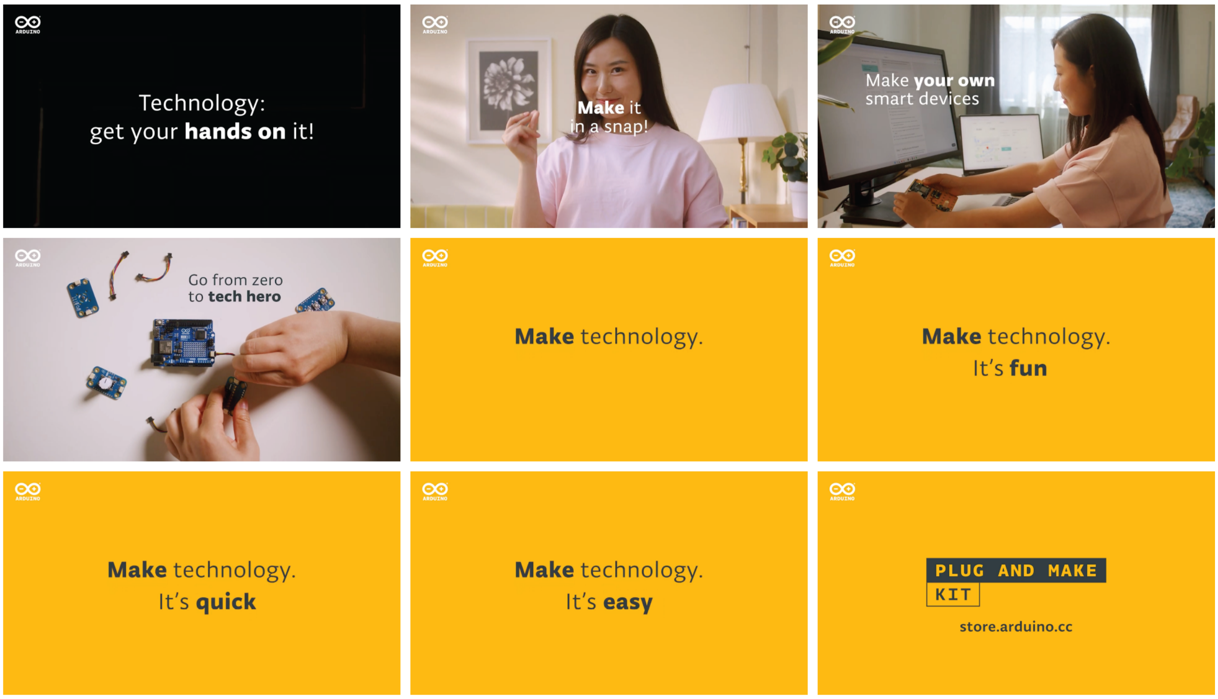


FIGURE 9.1: 2024 Arduino Plug and Make Kit promotional video.
Reproduced from *Arduino – YouTube*, https://www.youtube.com/@Arduino.

(Figure 9.1), the gap between these claims and the actual experience of end-user developers persists. End users still face significant barriers, particularly when transitioning from ideation to functional prototypes and troubleshooting complex technical issues.

This ongoing challenge reflects the broader implications of this research, highlighting the persistent need for more accessible and intuitive IoT development environments. Although some progress has been made—particularly through the use of hybrid tangible-digital tools and enhanced support for troubleshooting—there remains much to be done to make IoT development truly accessible to non-experts. Closing this gap aligns with the potential that technologies like Arduino promise, suggesting that research must explore how to simplify the transition from ideation to prototype more effectively, addressing the technical barriers that non-experts continue to face.

The societal implications of this research are significant, as it aims to democratise IoT development and empower non-expert users to create and manage their IoT systems. Future research directions, such as in-the-wild user studies and expanding the diversity of the user base, are crucial for building on these findings and advancing the field. By developing more inclusive and user-friendly IoT tools, we can foster broader innovation and accessibility, making bespoke IoT projects more approachable and encouraging wider adoption in the field.

9.4.3 | Concluding Thoughts

Although this thesis has been written using the plural form (*"we have developed..."*), I will now shift to the singular for this final section. In conclusion, this research has made significant strides in understanding and supporting IoT development for end-user developers. The contributions made have the potential to drive future advancements in the field, supporting the broader adoption and effective use of IoT technologies by non-experts. For future researchers, I encourage you to build on this work, continue exploring innovative methodologies, and remain dedicated to creating inclusive and user-centred technological solutions. Together, we can advance the field and empower more people to engage with and benefit from IoT technology.

References and Appendices

Bibliography

[1] Abbate, J. (1999). Getting small: A short history of the personal computer. *Proceedings of the IEEE*, *87*(9), 1695–1698.

[2] Adams, A., Lunt, P., & Cairns, P. (2008). A qualitative approach to HCI research. In P. Cairns & A. Cox (Eds.), *Research methods for Human-Computer Interaction* (pp. 138–157). Cambridge University Press.

[3] Aghajani, E., Nagy, C., Vega-Márquez, O. L., Linares-Vásquez, M., Moreno, L., Bavota, G., & Lanza, M. (2019). Software documentation issues unveiled. *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, 1199–1210.

[4] Ambe, A. H., Brereton, M., Soro, A., Chai, M. Z., Buys, L., & Roe, P. (2019). Older people inventing their personal Internet of Things with the IoT Un-Kit experience. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–15.

[5] Anderson, C. (2014). *Makers*. Random House USA.

[6] Arduino Team. (2024). Kickstart your tech journey with the new Arduino Plug-and-Make Kit. https://blog.arduino.cc/2024/07/11/kickstart-your-tech-journey-with-the-new-arduino-plug-and-make-kit/

[7] Arquilla, V., & Vitali, I. (2016). Designing in the IoT era: Role and perspectives in design practices. *6th International Forum of Design as a Process Systems & Design: Beyond Processes and Thinking*, 871–882.

[8] Ashton, K. (2009). That 'internet of things' thing. *RFiD Journal*, *22*(7), 97–114. Retrieved May 28, 2017, from http://www.rfidjournal.com/articles/view?4986

[9] Aspiala, T. (2014, December 10). *Cards for brainstorming product design* [Boing boing]. Retrieved March 5, 2021, from https://boingboing.net/2014/12/10/cards-for-brainstorming-produc.html

[10] Aspiala, T., & Deschamps-Sonsino, A. (2016). *Learn. play. collect.* [Know cards]. Retrieved April 4, 2022, from https://know-cards.myshopify.com/

[11] Barricelli, B. R., Cassano, F., Fogli, D., & Piccinno, A. (2019). End-user development, end-user programming and end-user software engineering: A systematic mapping study. *Journal of Systems and Software*, *149*, 101–137.

[12] Barricelli, B. R., & Valtolina, S. (2015). Designing for end-user development in the internet of things. *International symposium on end user development*, 9–24.

[13] Bassett, L. (2015). *Introduction to JavaScript object notation: A to-the-point guide to JSON*. O'Reilly Media, Inc.

[14] Beck, K., & Cunningham, W. (1989). A laboratory for teaching object oriented thinking. *ACM Sigplan Notices*, *24*(10), 1–6.

[15] Becker, B. A., Denny, P., Pettit, R., Bouchard, D., Bouvier, D. J., Harrington, B., Kamil, A., Karkare, A., McDonald, C., Osera, P.-M., et al. (2019). Compiler error messages considered unhelpful: The landscape of text-based programming error message research. *Proceedings of the working group reports on innovation and technology in computer science education*, 177–210.

[16] Bell, C. (2021). *Beginning IoT projects*. Springer.

[17] Blackwell, A. F. (2017). End-user developers–what are they like? *New perspectives in end-user development*, 121–135.

[18] Booth, T. (2015). Investigating the barriers experienced by adult end-user developers when physical prototyping. *International Symposium on End User Development*, 255–258.

[19] Booth, T. (2022). *Tactical troubleshooting: Investigating support for end-user developers in physical computing* [Doctoral dissertation, City, University of London].

[20] Brown, T., & Katz, B. (2009). *Change by design: How design thinking transforms organizations and inspires innovation* (1st ed). Harper Business.

[21] Brumby, D. P., Blandford, A., Cox, A. L., Gould, S. J., & Marshall, P. (2016). Research methods for HCI: Understanding people using interactive technologies. *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 1028–1031.

[22] Burnett, M., & Kulesza, T. (2015). End-user development in internet of things: We the people. *International Reports on Socio-Informatics (IRSI)*, *12*(2), 81–86.

[23] Burnett, M. M., & Scaffidi, C. (2011). End-user development. *Encyclopedia of Human-Computer Interaction*.

[24] Buse, R. P., & Weimer, W. (2012). Synthesizing API usage examples. *2012 34th International Conference on Software Engineering (ICSE)*, 782–792.

[25] Buxton, B. (2010). *Sketching user experiences: Getting the design right and the right design*. Morgan Kaufmann.

[26] Buyya, R., & Dastjerdi, A. V. (2016). *Internet of Things: Principles and paradigms*. Morgan Kaufmann.

[27] Chacon, S., & Straub, B. (2014). Pro Git. 2nd. *Berkely, CA, USA: Apress*.

[28] Chen, G.-D., Li, L.-Y., & Wang, C.-Y. (2012). A community of practice approach to learning programming. *Turkish Online Journal of Educational Technology-TOJET*, *11*(2), 15–26.

[29] Cooper, A., Reimann, R., Cronin, D., & Noessel, C. (2014, September). *About face: The essentials of interaction design* (4th). Wiley.

[30] Corno, F., De Russis, L., & Roffarello, A. M. (2017a). A high-level approach towards end user development in the IoT. *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 1546–1552.

[31] Corno, F., De Russis, L., & Roffarello, A. M. (2017b). A semantic web approach to simplifying trigger-action programming in the IoT. *Computer*, *50*(11), 18–24.

[32] Corno, F., De Russis, L., & Sáenz, J. P. (2019). On the challenges novice programmers experience in developing IoT systems: A survey. *Journal of Systems and Software*, *157*, 110389.
[33] Corno, F., Russis, L. D., & Monge Roffarello, A. (2019). My IoT puzzle: Debugging if-then rules through the jigsaw metaphor. *International Symposium on End User Development*, 18–33.
[34] Costabile, M. F., Mussio, P., Parasiliti Provenza, L., & Piccinno, A. (2008). End users as unwitting software developers. *Proceedings of the 4th international workshop on End-user software engineering*, 6–10.
[35] Cotti, L., Guizzardi, D., Barricelli, B. R., & Fogli, D. (2024). Enabling end-user development in smart homes: A machine learning-powered digital twin for energy efficient management. *Future Internet*, *16*(6), 208.
[36] Council, D. (2015, March 17). *What is the framework for innovation? design council's evolved double diamond* [Design council]. Retrieved December 7, 2020, from https://www.designcouncil.org.uk/news-opinion/what-framework-innovation-design-councils-evolved-double-diamond
[37] Cross, N. (2007). *Designerly ways of knowing*. Birkhäuser.
[38] Cross, N. (2021, March). *Engineering design methods: Strategies for product design* (5th). Wiley.
[39] Curley, A. (2017). What is citizen science? *Stream of Consciousness*, 12.
[40] Cypher, A., & Halbert, D. C. (1993). *Watch what i do: Programming by demonstration*. MIT press.
[41] Dahl, O. (2023, September). It doesn't work? *Build Electronic Circuits*.
[42] Daley, B. J. (1999). Novice to expert: An exploration of how professionals learn. *Adult education quarterly*, *49*(4), 133–147.
[43] Dam, R. F., & Siang, T. Y. (2017). *What is ideation – and how to prepare for ideation sessions* [The interaction design foundation]. Retrieved March 27, 2022, from https://www.interaction-design.org/literature/article/what-is-ideation-and-how-to-prepare-for-ideation-sessions
[44] Danado, J., & Paternò, F. (2015). A mobile end-user development environment for IoT applications exploiting the puzzle metaphor. *ERCIM News*, *101*(April), 26–27.
[45] Darzentas, D., Velt, R., Wetzel, R., Craigon, P. J., Wagner, H. G., Urquhart, L. D., & Benford, S. (2019). Card mapper: Enabling data-driven reflections on ideation cards. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–15.
[46] De Roeck, D. (2019). IoT ideation cards. dott. research.
[47] De Roeck, D., Stappers, P. J., & Standaert, A. (2014). Gearing up! a designer-focused evaluation of ideation tools for connected products. *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, 521–530.
[48] De Roeck, D., Tanghe, J., Jacoby, A., Moons, I., & Slegers, K. (2019). Ideas of things: The IoT Design Kit. *Companion Publication of the 2019 on Designing Interactive Systems Conference 2019 Companion*, 159–163.

[49] De Souza, C. S. (2005). *The Semiotic Engineering of Human-Computer Interaction*. MIT press.

[50] Deng, Y., Antle, A. N., & Neustaedter, C. (2014). Tango cards: A card-based design tool for informing the design of tangible learning games. *Proceedings of the 2014 conference on Designing interactive systems*, 695–704.

[51] Denning, P. J., & Tedre, M. (2019). *Computational thinking*. MIT Press.

[52] Deschamps-Sonsino, A. (n.d.). Good night lamp. https://www.designswarm.com/good-night-lamp/

[53] Desolda, G., Ardito, C., & Matera, M. (2017). Empowering end users to customize their smart environments: Model, composition paradigms, and domain-specific tools. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *24*(2), 1–52.

[54] Desolda, G., Deufemia, V., Gena, C., Matera, M., Paternò, F., & Treccani, B. (2020). Empathy: Empowering people in dealing with Internet of Things ecosystems (workshop). *Proceedings of the International Conference on Advanced Visual Interfaces*, 1–1.

[55] Despa, M. L. (2014). Comparative study on software development methodologies. *Database Systems Journal*, *5*(3), 37–56.

[56] Deterding, S. (2012). Gamification: Designing for motivation. *interactions*, *19*(4), 14–17.

[57] Deterding, S., Björk, S. L., Nacke, L. E., Dixon, D., & Lawley, E. (2013). Designing gamification: Creating gameful and playful experiences. In *Chi'13 extended abstracts on human factors in computing systems* (pp. 3263–3266).

[58] Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining"gamification". *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, 9–15.

[59] Deterding, S., Sicart, M., Nacke, L., O'Hara, K., & Dixon, D. (2011). Gamification: Using game-design elements in non-gaming contexts. In *Chi'11 extended abstracts on human factors in computing systems* (pp. 2425–2428). Association for Computing Machinery.

[60] Dibitonto, M., Tazzi, F., Leszczynska, K., & Medaglia, C. M. (2017). The IoT design deck: A tool for the co-design of connected products. *International Conference on Applied Human Factors and Ergonomics*, 217–227.

[61] Dibitonto, M., Tazzi, F., Leszczynska, K., & Medaglia, C. M. (2019). The IoT design deck 2.0: Improving the tool for the co-design of connected products. *International Conference on Human-Computer Interaction*, 63–74.

[62] Divitini, M., Giannakos, M. N., Mora, S., Papavlasopoulou, S., & Iversen, O. S. (2017). Make2learn with IoT: Engaging children into joyful design and making of interactive connected objects. *Proceedings of the 2017 Conference on Interaction Design and Children*, 757–760.

[63] Dourish, P. (2003). The appropriation of interactive technologies: Some lessons from placeless documents. *Computer Supported Cooperative Work (CSCW)*, *12*, 465–490.

[64] Drajic, D., Andersson, K., Zhang, K., Stembert, N., Malmberg, K., Brekine, A., Vanobberghen, W., Habibipour, A., & Waeben, J. (2019). User engagement for large scale pilots in the Internet of Things. *2019 14th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, 46–53.

[65] Dreyfus, H., Dreyfus, S. E., & Athanasiou, T. (2000). *Mind over machine*. Simon; Schuster.

[66] Drubin, C. (2014). The internet of things will drive wireless connected devices to. 40.9 billion in 2020. *Microwave Journal*, *57*(10), 51.

[67] Duncan, C., Bell, T., & Atlas, J. (2017). What do the teachers think? introducing computational thinking in the primary school curriculum. *Proceedings of the Nineteenth Australasian Computing Education Conference*, 65–74.

[68] Dunne, A., & Raby, F. (2013). *Speculative everything: Design, fiction, and social dreaming*. The MIT Press.

[69] Eisenberg, M. (2013). 3d printing for children: What to build next? *International Journal of Child-Computer Interaction*, *1*(1), 7–13.

[70] Evans, D. (2011). The internet of things: How the next evolution of the internet is changing everything. *CISCO White Paper*, 1–11.

[71] Ferrari, M. I., Bim, S. A., & Aquino, P. T. (2018). The signs of semiotic engineering in the IoT interaction design. In C. Baldwin (Ed.), *Advances in neuroergonomics and cognitive engineering* (pp. 421–433). Springer International Publishing.

[72] Fischer, G. (2021). End-user development: Empowering stakeholders with artificial intelligence, meta-design, and cultures of participation. *International Symposium on End User Development*, 3–16.

[73] Fischer, G., Fogli, D., & Piccinno, A. (2017). Revisiting and broadening the meta-design framework for end-user development. In F. Paternò & V. Wulf (Eds.), *New perspectives in end-user development* (pp. 61–97). Springer International Publishing.

[74] Fischer, G., & Giaccardi, E. (2006). Meta-design: A framework for the future of end-user development. *End user development*, 427–457.

[75] Friedman, B., & Hendry, D. (2012). The envisioning cards: A toolkit for catalyzing humanistic and technical imaginations. *Proceedings of the SIGCHI conference on human factors in computing systems*, 1145–1148.

[76] Friedman, B., Kahn, P. H., & Borning, A. (2002). Value Sensitive Design: Theory and Methods, 8.

[77] Gallacher, S., O'Connor, J., Bird, J., Rogers, Y., Capra, L., Harrison, D., & Marshall, P. (2015). Mood Squeezer: Lightening up the Workplace through Playful and Lightweight Interactions. *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW '15*, 891–902.

[78] García Ferrari, T., Hinze, A., & Bowen, J. (2018). An IoT for Everyone: Fact or Fiction? *Proceedings of the 32nd International BCS Human Computer Interaction Conference*, 1–5.

[79] Gennari, R., Roumelioti, E., & Ugur Yavuz, S. (2021). Help me create smart things: How to support design and art students at a distance. *International Symposium on End User Development*, 87–101.

[80] Gericke, K., & Blessing, L. (2011). Comparisons of design methodologies and process models across disciplines: A literature review. *Proceedings of the 18th International Conference on Engineering Design*.

[81] Gianni, F., Mora, S., & Divitini, M. (2018). Rapid prototyping internet of things applications for augmented objects: The tiles toolkit approach. *European Conference on Ambient Intelligence*, 204–220.

[82] Gianni, F., Mora, S., & Divitini, M. (2019). Rapiot toolkit: Rapid prototyping of collaborative Internet of Things applications. *Future Generation Computer Systems*, *95*, 867–879.

[83] Gianni, F. V. (2019). *From ideation to prototyping of IoT systems: The case of smart cities* [Doctoral dissertation, Norwegian University of Science and Technology]. NTNU.

[84] Gillmor, D. (2006). *We the media: Grassroots journalism by the people, for the people* (Pbk. ed). "O'Reilly Media, Inc."

[85] Golembewski, M., & Selby, M. (2010). Ideation decks: A card-based design ideation tool. *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, 89–92.

[86] Gray, K. A., Guzdial, M., & Rugaber, S. (2003). Extending crc cards into a complete design process. *ACM SIGCSE Bulletin*, *35*(3), 226–226.

[87] Haiduc, S., Aponte, J., Moreno, L., & Marcus, A. (2010). On the use of automated text summarization techniques for summarizing source code. *2010 17th Working conference on reverse engineering*, 35–44.

[88] Halbert, D. C. (1984). *Programming by example* [Doctoral dissertation, University of California, Berkeley].

[89] Haller, S., Karnouskos, S., & Schroth, C. (2008). The internet of things in an enterprise context. *Future Internet Symposium*, 14–28.

[90] Hars, A., & Ou, S. (2002). Working for free? motivations for participating in open-source projects. *International Journal of Electronic Commerce*, *6*(3), 25–39.

[91] Hornecker, E. (2010). Creative idea exploration within the structure of a guiding framework: The card brainstorming game. *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, 101–108.

[92] Howard, P. N. (2015). *Pax technica: How the internet of things may set us free or lock us up*. Yale University Press.

[93] Jadud, M. C. (2006). *An exploration of novice compilation behaviour in bluej*. University of Kent (United Kingdom).

[94] Jencks, C., & Silver, N. (2013). *Adhocism, expanded and updated edition: The case for improvisation*. Mit Press.

[95] Jensen, C. (2018). APIs for dummies®, 3rd ibm li. *Hoboken: IBM*.

[96] Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F., & Alonso-Zarate, J. (2015). A survey on application layer protocols for the internet of things. *Transaction on IoT and Cloud computing*, *3*(1), 11–17.

[97] Kelley, T., & Kelley, D. (2013). *Creative confidence: Unleashing the creative potential within us all*. Currency.

[98] Kernighan, B. W. (2017). *Understanding the digital world: What you need to know about computers, the internet, privacy, and security*. Princeton University Press.

[99] Kubitza, T., & Schmidt, A. (2015). Towards a toolkit for the rapid creation of smart environments. *International Symposium on End User Development*, 230–235.

[100] Kulesza, T., Burnett, M., Wong, W.-K., & Stumpf, S. (2015). Principles of explanatory debugging to personalize interactive machine learning. *Proceedings of the 20th international conference on intelligent user interfaces*, 126–137.

[101] Kurlander, D., Cypher, A., & Halbert, D. C. (1993). *Watch what i do: Programming by demonstration*. MIT Press.

[102] Kurze, A., Totzauer, S., Deschamps-Sonsino, A., & Berger, A. (2019). A collaborative landscaping exercise of IoT design methods. *Proceedings of the 31st Australian Conference on Human-Computer-Interaction*, 307–311.

[103] Lakhani, K. R., & Wolf, R. G. (2003). Why hackers do what they do: Understanding motivation and effort in free/open source software projects. *Open Source Software Projects (September 2003)*.

[104] Lawson, B. (2006). *How designers think: The design process demystified*. Routledge.

[105] Ledo, D., Houben, S., Vermeulen, J., Marquardt, N., Oehlberg, L., & Greenberg, S. (2018). Evaluation strategies for HCI toolkit research. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–17.

[106] Leedy, P. D., & Ormrod, J. E. (2019). *Practical research: Planning and design*. ERIC.

[107] Lewis, J. R. (1991). Psychometric evaluation of an after-scenario questionnaire for computer usability studies: The asq. *ACM Sigchi Bulletin*, *23*(1), 78–81.

[108] Li, T. J.-J., Li, Y., Chen, F., & Myers, B. A. (2017). Programming IoT devices by demonstration using mobile apps. *International Symposium on End User Development*, 3–17.

[109] Lieberman, H. (2001). *Your wish is my command: Programming by example*. Morgan Kaufmann.

[110] Lieberman, H., Paternò, F., Klann, M., & Wulf, V. (2006). End-user development: An emerging paradigm. In *End user development* (pp. 1–8). Springer.

[111] Lim, Y.-K., Stolterman, E., & Tenenberg, J. (2008). The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *15*(2), 1–27.

[112] Lindley, J., Coulton, P., & Cooper, R. (2017). Why the internet of things needs object orientated ontology. *The Design Journal*, *20*(sup1), S2846–S2857.

[113] Logler, N., Yoo, D., & Friedman, B. (2018). Metaphor cards: A how-to-guide for making and using a generative metaphorical design toolkit. *Proceedings of the 2018 Designing Interactive Systems Conference*, 1373–1386.

[114] Lowe, D. (2017). *Electronics all-in-one for dummies*. John Wiley & Sons.

[115] Lucero, A., Dalsgaard, P., Halskov, K., & Buur, J. (2016). Designing with cards. *Collaboration in creative design: Methods and tools*, 75–95.

[116] Ludwig, T., Boden, A., & Pipek, V. (2017). 3d printers as sociable technologies: Taking appropriation infrastructures to the internet of things. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *24*(2), 1–28.

[117] Ludwig, T., Dax, J., Pipek, V., & Wulf, V. (2017). A practice-oriented paradigm for end-user development. In *New perspectives in end-user development* (pp. 23–41). Springer.

[118] Luger, E., Urquhart, L., Rodden, T., & Golembewski, M. (2015). Playing the legal card: Using ideation cards to raise data protection issues within the design process. *Proceedings of the 33rd Annual ACM conference on human factors in computing systems*, 457–466.

[119] Lupton, D. (2016). *The quantified self: A sociology of self-tracking*. Polity.

[120] Mæhlum, A. R. (2017). *Extending the tiles toolkit-from ideation to prototyping* [Master's thesis, NTNU].

[121] Malizia, A., Valtolina, S., Morch, A., Serrano, A., & Stratton, A. (2019). *End-user development: 7th international symposium, is-eud 2019, hatfield, uk, july 10–12, 2019, proceedings* (Vol. 11553). Springer.

[122] Manca, M., Paternò, F., Santoro, C., & Corcella, L. (2019). Supporting end-user debugging of trigger-action rules for IoT applications. *International Journal of Human-Computer Studies*, *123*, 56–69.

[123] Markopoulos, P., Nichols, J., Paternò, F., & Pipek, V. (2017). End-user development for the Internet of Things. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *24*(2), 1–3.

[124] Mavroudi, A., Divitini, M., Gianni, F., Mora, S., & Kvittem, D. R. (2018). Designing IoT applications in lower secondary schools. *2018 IEEE Global Engineering Education Conference (EDUCON)*, 1120–1126.

[125] McBurney, P. W. (2015). Automatic documentation generation via source code summarization. *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, *2*, 903–906.

[126] Mercieca, B. (2017). What is a community of practice? In J. McDonald & A. Cater-Steel (Eds.), *Communities of practice: Facilitating social learning in higher education* (pp. 3–25). Springer Singapore.

[127] Mora, S., Divitini, M., & Gianni, F. (2016). Tiles: An inventor toolkit for interactive objects. *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 332–333.

[128] Mora, S., Gianni, F., & Divitini, M. (2017). Tiles: A card-based ideation toolkit for the internet of things. *Proceedings of the 2017 conference on designing interactive systems*, 587–598.

[129] Mora, S., Gianni, F., Nichele, S., & Divitini, M. (2018). Introducing IoT competencies to first-year university students with the tiles toolkit. *Proceedings of the 7th Computer Science Education Research Conference*, 26–34.

[130] Moulton, P. G., & Muller, M. E. (1967). Ditran—a compiler emphasizing diagnostics. *Commun. ACM*, *10*(1), 45–52.

[131] Mueller, F., Gibbs, M. R., Vetere, F., & Edge, D. (2014). Supporting the creative game design process with exertion cards. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2211–2220.

[132] Munari, B. (2018). *Da cosa nasce cosa: Appunti per una metodologia progettuale*. Gius. Laterza & Figli Spa.

[133] Muñoz, Á. A., Florin, U., Eriksson, Y., Yamamoto, Y., & Sandström, K. (2020). The karakuri card deck: Co-designing industrial IoT conceptual solutions. *Proceedings of the Design Society: DESIGN Conference*, *1*, 807–816.

[134] Nassif, M., & Robillard, M. P. (2023). A field study of developer documentation format. *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–7.

[135] Neff, G., & Nafus, D. (2016). *Self-tracking*. The MIT Press.

[136] Nelson, H. G., & Stolterman, E. (2014). *The design way: Intentional change in an unpredictable world*. MIT Press.

[137] Noone, M., & Mooney, A. (2018). Visual and textual programming languages: A systematic review of the literature. *Journal of Computers in Education*, *5*(2), 149–174.

[138] Norman, D. (2013). *The design of everyday things: Revised and expanded edition*. Basic books.

[139] Norman, D. A. (1988). *The psychology of everyday things.* Basic books.

[140] Oliver, H. (2019). Participatory design fiction for innovation in everyday wearable IoT systems: Demo abstract. *Proceedings of the International Conference on Internet of Things Design and Implementation*, 285–286.

[141] O'Reilly, T. (2009). *What is web 2.0*. O'Reilly Media.

[142] Paternò, F. (2013). End user development: Survey of an emerging field for empowering people. *International Scholarly Research Notices*, *2013*.

[143] Paternò, F. (n.d.). End-user development: Concepts, approaches, new frontiers.

[144] Paternò, F., & Santoro, C. (2017). A design space for end user development in the time of the internet of things. *New perspectives in end-user development*, 43–59.

[145] Phan, T. A. M., Nurminen, J. K., & Di Francesco, M. (2014). Cloud databases for internet-of-things data. *2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, 117–124.

[146] Ramohalli, N., & Adegbija, T. (2018). Modular electronics for broadening non-expert participation in STEM innovation: An IoT perspective. *2018 IEEE Integrated STEM Education Conference (ISEC)*, 167–174.

[147] Ray, P. P. (2017). A survey on visual programming languages in internet of things. *Scientific Programming*, *2017*.

[148] Rizvi, M. (2021). Supporting end users in designing IoT smartthings with the IoTgo Toolkit. *EMPATHY@INTERACT*.

[149] Rogers, Y., Marshall, P., & Carroll, J. M. (2017). *Research in the wild*. Springer.

[150] Rogers, Y., Sharp, H., & Preece, J. (2023, April). *Interaction design: Beyond human-computer interaction* (6th). Wiley.

[151] Rose, D. (2014). *Enchanted objects: Design, human desire, and the Internet of things* (First Scribner hardcover edition). Scribner.

[152] Rose, K., Eldridge, S., & Chapin, L. (2015). The internet of things: An overview. *The internet society (ISOC)*, *80*, 1–50.

[153] Roy, R., & Warren, J. P. (2019). Card-based design tools: A review and analysis of 155 card decks for designers and designing. *Design Studies*, *63*, 125–154.
[154] Ryan, R. M., & Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, *25*(1), 54–67.
[155] Saeed, S., Jhanjhi, N., Naqvi, M., & Humayun, M. (2019). Analysis of software development methodologies. *International Journal of Computing and Digital Systems*, *8*(5), 446–460.
[156] Said, O., & Masud, M. (2013). Towards internet of things: Survey and future vision. *International Journal of Computer Networks*, *5*(1), 1–17.
[157] Sas, C., & Neustaedter, C. (2017). Exploring diy practices of complex home technologies. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *24*(2), 1–29.
[158] Schach, S. R. (2007). *Object-oriented and classical software engineering* (Vol. 6). McGraw-Hill New York.
[159] Schaefer, M. (2018). *The certainty of uncertainty: The way of inescapable doubt and its virtue*. Wipf; Stock Publishers.
[160] Silva, M., Dias, J. P., Restivo, A., & Ferreira, H. S. (2021). A review on visual programming for distributed computation in iot. *International Conference on Computational Science*, 443–457.
[161] Simon, H. A. (1996). *The sciences of the artificial* (3rd ed). MIT Press.
[162] Speelman, C. P., & Kirsner, K. (2005). *Beyond the learning curve: The construction of mind*. Oxford University Press on Demand.
[163] Spinellis, D. (2018). Being a software developer. *IEEE Software*, *35*(4), 4–7.
[164] Stembert, N. (2017, June). Co-creative workshop methodology handbook.
[165] Sweller, J., Ayres, P., & Kalyuga, S. (2011). The expertise reversal effect. In *Cognitive load theory* (pp. 155–170). Springer New York.
[166] Team, A. (2019, February 6). *Announcing the arduino IoT cloud public beta*. Retrieved June 28, 2022, from https://blog.arduino.cc/2019/02/06/announcing-the-arduino-iot-cloud-public-beta/
[167] Tesler, L. G. (1991). Networked computing in the 1990s. *Scientific American*, *265*(3), 86–93.
[168] Tetteroo, D., Markopoulos, P., Valtolina, S., Paternò, F., Pipek, V., & Burnett, M. (2015). End-user development in the internet of things era. *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, 2405–2408.
[169] The Seattle Times. (2024). Sony kit helps tinkerers invent sensor devices. *The Seattle Times*. https://www.seattletimes.com/business/sony-kit-helps-tinkerers-invent-sensor-devices/
[170] Thomas, L., Boustedt, J., Eckerdal, A., McCartney, R., Moström, J.-E., Sanders, K., & Zander, C. (2014). A broader threshold: Including skills as well as concepts in computing education. *NAIRTL 2014, National Academy's Sixth Annual Conference and the Fourth Biennial Threshold Concepts Conference, 27-29 June 2012, Dublin, Ireland*, 154–158.

[171] Uckelmann, D., Harrison, M., & Michahelles, F. (2011). An Architectural Approach Towards the Future Internet of Things. In D. Uckelmann, M. Harrison, & F. Michahelles (Eds.), *Architecting the Internet of Things* (pp. 1–24). Springer Berlin Heidelberg.

[172] Ur, B., McManus, E., Pak Yong Ho, M., & Littman, M. L. (2014). Practical trigger-action programming in the smart home. *Proceedings of the SIGCHI conference on human factors in computing systems*, 803–812.

[173] Ur, B., Pak Yong Ho, M., Brawner, S., Lee, J., Mennicken, S., Picard, N., Schulze, D., & Littman, M. L. (2016). Trigger-action programming in the wild: An analysis of 200,000 ifttt recipes. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 3227–3231.

[174] Utting, I., Cooper, S., Kölling, M., Maloney, J., & Resnick, M. (2010). Alice, greenfoot, and scratch–a discussion. *ACM Transactions on Computing Education (TOCE)*, *10*(4), 1–11.

[175] Van Kranenburg, R., Stembert, N., Moreno, M. V., Skarmeta, A. F., López, C., Elicegui, I., & Sánchez, L. (2014). Co-creation as the key to a public, thriving, inclusive and meaningful eu IoT. *International Conference on Ubiquitous Computing and Ambient Intelligence*, 396–403.

[176] Vasilescu, B., Filkov, V., & Serebrenik, A. (2013). Stackoverflow and github: Associations between software development and crowdsourced knowledge. *2013 International Conference on Social Computing*, 188–195.

[177] Vitali, I., & Arquilla, V. (2018). Developing a design toolkit for the Internet of Things.

[178] Vitali, I., Arquilla, V., & Rifino, I. (2019). Design for meaning of smart connected products. *Academy for Design Innovation Management 2019*, 1189–1202.

[179] Vitali, I., Arquilla, V., & Tolino, U. (2017). A Design perspective for IoT products. A case study of the Design of a Smart Product and a Smart Company following a crowdfunding campaign. *The Design Journal*, *20*(sup1), S2592–S2604.

[180] Vitali, I., Rognoli, V., & Arquilla, V. (2016). Mapping the iot: Co-design, test and refine a design framework for IoT products. *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, 1–3.

[181] Want, R. (2015). The physical web. *Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems*, 1.

[182] Wehner, P., Piberger, C., & Göhringer, D. (2014). Using json to manage communication between services in the Internet of Things. *2014 9th International Symposium on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC)*, 1–4.

[183] Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, *3*(3), 3–11.

[184] Wiener, N. (1949). Machine age. https://cdn.libraries.mit.edu/dissemination/diponline/MC0022/MC0022_MachineAgeV3_1949.pdf

[185] Wittgenstein, L., Bazzocchi, L., & Hacker, P. M. S. (M. S. (2021). *Tractatus logico-philosophicus* (1st ed.). Anthem Press.

[186] Wölfel, C., & Merritt, T. (2013). Method card design dimensions: A survey of card-based design tools. *IFIP conference on Human-Computer Interaction*, 479–486.

[187] Wright, P., Blythe, M., & McCarthy, J. (2006). User Experience and the Idea of Design in HCI. In S. W. Gilroy & M. D. Harrison (Eds.), *Interactive Systems. Design, Specification, and Verification* (pp. 1–14). Springer Berlin Heidelberg.

[188] Yamamoto, Y., Sandström, K., & Munoz, A. A. (2018). Karakuri IoT. *Advances in Manufacturing Technology XXXII: Proceedings of the 16th International Conference on Manufacturing Research, incorporating the 33rd National Conference on Manufacturing Research, September 11–13, 2018, University of Skövde, Sweden*, *8*, 311.

[189] Ye, Y., & Fischer, G. (2007). Designing for participation in socio-technical software systems. *International Conference on Universal Access in Human-Computer Interaction*, 312–321.

[190] Ye, Y., & Kishida, K. (2003). Toward an understanding of the motivation of open source software developers. *25th International Conference on Software Engineering, 2003. Proceedings.*, 419–429.

Appendix A

Ethical Consent

This appendix presents the letters of ethics approval obtained for the research outlined in this thesis, in accordance with the ethical approval process of the School of Computing and Mathematical Sciences. The documents included reflect the completion and acceptance of this review process, ensuring full compliance with the ethical standards of the University of Waikato.

School of Computing and Mathematical Sciences
Rorohiko me ngā Pūtaiao Pāngarau
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

Phone: +64 7 838 4322
Website: www.cms.waikato.ac.nz
Email: cms@waikato.ac.nz

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

11 February 2020

Tomás García Ferrari
c/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Tomás

Request for approval to conduct a user study with human participants

On the basis of the information you have provided on the SCMS Preliminary Ethics Application Form relating to your research "Interviewing IoT designers and developers", the committee has given you approval to proceed with your proposed study. The approval number is CMS-20-03, which you should include on the Participant Information Sheet.

We wish you well with your research.

Mark Apperley
CMS Ethics Committee Convenor
School of Computing and Mathematical Sciences

Faculty of Computing and Mathematical Sciences
Rorohiko me ngā Pūtaiao Pāngarau
The University of Waikato
Private Bag 3105
Hamilton
New Zealand

Phone +64 7 838 4322
www.cms.waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

2 March 2018

Tomas Garcia Ferrari
C/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Tomas

Request for approval to conduct a user study with human participants

On the basis of the information you have provided on the FCMS Preliminary Ethics Application Form relating to your research "Using conceptual cards to design mobile apps", the committee has given you approval to proceed with your proposed study.

We wish you well with your research.

Mike Mayo
Human Research Ethics Committee
Faculty of Computing and Mathematical Sciences

The University of Waikato
Private Bag 3105
Hamilton, New Zealand, 3240
0800 WAIKATO (924 528)

HECS Human Ethics Committee
Brett Langley
Telephone +64 77 838 4060
Hecs-ethics@waikato.ac.nz

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

22 July 2021

Tomás García Ferrari
Annika Hinze

Re: HECS Ethics Approval of Application HREC(HECS)2021#33 "From an Idea to a Prototype: assisting non-experts to prototype IoT projects"

Dear Tomás:

Thank you for submitting your amended application HREC(HECS)2021#33 for ethical approval.

We are pleased to provide formal approval for your project, including the following activities:

- Recruit approximately 10 adult participants for a single session survey, interview, and user observation study.
- Sessions will last approximately 2 hours, will be video and audio recorded, and take place in an office in G Block at the University of Waikato.
- Offer a $30 book voucher to participants for their participation.

Please contact the committee by email (hecs-ethics@waikato.ac.nz) if you wish to make changes to your project as it unfolds, quoting your application number with your future correspondence. Any minor changes or additions to the approved research activities can be handled outside the monthly application cycle.

We wish you all the best with your research.

Kind regards,

Brett Langley, PhD
Chairperson
HECS Human Ethics Committee
University of Waikato

The University of Waikato
Private Bag 3105
Hamilton, New Zealand, 3240
0800 WAIKATO (924 528)

HECS Human Ethics Committee
Brett Langley
Telephone +64 77 838 4060
Hecs-ethics@waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

16 December 2021

Tomás García Ferrari
Annika Hinze
Judy Bowen
David Brainbridge

Re: HECS Ethics Approval of Application HREC(HECS)2021#62 "Break This Glass"

Dear Tomás:

Thank you for submitting your amended application HREC(HECS)2021#62 for ethical approval.

We are pleased to provide formal approval for your project, including the following activities:

- Recruitment of up to 10 participants for a study that investigates assisting tools – such as custom-made playing cards and online tutorials – that could help non-experts isolate problems in IoT projects.
- Have participants complete a Qualtrics online survey questionnaire.
- Have participants follow the process of isolating a problem with an IoT prototype following a set of instructions (a bespoke deck of cards) provided by the researcher.
- Conduct a face-to-face semi-structured interviews with the participants.
- Participants may be audio and video recorded with consent.

Please contact the committee by email (hecs-ethics@waikato.ac.nz) if you wish to make changes to your project as it unfolds, quoting your application number with your future correspondence. Any minor changes or additions to the approved research activities can be handled outside the monthly application cycle.

We wish you all the best with your research.

Kind regards,

BLangley

Brett Langley, PhD
Chairperson
HECS Human Ethics Committee
University of Waikato

Appendix B

Case Study – Diary Excerpts

This Appendix presents selected excerpts from the digital diary maintained by Participant 2 (P2) during creating a bespoke IoT project, as documented in Chapter 4: Observing an End-User Developer. The diary, spanning 231 pages, offers a detailed account of the development process, including both successes and setbacks. The following excerpts were chosen to illustrate key moments and challenges in the project, providing a rich dataset for the analysis presented in the main body of the thesis. These records include textual notes, code snippets, and multimedia elements, offering a comprehensive view of the participant's journey through the project. The project's development days are labelled PD1 through PD148. People that P2 interacted with during the project are anonymised as Sn (e.g., S1).

Phase 1 – Learning

The participant began by mastering foundational components through Arduino and Firebase tutorials, building essential skills for the project. The following excerpts from the diary demonstrate how P2 focused on learning the initial components, beginning with the input device and database.

PD1 | *Do the first tutorials / projects with the Arduino Kit.*

PD2 | *Completed the first three example projects in the Arduino projects book.*

PD3 | (Firebase) *Started the tutorial, stopped due to not being able to install the Command Line Interface.*

PD9 | (Arduino) *No remaining projects to do.*

(Firebase) *Completed the friendly-chat tutorial.*

Phase 2 – Connecting

The participant transitioned to connecting the microcontroller to the cloud database using the Arduino MKR1000. Despite completing prior tutorials, integrating these technologies posed the first significant challenges. These diary excerpts illustrate the process.

PD10 | *Received an MKR1000 Wi-Fi board from S1.*

After searching for how to connect and MKR1000 to Firebase it's looking increasingly like I will be having to write a website to connect through.

PD12 | *Cannot work out at all how to connect Arduino to Firebase [...]*

PD13 | *Still cannot find a way to connect the Arduino to Firebase, I can see that S1 code on the Arduino forums is creating a JSON file but my very limited knowledge of web technologies is creating a brick wall.*

PD14 | *Spectacular, I might actually have made contact. I've never been so excited to see null.*

Through modification of S1 code I now have a MKR1000 that uploads sensor data to a Firebase database.

Phase 3 – Creating Visual Feedback

During Phase 3, P2 concentrated on creating visual feedback as an initial step in actuating, starting with simple tasks like blinking an onboard LED and advancing to real-time data visualisation on a website. Eventually, more sophisticated options like smart LED light strips were preferred. The following excerpts from the diary illustrate this progression.

PD15 | *Trying to get the Arduino Uno to blink its on-board LED in relation to the arrival of a new entry in the database then we'll have an actual IoT system.*

PD17 | *I have working webapp connecting to firebase and getting the information from the database though this is achieved via the slightly dubious method of being entirely unsecured with public security rules.*

PD20 | *Currently I can login and the page is getting the information from the database but it is not drawing the circles like I want.*

PD22 | *The divs and spans are being drawn, turns out they are at the bottom of the page and separate so it's just a contiguous block of attractive black text.*

Phase 4 – Working with Sensors and Microcontrollers

In Phase 4, P2 started integrating force-sensitive resistors (FSRs) with the Arduino MKR1000, initiating a phase characterised by overlapping and parallel progress. This phase spanned the first 40 project days and continued after the Summer break. The following excerpts from the diary illustrate this work.

PD23 | *FSR works on the Arduino Uno, now to port to the MKR1000, though I need to work out if connecting the 5v to the breadboard is correct or not when using a battery.*

PD24 | *Received a Multimeter from S1 which allows us to debug the WiFi board to see if the FSR is allowing enough current through to power the components.*

Can I power the FSR from pin 9? Update Yes I can, it works on battery so that I can squeeze the FSR and have it light up an LED.

Our current squeezer setup appears to use 141mA so the battery should power it for 8.5 hours.

PD25 | *Met with S1 and presented the solution to the wireless battery powered issue. This was quickly implemented as he only need to add the following code snippet for it to function as intended.*

Excerpts from the diary, second part of Phase 4:

PD46 | *Met with S1, received two FSRs [...]*

PD50 | *Add LEDs to the squeezer, 220 ohm resistor in.*

PD55 | *Went back to the learning about electronics example and tried to build it on another board with just one fsr, then two etc... turns out it works fine, my initial build had the transistor backwards. So now I have two fully functional prototypes.*

We will also no longer be incorporating the SparkFun into the box as the MKR1000 does the squeezers, LED and vibration motor. A nice clean package.

PD69 | *Squeezer code is still broken, have asked a question on Reddit in hopes someone can help.*

Fixed the squeezer code, It might have been the calls to Serial.println after the client.println, i'm not sure.

PD72 | *Turns out I had the power in the ground and vice versa, oops. Solved now, the prototype works fine.*

Phase 5 – Subscribing to Firebase

In Phase 5, P2 focused on subscribing to Firebase to create a responsive system that could react to database updates. This phase, from PD26 to PD37, involved experimenting with different approaches, including using the SparkFun ESP8266 Thing due to its Firebase library. Despite eventually achieving the subscription task, the solution remained unstable and required an inconvenient process. The following excerpts from the diary illustrate this effort.

PD26 | *He worked around the subscription issue by having the Arduino send an http request every three seconds. Whilst this works it's not ideal. The best bet is subscribing to the push notifications we can create from Firebase.*

PD28 | *Sounds like MQTT is the way to go. Lightweight, Publish/Subscribe options from a central hub.*

PD29 | *I did find Mosquitto which is an open source message broker, but cannot see whether it is possible to deploy this to Firebase.*

PD30 | *Met with S1, spoke about MQTT and the possible uses. It became evident that I need deeper knowledge of this before I can accurately assess the ways we can incorporate it into a project.*

Found cloudMQTT which as it's name suggest is a cloud hosted MQTT solution. Could be useful. Though it is rather difficult to use and has no examples or tutorials.

PD31 | *Spoke to S1, he has purchased the Sparkfun ESP8266 Thing Board so that we can try and make it work with a pub/sub model with our Firebase data.*

PD32 | *Did a frustration search for pub/sub for Firebase due to things not working. Lo and behold it does exist!*

PD33 | *ARTIK is the closest we've come outside of Firebase to having a useful prototype. MS and AWS could be useful but their desire to charge upon the end of a free trial is not particularly appetising.*

PD34 | *Started following this tutorial for ARTIK but realised halfway through that its using HTTP re- quests instead of MQTT. Waste of time.*

S1 came knocking with a gift, a SparkFun Thing ESP8266 so we should be able to get it working with the Firebase pub/sub library.

PD35 | *It looks increasingly likely that the SparkFun ESP8266 Things is not a supported board as trying to load the WiFi101 examples onto it causes the same errors to be thrown as the ArduinoStream sketch.*

PD35 | *Plugged in and tried to upload the FirebaseArduino sketch to S1' SparkFun Thing and I am receiving the same issue as last week.*

PD37 | *I tried simply adding the following code in order to have some feedback when the board is receiving from Firebase and big surprise, it's not connecting AGAIN and the computer is no longer registering it as a network device or whatever. This shit is flaky at the best of times. The simple act of uploading a different piece of code upsets the board so much it doesn't act like a WiFi chip.*

It's ugly and not ideal but it does work at this point.

Made a few videos of this setup. They're not long as I'm not sure the amount of detail to go into but there's some options to choose from.

Unplugging and plugging it back in is not a thing only stupid thing call center workers suggest. It might actually work.

Phase 6 – Working with Led Lights

In Phase 6, P2 focused on setting up LED lights as the project's actuating component. This phase unfolded from PD31 to PD46 and continued from PD53 to PD70. The following excerpts from the diary illustrate this phase of the project.

PD31 | *He is also interested in using the Phillips Hue for the output portion of the faculty project and would like me to find out if there is a library or tutorial or someway to use the hue from Arduino.*

There is an Arduino library here and here are some useful tips about using the Hue API.

Hue runs using a RESTful API but it is local so no internet access to it except through IFTTT.

PD36 | *Look for Phillips Hue compatible light bulbs as Hue bulbs are too expensive*

PD37 | *Phillips Hue: Has the gateway; Cree Connected: Can be paired with the Hue Bridge (as of 2015); [...]; LIFX Gen 3: Available in Australia, No bridge, just WiFi, Works with IFTTT, $74/Bulb.*

PD38 | *A Smart LED bulb and a Colour LED bulb are not the same thing. Colour LED bulbs are expensive. LED Bulbs are cheap because they are only white and dimmable.*

PD40 | *Working from home to find out what bulbs are compatible with the Hue Bridge.*

Zigbee 3.0 is the endorsed standard now so bulbs should be 3.0 compatible.

PD44 | *How to power bulbs without being in a socket*

PD45 | *Researched the bulbs in terms of ease of use, cost, documentation, other projects/hacks etc... Ranked the list by cost.*

Spoke to S1, added LED strips to the list as they could be more effective and cheaper.

PD46 | *The box is to be considered as having three squeezy balls and we will be using one length of LIFX Z LED strips.*

Excerpts from the diary, second part of Phase 6:

PD53 | *He has ordered the following: LIFX Z lightstrip; [...]*

PD61 | *Spoke with S1, received the LIFX Z kit and 3 more FSRs.*

I brought the LIFX Z and an ESP8266 home to test with on a computer that isn't stunted.

PD62 | *Have been working to try and get the connection to the LIFX working by using the code from the MKR1000 that builds and sends JSON files.*

PD63 | *S1 realised that we could do the LIFX API call using JavaScript functions from Firebase.*

PD64 | *I have been searching tirelessly for any semblance of actual information, no one has responded on the posts on arduino or LIFX but I realised I could post on reddit as well. I got a response very quickly about adding a / to the URL, no result.*

PD65 | *Sending requests through Postman to the LIFX API works but the mock server doesn't seem to work, or more likely, I'm not sure how to effectively use it.*

PD66 | *Attempted to run the JSONTest headers API again to no success.*

PD68 | *S1 came and set up a web server on his laptop, we then had time for one test fire to get the JSON and the connection connected, measured the length of the JSON, then began connecting again and promptly failed.*

PD70 | *Success! The ESP8266 is able to change the colour of the LIFX Z Strip.*

It looks like all it needed was the request to be encoded as a single string.

Phase 7 – 3D Printing

In Phase 7, P2 began designing and 3D printing a box to house the project's electronics and support the rugby stress ball. This phase, spanning PD49 to PD72, involved iterative trial and error due to the slow printing process. The following excerpts from the diary demonstrate the steps P2 took in this phase.

PD49 | *He sketched a basic idea for a box we can 3d print for me to model.*

PD50 | *Continue work on the boxes and tongues.*

PD51 | *Talked to S1, tongues not necessary, we want spikes to hold the balls.*

We discussed the box design and ball spacing, we will print the box in two halves in order to maximise the space as 22cm is not enough for the squeezy rugby balls to sit comfortably.

The box will be 25cm long, 4cm high and 7cm deep with a separation in the middle which will be joined using tongue and groove as well as glue. It will have a slide out bottom in order to access the internal working and might have Bauhaus style rounded edges.

This we will 3d print.

PD52 | *Continue work on the boxes and tongues.*

What material are we using? Wall thicknesses.

PD53 | *The pieces are a bit snug but with some working they move. They seem to all fir together fairly nicely, thought the curved corners have not worked as the printer has filled them in for support.*

PD54 | *Showed S1 the completed slice test print of the box. The tongue slides in and out, with some effort, but it works as a concept. He was immediately concerned about the amount of space for the components, as was I.*

PD55 | *Adjust 3d model dimensions for printing - partial*

PD56 | *Adjust 3d model dimensions for printing - partial. Create spike solution for box.*

PD57 | *Print test SpikeSolution.stl for box - in progress.*

Designed a couple of different spike assemblies/ attachment systems for the box to hold fast to the squeezy ball.

Took them to S4 to get them 3d printed and he said that the spike setup is too brittle. The tongue idea could work, though he suggests that the FSR should be contained [...]

PD58 | *Print test SpikeSolution.stl for box - in progress*

PD59 | *Got the 3D printing back. The spikes are worthless, there's no strength to them at all. Utterly unusable.*

The tongue looks to be the way forward, though I have made the head too big to fit through the hole.

PD60 | *Increased the size of the portal for the FSR and the tongue, deepened the slot for the tabs to fit into.*

Went to see Phee about printing half the box and a tongue, the printer is engaged until Wednesday. Will print them then.

PD61 | *Still waiting for the 3D printer.*

PD62 | *Handed the files to S5 and S4 for 3D printing.*

PD63 | *Adjusted box tongue for strength at S4's suggestion (added a scalloped curve to strengthen the 90degree which is very weak).*

Handed the files to S5 and S4 for 3D printing again.

Spoke to S4 yesterday afternoon and he made some very good suggestions about the design of the box.

The taller something is, the longer it takes to print, so lying it as flat as possible is advantageous [...]

PD64 | *I sent the modified file to S4 at 1:30am but apparently he didn't get it [...] Not ideal. Have resent it, I imagine that the box will be printed on Monday.*

Box has been printing since about 11am, neat.

PD65 | *Met with S1, he is happy with the look of the partial box with tongue and ball.*

PD67 | *The second half of the box was completed last night, I have filed and fitted the two halves together. Very disappointingly it appears that the filament contracts slightly on cooling (logical) and this really emphasises the join in the boxes. It's definitely not invisible.*

I don't think we need to split the sliding lid pieced at all as if we decrease the length it has to cover by 50mm we could print it at the printers max of 220mm. This would simply mean that we add a lower piece to obscure what the lid doesn't cover and for it to butt against.

PD72 | *Got the printed lid, I forgot to adjust the offset for the split.*

It's a rough fit, tight and noisy and gets stuck on the middle split, but it's a lid.

Phase 8 – Facing DB, Microcontrollers and Light Problems

In Phase 8, P2 encountered significant challenges across all project areas, focusing on resolving issues with the database, microcontroller, and LED lights. The following excerpts from the diary illustrate P2's intense problem-solving efforts and the frustrations experienced during this phase.

PD71 | *After tinkering with the prototype for a while it appears that for the ESP8266 to function properly the squeezer prototype must be connected and powered before the ESP8266 prototype.*

It sometimes outputs weird errors [...]

PD73 | *Trying to solve the speed issue, have removed the reading of the response, this makes the time faster however the stack is still printed after every interaction.*

I am trying to use the yield() function which allows the necessary bg functions to run and maintain connections and the Wi-Fi stack etc... It doesn't seem to have any noticeable effect.

Tried to redeploy the website with a function for testing and now the fucking esp8266 is creating events all by itself again and the order of operations fix doesn't work anymore!!!

Tried redeploying the website again to fix the issue I think is created by the first redeploy and it fails like it did the first time.

PD74 | *Changed the rules to [code snippet]. And that has stopped the put-pox but it has also stopped all input. Progress I suppose.*

The squeezer prototype is still able to upload to firebase, so that's not broken. It is just the ESP8266 (the eternal problem) and the website.

PD75 | *Plugged the system back in and it went but it started to put-pox when I had deleted most of the entries from the database. When I just spammed squeezes from prototype 3 it stopped put-poxing.*

In order to try and solve this problem I tried converting all the calls to delay() to delayMicroSeconds() as per this page. All this achieved was the following error constantly: [code snippet]

Looking at the errors and solutions on this page again, I found that the section about the WDT errors is relevant as it says avoid long loops and I have a while loop that waits for a header response, so this could be causing the disconnection.

Looks like the startStream() call at the end of the sendToLIFX() method was unnecessary as disabling it has stopped the WDT resets and stack errors, somewhat. We are still receiving exception 28 intermittently.

PD76 | *Changed the devices to upload to /devices/device_NUM and have the ESP8266 listening to that /devices bucket, now it should hear events from all devices.*

Changed the getString calls to the following in order to receive the data from firebase, no dice. Will just have to rely in the JavaScript functions however that is going to work.

I have no idea if this will work or how to get hold of the current light data.

PD77 | *All of these end with an error that says update, push, set and add are not functions.*

Unsure what type of data the update function wants. Found the reference here, though it is woefully lacking in information.

PD78 | *Posted a question to Reddit about firebase JavaScript functions.*

trying to get the JavaScript functions working

Found out that the entries in the database are called nodes, so using this new search term, I've found some relevant code on stack exchange

With this new found knowledge I have constructed the following: [code snippet]

It no longer throws errors but it also doesn't write where I want it to either. Progress?

PD79 | *There is a response to my query on Reddit, it was suggested that the compiler is unsure what database is and so does not recognise ref() or push().*

I'll give it a shake and see what falls out.

It looks like it actually did what I wanted, in part.

PD80 | *Assistance from Reddit again suggested I need to be using event.data.val() which I am already using but they did link to this page which I am trying to use.*

It is obvious that I need to be accessing the data of the specific child but how to get in there still eludes me.

PD81 | *Tried the code from here to see if it works, if so I should be able to adapt it, hopefully.*

Turns out the problem is that it was pointing at devices where it needed to be pointing a layer deeper devices/device_NUM in order to capture the latest entry.

We don't need to query the current state, there is an endpoint called state delta that adds the new value to the current [...]

We still need to get the emotion and pressure from firebase.

PD82 | *It works, however it keeps going and adding calls as it fires on every write. Recursive doy.*

Just trying to create the appropriate function to pass the data to. Yeesh JavaScript, why you so ugly.

PD83 | *Lo and behold, its now broken again. What worked yesterday to get the single object now does not and is again telling me that limitToLast is not a function. How wonderfully fulfilling to be constantly working against a tide of shit.*

Looking at this conversation it looks like I need either snapshot.child() or snapshot.val().

As useless as JavaScript and JSON are they continue to amaze me.

And this just turns itself inside out and refuses to cooperate.

I'd love to punch the idiot that created JavaScript.

PD84 | *And yet it still doesn't work. The Emotion println is still empty.*

Nope, still failed to convert to string.

I have tried: [code snippet]. And none of these have worked.

Just changing the PUT to POST doesn't solve the issue, now it returns 400 Bad Request.

PD85 | *Turns out the PUT/POST issue was simply not having the POST wrapped in "" quotation marks.*

The solution works, it is additive using the setState endpoint.

PD86 | *So dividing the total number of presses by 255 doesn't work as they inevitably get stuck at about 90 as the total available is 255 so they can only get there with nothing else being activated.*

Using this approach the colours can go well outside the allowable range (e.g.: 385.5) so this is not the way to do it.

PD87 | *So as per the end of yesterday, the more squeezes, the smaller the effect on the light produced.*

Looking again, it might not be two different objects, now it appears to be playing ball.

Still have to build the new brightness logic and investigate the multi obj light value problem and fanagle the colour logic.

PD88 | *So have tried the following and it doesn't work as the value is always out of range.*

IF YOU WANT TO DEBUG THE API YOU WILL NEED TO TURN OFF FAST AS IT DOESN'T RETURN FULL ERRORS AS FAR AS I UNDERSTAND. THIS IS QUITE A LOT OF UPPERCASE NONSENSE.

So no for some reason the lightstrip becomes entirely unresponsive. I'm going to remove the fast variable and see if that solves it.

Removed FAST and the bastard light stalls after 10 maybe a few more calls to the API, so either I'm being limited or something's broken.

Now nothing works, not squeezing, not the reset button, nothing in these prototypes.

It stopped responding...

I am disturbed.

Hard reset the system (unplugged everything) and it began to respond again.

PD89 | *I think to two object problem is exactly that, multiple objects that are created inside each of the functions. By crating one global object I think I should be able to circumvent this problem.*

That doesn't work, it is still passing completely separate objects.

The above does actually work, the two object problem is solved, though the reset sometimes doesn't reset and when a reset is pressed it sometimes limits the system to interactions only from that device.

PD90 | *Trying to convert the timestamp from epoch to local time so that I can use the time of day to inform the amount of saturation to apply.*

PD91 | *Have tried setInterval(function (obj) { in order to pass the obj through to the method so that it can be pushed. Unsure if this will work, worth a try.*

Is this a SparkFun board issue, MKR1000 issue or a JavaScript functions issue?

PD92 | *I have added a console.log() to the setInterval outside the if() to continues to write to the console. This should show me that the setInterval function is working constantly.*

So I was right, the setInterval() is still functioning, so the value is getting reset somewhere.

So it looks to have stopped at 1609. Why would it disconnect after 3 hours?

PD93 | *Updates the update so that it is (hopefully) returning a promise.*

Could be too many entries in the database. Something to keep in mind.

So there musn't be a Wi-Fi connection. Did the MKR1000 go to sleep?

It wasn't logging to the console either so does the JavaScript function sleep after a time of inactivity?

That manages to deploy so lets see if it makes any great difference on Monday.

PD94 | *Test the system longevity to see if the promise in updateDatabase() is working this time.*

I'm really thinking this is the JavaScript function that is causing all the issues as it is not pushing to the console. But squeezing the squeezer wakes it up again. However the SparkFun Thing is also not responding when I squeeze the squeezer.

I've reset the SparkFun (physical reset, turned off and on again) to no change.

Everyone of the Arduino updates breaks my code until I update, it's an awful setup they have. Find the issue, the SparkFun is unresponsive with the current CODE. I don't think the MKR1000's are the source of the issue.

PD95 | *The SparkFun Thing has completely stopped listening to the realtime database.*

Looks like that SparkFun Thing no longer responds to Wi-Fi.

Squeezing the squeezer the whole system responded. So my guess is that it is the JavaScript function that is stopping.

The goddamned timestamp never updates, how useless.

PD96 | *Don't upgrade to beta libraries, accidentally upgraded the arduinoJson to a beta release and it threw errors on working CODE.*

Almost got the system back up to where it was a month ago. Still not sending to the lightstrip...

Have reconnected the lights to the WiFi. It took much more effort than it should have, sometimes the WaikatoIoT is flaky.

It works, the beast lurches back in to life.

Phase 9 – More 3D Printing

In Phase 9, P2 focused on printing additional boxes to enclose the electronics and attach the stress rugby balls, continuing the work from Phase 7. The following excerpts from the diary demonstrate P2's limited involvement in this phase, primarily overseeing the 3D printing process while addressing issues from Phase 8.

PD73 | *Get S4 to print more boxes when he's available.*

PD75 | *Get S4 to print more boxes when he's available.*

PD78 | *Received another box from S4.*

Phase 10 – Putting Electronics in the Box

In Phase 10, P2 focused on placing the circuit into the enclosure, though the process revealed the need for further reorganisation. The following excerpts from the diary illustrate the challenges and decisions that led to considering a custom PCB design.

PD91 | *Boxes need to be sanded and glued to get best fit.*

Electronics needs to be fitted to box to get wire lengths.

Need to work out where and how to fit the reset buttons to the box.

PD97 | *Further things to do:*

– Assemble the boxes (how do I safely use Epoxy?)

– Solder the circuits to fit the boxes (need an electrical engineer, luckily I might know one.).

PD98 | *Took the poorly printed box and the epoxy home on Monday night and glued the box and lid this morning. It will take 16 hours before it is at full strength.*

PD99 | *The box and lid seem to be fairly sturdy, though the epoxy leaves dirty marks where it is visible.*

Do we want the tongues permanently glued in place? It's more useful if they can be removed but they do have the tendency to pop out under pressure is simply friction fitted.

Further things to do:

– Assemble the remaining boxes, lids and tongues

– Solder the circuits to fit the boxes (need an electrical engineer, luckily I might know one).

PD100 | *Epoxied the two remaining boxes and lids, S1 responded about the tongues, they are to be epoxied in also.*

PD101 | *Made a Fritzing diagram of the competed squeezer circuit. I thought I had done this, apparently not.*

Phase 11 – Creating a Custom PCB

In Phase 11, P2 collaborated with S2 and S3 to create a custom PCB, which proved essential for the project's progress. The following diary excerpts highlight the development of this phase and the contributions of these key helpers.

PD102 | *Met with S2, the electrical engineering undergrad I know. He had some suggestions for soldering the circuit.*

He suggested that designing a PCB that we could solder cables and components to would be much easier. He designed the PCB and has some suggestions about who to print them with.

PD103 | *Ordered 10 PCBs from JLCPCB*

PD104 | *Printed the PCB layout from Fritzing and set it inside the prototype box in order to begin working out wire lengths and layouts.*

I realise that I should have created more connections to more of the pwm plugs so that I could pick and choose which plugs to use. Live and learn.

I've also neglected to add holes for the standoffs so I will have to resort to drilling them manually.

It looks like I might be able to drill holes into the side of the box for the standoffs to screw into but I think it would be a better idea to adhere them somehow. To avoid accidentally punching through.

PD105 | *PCBs have shipped and will be here next Wednesday according to DHL.*

Talking with S2 I told him of the reluctance to solder the MKR1000 boards and the need to run wires, he suggested getting some stackable headers and soldering those to the PCB, this way we can sit the MKR1000 solidly on the PCB.

PD107 | *Received the PCBs yesterday evening, went out this morning and got some stackable headers and some additional wires.*

Trying every possible configuration and position has yielded no result, the stack is just too high. Disappointingly i'll have to separate the two and run wires from the MKR1000 to the PCB and on to component where required. Messy.

I forgot to add holes in the PCB for standoffs so either I reorder new PCBs or just drill holes in the current stock.

Instead of costly redesigns, I think I could just adhere some nuts to the inside of the box and screw through the PCB into those, this way I don't have to drill into the sides of the box, this lowers the height of the standoff (improvisation for the win!) and should allow me to use the stack thereby lessening the number of wires that will need to be run about the box.

PD108 | *Drilled 4 holes in the PCB so that I can pass standoffs through it to hold it in place.*

PD109 | *S3 was kind enough to help me with the soldering. Specifically he soldered the first board as I watched and he talked about what he was doing. This has resulted in a board that is completely assembled.*

Testing has revealed that the MKR1000 is not even available over WiFi so something is shorting it as when it is removed from the PCB it is immediately available.

We found that that a trace was running over the reset pin for the arduino which could have been resetting the board constantly.

PCB design is an art form and quite a complicated problem.

S3 scratched the trace out to break it and then ran a "bodge wire" to jump it to where it needed to be. This worked and the board now functions. Huzzah.

Phase 12 – Seeking Stability

In Phase 12, P2 refined the system to achieve stability and consistent functionality. The following diary excerpts illustrate this focus during the project days.

PD109 | *Secure the vibration motor to the board permanently.*

Refine the communication with the LIFX Z Light Strip.

PD110 | *I've been trying to work out a solution to the foot problem for the board as the prior solution of gluing down a nut and screwing a standoff into it won't provide enough height due to the bodge wire on the underside.*

PD111 | *Last night I Epoxied the board feet and adhered them to the inside of the box.*

This seems to have held and testing this morning reveals that the system still works, huzzah!

PCB Modifications: Fix the traces running over one another; Rotate the connection for the LED by 90 degrees [...]; [etc.].

Box modifications: Reduce the size of the holes as the balls don't really sit down into them as I had envisaged. Make the box deeper to allow the use of unmodified standoffs. [etc.].

It would appear that the stalling that causes the lights not to respond is nothing to do with the lights, it's the SparkFun Thing that is stopping.

Trying to add a manual reconnect button to the firebase receiver so that I can reconnect it when I think it has lost its connection. Good for debugging.

PD112 | *Pin 4 on the SparkFun Thing doesn't go for some reason. I could be initialising it incorrectly but setting it as output and setting it to high to light an LED doesn't work. That's a bugger as that leaves me with only 2 I/O ports.*

Fixed the red problem from yesterday, I had accidentally commented out the assignment of the R, G and B. Colour works again.

Still having the irritating error where things stop responding. Even the reset button on on the receiver doesn't get the lightstrip responding again. Further testing required.

PD113 | *Exception(29) of the SparkFun board is almost entirely random with the period at which it occurs. Sometimes after 7 connections to the LIFX API, sometimes after 15, but so far the connection hasn't been lost whilst the serial monitor has been open.*

PD114 | *Realised that I could use one of the machines in the grad lab in order to access the esp exception deCODEr that this computer can no longer access.*

The computer in the grad lab doesn't recognise something about the library and keeps throwing an error about using a deleted function (firebase.getEvent()) and the arduinoJSON library doesn't update properly.

[...] ratnick said that he needed to reduce the size of a variable that was allocating too much space. I set the staticJsonBuffer to 1000 so could this be the issue, that I am sending messages that are too long?

I think this exception error is the main problem, it disconnects the board and if the serial monitor is not open it doesn't reconnect. So once his problem is sorted then the system might work longer than a couple of hours.

PD115 | *I thought that the unreturned promise in the js function may have been the cause of the esp exception so I set about watching both the function log and the serial monitor to see if there was a correlation between the errors.*

The esp exception doesn't occur when the saturation is unable to be decremented, so it must be the decrement or something in that function that is causing the issue.

This is the function that is causing the error to say "Function returned undefined, expected Promise or value". So I am still doing this wrong as I thought this was returning a promise.

So I've still got absolutely no idea what this issue is caused by honestly.

PD116 | *Added a 5 second timer and listener that monitors the SparkFun board. If the Wi-Fi is disconnected for more than 5 seconds it will auto reconnect, I hope.*

The lights are not responding currently. Irritating, might be due to order of connection or some other thing.

PD117 | *The system is responding today, no idea why it wouldn't yesterday. It doesn't seem to freeze and become unresponsive so maybe the reconnect function is working.*

Opened the serial monitor and squeezed, once the serial monitor says it is sending the issue is solved. The SparkFun is clearly the entire issue.

It turns out that lost connections are not an uncommon issue and the problem appears to deeper than something I can fix, it has to do with the library itself and Arduino. However some people do have workarounds that I am going to try.

PD118 | *Have added Firebase.stream("/processed") into the the start of the loop so that the board is constantly refreshing its connection. We'll see if this mitigates the exception.*

Well that's a fail, it still gets exception 29, On the very first 5 minute decrement it threw the error.

So now I am entirely out of solutions.

Have disabled the call to setInterval and will see if the exception rears it's ugly head now.

That's a big fat nope. It still gets the error even without the setInterval. Well at least that eliminates the setInterval() and updateDatabase() functions.

My only remaining thoughts are yesterdays information about the problem being within the library and arduino.

If we could just pay for a Firebase subscription plan then we could sidestep the issue of the spark- Fun exception error completely as we would be able to connect to the LIFX X API directly from the JavaScript function in Firebase.

Small victory though, I realised the other day that I had been trying to build the reset button CODE for the SparkFun receiver in the bottom of the send method instead of in the loop. Now the reset button works.

PD121 | *Posted the CODE and error on Reddit out of exasperation. Hope I can get some intelligent help.*

PD122 | *I have no idea but these SparkFun boards are anything but reliable.*

I put a Serial.println() in the if(firebase.available()) block [...]

I've added a while loop to wait while firebase is not available and it never connects. Joy, something is no longer working in the connection.

I also tried flashing the board with the old CODE that doesn't have the reconnect button in it. That didn't work either. This might be a firebase issue as the board says it's connected to the Wi-Fi.

PD125 | *Also still having the Reddit conversation about the ESP8266 though that board no longer seems to connect to Wi-Fi even though it says it is. I will have to test it with a basic Wi-Fi sketch.*

Tested the ESP8266 with a trimmed down sketch that just randomises the RGB values and pushes them to the LIFX API, this works flawlessly so the connection to firebase is what the problem is for this board. They must have changed something on their end as the library is the same version that has been working for me.

Looks like Firebase has a new Fingerprint and this is why I've been unable to connect.

Firebase informed me that the rules I have been using are public [...]

Changing the rules to require auth stops all writes to the database.

Asked for help on r/Firebase and r/Arduino on Reddit for how to authenticate from a MKR1000.

The REST API seems the most straight forward and should allow each device to authenticate individually.

It seems to work. I am surprised. I wonder if this stops outside shenanigans from getting into the database.

PD126 | *Oh well, check it off the list of possibilities, it must be the fingerprint.*

I'm going to try and reflash the SparkFun on a windows machine here in the lab [...]

So the serial monitor on windows throws nothing but garbage at me so I've been walking the accursed SparkFun back and forth across the lab.

So I reflashed the board with the CODE on the mac which includes the fingerprint verification again and it just sits idle, doesn't even get the exciting connection and then stream error. So the fingerprint is absolutely the issue but none of the available workarounds work, apart from one that uses PHP to auto update the fingerprint, which I have no experience with.

Phase 13 – Using Firebase Functions

In Phase 13, P2 revisited and implemented the idea of using Firebase functions to control the actuators, replacing the unstable second microcontroller approach. The following diary excerpts show the progression and implementation of this solution.

PD63 | *S1 realised that we could do the LIFX API call using JavaScript functions from Firebase.*

PD118 | *If we could just pay for a Firebase subscription plan then we could sidestep the issue of the SparkFun exception error completely as we would be able to connect to the LIFX API directly from the JavaScript function in Firebase.*

PD133 | *Looking at how to call the LIFX API from JavaScript has led me to this website which converts cURL to node.js. I'm not sure if this is the right direction but it is a starting point.*

PD134 | *Use the Lif X page by Pascal Moser to generate the exact cURL you want, then put it into the page by NickC to get the node.js CODE. This might be folly but it's worth a try.*

PD135 | *Using the page by Pascal Moser and the page by NickC in concert with the JSON.stringify reference I have managed to create the following.*

Not yet working, it's not even logging anything to the console.

Looks like I need to install the request nodejs module which is outside my control. I have emailed S6. Once this is done I will test to see if this works.

PD137 | *S5 managed to get the module installed and now the function is churning out all matter of interesting stuff, JSON looking objects and new errors. Again, work for Monday. As it sits, the function is still not talking to the LIFX API, so there's something to investigate.*

PD139 | *Trying to get the function to talk to the LIFX API.*

This works, so why didn't it before? I'll try again with the saturation added.

Connecting to the API consistently is difficult.

PD140 | *So it looks like the error from the LIFX API is due to decimal places. To combat this I have implemented two rounding methods (one for the saturation which require one decimal place and one for the RGB values which need to be whole numbers) for the values generated by the JavaScript function to keep them nice and tidy.*

These works and now the colour is working properly.

PD141 | *Trying to get the function to read from the array and generate the colour.*

The JavaScript function and firebase had a bit of a brainfart and refused to show me my data and refused to delete entries. I realised that maybe closing and re-opening the tab was a good idea and it worked. Weird.

Now I am testing if moving the following commands outside the else in the process function will work to stop the nuisance additions to my array.

Nope, doesn't work. Now it's only listening to the last entry. Pointless.

PD142 | *Trying to isolate what I broke on Friday.*

I have commented this out and the colour responds again.

Found this about array.shift() and array.pop() when removing the first and last items from an array. Will see if I can leverage this to my advantage.

I was correct, the colour values are retained and the saturation is decremented.This also still works with all three boxes.

PD143 | *Check the log to see if the setInterval was running overnight.*

Yes the function runs all night.

Watching the firebase promises videos again, I've tried return null; and that also doesn't work.

I've posted on Reddit looking for help with the promise. Let's hope someone helps.

PD144 | *Spoke to S1 about a possible security issue as it looked like someone was accessing the database at 4am. Turns out that the database doesn't log using 24hour time, so the 4 was 4pm the day before.*

PD145 | *I am trying to get the js function to send white when the numSad, numMeh & numHpy values are equal.*

Testing with the LIFX API documentation shows that any value that is shared by all three channels creates white as does the keyword white, so why it is creating red from the function is perplexing.

It really doesn't like the calculation so now I am trying to circumvent the red plague by adding an if else to the send() function in order to send a string containing white to the API.

Further investigation yields that sending saturation with white causes an error that is not being returned to me through firebase. On thinking about this, saturation for white is ridiculous. So on removing the saturation part of the JSON the white now works exactly as expected. No more red plague.

Phase 14 – Creating Haptic Feedback

In Phase 14, P2 worked on adding haptic feedback to the input device using a vibration motor, ensuring successful interaction. The following diary excerpts illustrate the steps taken to achieve this, including overcoming challenges with solderless connections.

PD119 | *Talked to S3 about JST plugs for the board so that we can make the vibration motor removable. He suggested digiKey, element14 or RS Electronics.*

PD120 | *Noticed that the SparkFun boards also have a JST lipo connector, but this one is right angle through hole. Spoke to S5 and he worked out that it is a JST PA and found it here at RSElectronics. Bonus is that RS have a distribution node here in NZ so I could have them here in a decent amount of time. 48c each, not bad.*

I've laid out a small piece of paper to outline how I think the JST could fit. The over lapping is fine as that is where the wires connectors are and so should be able to extend one to fit into the off kilter hole.

PD123 | *Spoke to S3 about connecting the vibration motor to the connector housing and he pointed out that what was missing was the crimps. The metal part that attaches to the wire and acts as the contact.*

PD124 | *Emailed the crimps to S1 and asked him to purchase them.*

Might also need to get a crimping tool.

PD125 | *Tested to see if the header would fit on the board. It does, just needs a 45degree twist to the left so it doesn't encroach on the contacts for the MKR1000 headers and faces off the board.*

Crimping tutorial

PD127 | *I soldered the header for the vibration motor and my improvised headers onto the next PCB. Testing seems to show that they aren't shorted and are passing current to the right places. This looks like a good solution.*

I have hot glued the header for the vibration motor down as I didn't want it wobbling around when plugging and unplugging and breaking or damaging the soldered joins.

This or this could be the way to go for a crimping tool. Need to investigate and determine which will work.

PD128 | *Received the crimps, tried the crimping tool I purchased from Jaycar and big surprise it's slightly too big.*

I crimped using the tool and it can not squeeze it hard enough and the wire pulled out leaving the crimp in the housing.

I then re-crimped the same crimp (additional pressure) and could still pull it out so put it in the crimp tool upside down and it mangled the crimp. It no longer clicks into the housing but it isn't releasing the wire.

PD129 | *Looking at the crimping tool it says it should be able to crimp 26 - 28AWG and my crimps are 28AWG. So I tried again with the crimping tool and got lucky with one crimp that fits.*

Testing with the Multimeter has shown that there is voltage passing through it and also when it is plugged in to the board. Success.

*So trying again, the next three attempts failed as I think I was over squeezing them. I know this because when I insert them into the housing there is no *click* and they pull out easily.*

So I kept that part of the crimp out of the crimping tool in an effort to not squash the tab. It seems to have worked and that wire also passes voltage, so sayeth the multimeter.

Have managed to Crimp the larger wires in, I'm unsure if either of these are a good crimp or connection but I now have two options.

PD130 | *Found a a JST crimping check sheet.*

Phase 15 – Building More Input Devices

In the final phase, P2 focused on assembling additional input devices but encountered some errors despite previous experience. The following diary excerpts highlight the steps taken and challenges faced during this phase.

PD131 | *Soldered all the components to the second PCB. Pretty straight forward.*

Made a mistake with the headers for the MKR1000 though. [...]

I remembered that S3 had removed the shroud for the pins in the header accidentally and was able to remove just the pin we wanted instead of the whole header. [...]

Testing has revealed that this solution has worked. Yay!

I have also soldered the vibration motor onto the JST header and wires, this solution also works.

The headers for the FSRs also work, so now we have another working board and removable components. Huzzah!

I tested the two systems at once and they both feed into firebase correctly and increment the colour as expected.

PD132 | *Built the third circuit on a breadboard to test that the components work.*

I did test the latest and found that when the plugs for the FSRs are plugged in, the wires rub against the lid and it leaves a mark on the wires. I don't think this will be safe over time. How can I solve this?

– I could desolder the headers and solder the wires to the board like this first box but then that makes it very difficult to plug in the FSRs.

– I could get some vertical headers and housings to create plugs with a lower profile but that would require headers, housing and crimps all over again.

– I could just leave it as is and hope for the best.

PD136 | *The issue of the headers projecting the FSR wires too high was sorted by S1 suggesting that I desolder the female headers and replace them with short female to male wires, this way we will be able to route the plugs for the FSRs wherever we want them. This works, even though soldering wires together is fairly precise and finicky work.*

Testing has also shown that the thinner gauge wires used for the vibration motor is not getting a solid enough connection to the crimps and the motor doesn't vibrate unless they are pressed into the metal. Either I pull them and replace the wire with the thicker gauge or I use some needle-nose pliers and squeeze the wings down onto the exposed wires to create a better connection.

PD137 | *I have removed the low gauge wires from the vibration motor, crimped some thicker wires and now these wires are set to be soldered.*

I have soldered the final board (box 1) and have solved half of the LED issue, the blue and green work as they should but red is still not working, I think this could be to do with the pin on the MKR1000 itself. I'll have to investigate this on Monday.

PD138 | *Over the weekend I epoxied the third PCB into Box. Testing today has revealed that the LED is working as expected after the re-solder of the inverted blue, green wires.*

All circuits and MKR1000s are working as expected and the LEDs are now set to glow green when the FSRs are released as suggested by S8 and S7.

PD140 | *I have managed to successfully re-solder the vibration motor. It seems like hey used either hot glue or something similar over the pads to hold the soldered connection together. The simple solution would be to use thicker gauge wire, why they have insisted on using this fragile nuisance wire is beyond me.*

PD143 | *Looks like the Epoxying has worked, the lids still slide and the boxes don't rattle as much, still thunderous mooing from the herd but all assembled now. Except for the device to limit the Micro USB from being yanked out of the MKR1000.*

On the subject of cable routing, I think these command clips could be useful, or these self adhesive clips from Jaycar.

If they are placed at a right angle to direction of travel of the cable it could help to minimise the direct pressure applied by someone yanking on the cable.

PD144 | *S1 suggested that we hot glue the balls to the box to add stability, I suggested Epoxy on a spare ball and on the waste parts as a test.*

PD145 | *On Thursday I need to epoxy the balls to the boxes.*

PD146 | *Have cable tied all the remaining boxes, the cables appear to be solid.*

PD147 | *Epoxied the balls to their respective boxes. I had to re-cut a red one that had the uni logo up instead of down. Once attached I discovered this sensor no longer* *responds, turns out the ball is too tight and when the pressure is released the sensor works. I'll have to insert something in the ball to release the pressure. [...]*

PD148 | *All boxes epoxied and working, though the ugly box (Box 1) might discombobulate itself due to the sub par epoxying of the first two balls.*

Appendix C

Semi-structured Interviews

Here, we present the questions used in the semi-structured interviews that are detailed in Chapter 4, Chapter 6, and Chapter 7. These interviews were designed to explore developers' experiences, challenges, and strategies when working on Internet of Things (IoT) projects. The semi-structured approach provided flexibility, allowing the interviewer to adapt the discussion based on participants' responses while ensuring that key topics were addressed. The questions served as a guide to facilitate in-depth discussions about the barriers, tools, and practices in IoT development, with a particular focus on identifying the obstacles that non-expert developers face when creating custom IoT solutions.

Iot Developers
Semi-Structured Interview

About the project

Q1 | From your perspective and in your own words, can you describe the project?

Q2 | What were the motivations to do this project?

Time, Learning and Phases

Q3 | If you can measure it – how long did the process feel?

Q4 | Was there a bit of a learning curve? How steep was it?

Q5 | Were there different phases that you can identify?

Q6 | Were there any stressful points during the process?

Q7 | Were there any surprising things that you encountered when doing the project?

Failure and Successes

Q8 | Can you identify failure moments?

a. Did you at any moment think "this is not going to work..."?

b. And how did you solve it?

Q9 | Can you identify successful moments?

a. Were those moments where you thought, "Yes! I've got it! This is it!"?

b. Can you explain why?

People

Q10 | Were there people involved that helped you during the process? How often did you need to contact them?

Suggestions

Q11 | In hindsight, can you identify a list of things that you would have done differently?

Q12 | What would be your advice for someone that needs to start a project as you did?

Iot Developers

Semi-Structured Interview

Simple or Difficult

Q13 | How do you evaluate the process in terms of difficulty?

simple | regular | average | complicated | super complicated

Q14 | How frequent would you qualify the failure moments?

really frequent | frequent | average | not so frequent | rare

Q15 | How frequent would you qualify the success moments?

really frequent | frequent | average | not so frequent | rare

Follow up ideas

...and what happened next?

...who else was involved?

...and how did you solve it?

...can you explain why?

...and how did you overcome those moments?

FaltaP

Semi-Structured Interview

About the project

Q1 | From your perspective and in your own words, can you describe the project?

Q2 | What were the motivations to participate in this research project?

Time, Learning and Phases

Q3 | If you can measure it – how long did the process feel?

Q4 | Was there a bit of a learning curve? How steep was it?

Q5 | Were there different phases that you can identify?

Q6 | Were there any stressful points during the process?

Q7 | Were there any surprising things that you encountered when doing the project?

Failure and Successes

Q8 | Can you identify failure moments?

a. Did you at any moment think, "this is not going to work..."?

b. And how did you solve it?

Q9 | Can you remember successful moments?

a. Were those moments where you thought, "Yes! I've got it! This is it!"?

b. Can you explain why?

Support

Q10 | Was the provided support enough to complete the project?

Q11 | Do you think it could be better? What else would you need?

Suggestions

Q12 | In hindsight, can you identify a list of things that you would have done differently?

Q13 | What would be your advice for someone that need to start a project as you did?

FaltaP

Semi-Structured Interview

Simple or Difficult

Q14 | How do you evaluate the process in terms of difficulty?

simple | regular | average | complicated | super complicated

Q15 | How frequently did you encounter failures?

really frequent | frequent | average | not so frequent | rare

Q16 | How frequently did you encounter successful events?

really frequent | frequent | average | not so frequent | rare

Break This Glass

Semi-Structured Interview

About the project

Q1 | Can you describe what we have invited you to do from your perspective and in your own words?

Q2 | What were the motivations to participate in this research project?

Time, Learning and Phases

Q3 | If you can measure it – how long did the process feel?

Q4 | Can you describe the learning curve of this process?

Q5 | Can you identify different phases in the process?

Q6 | Were there any stressful periods during the process?

Q7 | Were there any surprising circumstances or facts that you encountered?

Failure and Successes

Q8 | Can you identify obstacles?

a. Did you at any moment think “This is not going to work…”?

b. Did you have enough support to solve it?

Q9 | Can you remember successful moments?

a. Were those moments where you thought, “Yes! I've got it!”?

b. Can you explain why?

Support

Q10 | Was the provided support enough to isolate a problem?

Q11 | Do you think it could be better?

Q12 | What else do you think would be required or could help?

Suggestions

Q13 | In hindsight, can you identify a list of things that you would have done differently?

Q14 | What would be your advice for someone who needs to identify a problem just as you did?

Appendix D

Preliminary Surveys for User Studies

This appendix contains the complete versions of the surveys used in our user studies presented in Chapter 6 and Chapter 7. We have included them precisely as given to the participants to provide a clear picture of the questions asked and how the data was gathered. This ensures that the context of our findings is understood and that our research methods remain transparent and replicable in future studies.

Welcome

From an Idea to a Prototype
Assisting non-experts to prototype IoT projects

Purpose
This research is conducted as a partial requirement for a PhD in Computer Science. This project requires the researcher to conduct research using surveys or interviews or combine the two techniques.

What is this research project about?
This research is to investigate assisting tools – such as custom-made playing cards and online tutorials – that could help non-experts to prototype an IoT project.

What will you have to do and how long will it take?
As part of this research, you will be invited to make a simple IoT project following a set of instructions (the researcher will provide all the components). You will be asked to complete a survey questionnaire online and interviewed by the researcher. The experiment should take approximately two hours. As compensation for your time, you will receive a $30 bookstore voucher. The researcher will record the interview. You will be asked to give consent before the interview and maybe ask to give consent later.

What will happen to the information collected?
The information collected will be used by the researcher to write a doctoral thesis. Articles and presentations may be the outcome of the research. Only the researcher and supervisors will be privy to the notes, documents, and recordings. The researcher will keep transcriptions of the recordings and a copy of the paper but will treat them with the strictest confidentiality. No participants will be named in the publications, and the researcher will make every effort to disguise their identity.

Declaration to participants
If you take part in the study, you have the right to:

- Refuse to answer any particular question, and to withdraw from the study before the 31st of March 2022, when analysis has commenced on the data. At the end of this survey, you will receive a unique number that you could use in case you want to withdraw from the study.

- Ask any further questions about the study that occurs to you during your participation.
- Be given access to a summary of findings from the study when it is concluded.

Who's responsible?
If you have any questions or concerns about the project, either now or in the future, please feel free to contact either:
Researcher: Tomás García Ferrari | tomasgf@waikato.ac.nz
Supervisor: Annika Hinze | hinze@waikato.ac.nz

Consent Form for Participants

I have read the Participant Information Sheet for this study and have had the details of the study explained to me. My questions about the study have been answered to my satisfaction, and I understand that I may ask further questions at any time.

I also understand that I am free to withdraw from the study before the 31st of August 2021, or to decline to answer any particular questions in the study. I understand I can withdraw any information I have provided up until the researcher has commenced analysis on my data. I agree to provide information to the researchers under the conditions of confidentiality set out on the Participant Information Sheet.

☐ I agree to participate in this study under the conditions set out in the previous page

Video and audio recording

○ I agree to my responses to be video and audio recorded
○ I prefer that the researcher will not video and audio record my responses

Use of images

- ○ I agree to my images being used
- ○ I prefer that the researcher will not use my images

Demographics

Age

What is your age?

- ○ Under 18
- ○ 18 - 24
- ○ 25 - 34
- ○ 35 - 44
- ○ 45 - 54
- ○ 55 - 64
- ○ 65 or older

Education

What is the highest degree or level of school you have completed?

- ○ High school degree or equivalent (e.g. GED)
- ○ Bachelor's degree (e.g. BA, BS)
- ○ Master's degree (e.g. MA, MS, MEd)
- ○ Doctorate degree (e.g. PhD)

Knowledge and Experience

Digital technologies

Digital technologies are electronic tools, systems, devices and resources that generate, store or process data in digital form (in uniform 0–1 form). Websites, mobile phones, digital televisions, or digital music players are examples of digital technologies.

The following questions will ask about your knowledge, experience and satisfaction with digital technologies. By asking these questions, we are not evaluating your knowledge but understanding your relationship with digital devices and services.

Computers in your work

Which computing platforms do you commonly use?

- ☐ Windows PC
- ☐ Macintosh
- ☐ Linux

Mobile phones and tablets

Which mobile operating systems do you use?

- ☐ Android – Google
- ☐ iOS – Apple
- ☐ Other

Knowledge

How would you qualify your digital technologies knowledge?

	Very poor	Below average	Average	Above average	Excellent
Setup and configure	○	○	○	○	○
Programming (using menus)	○	○	○	○	○
Coding (writing code)	○	○	○	○	○
Following instructions (e.g., manuals or tutorials)	○	○	○	○	○

Experience

How would you consider your level of satisfaction with digital technologies?

	Very dissatisfied	Dissatisfied	Neither	Satisfied	Very satisfied
Mobile phone	○	○	○	○	○
Tablet	○	○	○	○	○
Computer	○	○	○	○	○
Car digital tools (e.g., Bluetooth stereo, maps, etc.)	○	○	○	○	○
Home appliances	○	○	○	○	○
Other (e.g., ATM, online banking, etc.)	○	○	○	○	○

Internet of Things (IoT)

How would you rate your knowledge of IoT technologies?

	Very low	Below average	Average	Above average	Very high
Single-board microcontrollers (e.g., Arduino)	○	○	○	○	○
Cloud services (e.g., databases)	○	○	○	○	○
Sensors (e.g., temperature, proximity, etc.)	○	○	○	○	○
Actuators (e.g., servo motors, smart bulbs, etc.)	○	○	○	○	○

Community

How would you rate your awareness of the following community-based resources?

	Not at all aware	Slightly aware	Moderately aware	Very aware	Extremely aware
Arduino forum	○	○	○	○	○
Stack Overflow	○	○	○	○	○
Reddit	○	○	○	○	○
Other tech forums	○	○	○	○	○
Make: magazine	○	○	○	○	○
Instructables.com	○	○	○	○	○
Hacksters.io	○	○	○	○	○
Maker.io	○	○	○	○	○
GitHub	○	○	○	○	○

Help and assistance

How frequently do you ask for help online?
(beyond "asking Google")

	Never	Rarely	Sometimes	Often	Always
Post a question in a forum	○	○	○	○	○
Chat with experienced people	○	○	○	○	○
Ask on social media (e.g., Twitter, Facebook)	○	○	○	○	○

Your ID Number

How to withdraw

Your unique participant ID number is **${e://Field/Random%20ID}**.

If you would like to withdraw from the study, contact us before the 31st of August 2021, mentioning your number. The researcher will then remove that number and your participation from the study.

Powered by Qualtrics

Welcome

Break This Glass
Assisting non-experts to isolate problems in IoT projects

Purpose
This research is conducted as a partial requirement for a PhD in Computer Science. This project requires the researcher to conduct research using surveys or interviews or combine the two techniques.

What is this research project about?
This research project investigates assisting tools – such as custom-made playing cards and online tutorials – that could help non-experts isolate problems in IoT projects.

What will you have to do and how long will it take?
As part of this research, you will be invited to identify a problem occurring with a simple IoT project following a set of instructions (the researcher will provide all the components). You will be asked to complete a survey questionnaire online and interviewed by the researcher. The experiment should take approximately two hours. As compensation for your time, you will receive a $30 bookstore voucher. The researcher will record the interview. You will be asked to give consent before the interview and maybe ask to give consent later.

What will happen to the information collected?
The information collected will be used by the researcher to write a doctoral thesis. Articles and presentations may be the outcome of the research. Only the researcher and supervisors will be privy to the notes, documents, and recordings. The researcher will keep transcriptions of the recordings and a copy of the paper but treat them with the strictest confidentiality. No participants will be named in the publications, and the researcher will make every effort to disguise their identity. Images captured during the study may be used, but will not include faces and the researcher will make every effort to anonymise them by removing identifying details.

Declaration to participants
If you take part in the study, you have the right to:

- Refuse to answer any particular question, and to withdraw from the study before the 31st of March 2022, when analysis has commenced on the data. At the end of this survey, you will receive a unique number that you could use if you want to withdraw from the study.
- Ask any further questions about the study that occurs to you during your participation.
- Be given access to a summary of findings from the study when it is concluded.

Who's responsible?
If you have any questions or concerns about the project, either now or in the future, please feel free to contact either:
Researcher: Tomás García Ferrari | tomasgf@waikato.ac.nz
Supervisor: Annika Hinze | hinze@waikato.ac.nz

This research project has been approved by the Human Research Ethics Committee of the University of Waikato under HREC(HECS)2021#62. For any ethical questions or concerns please contact the Chair of the Committee, email hecs-ethics@waikato.ac.nz, postal address, University of Waikato, Te Whare Wananga o Waikato, Private Bag 3105, Hamilton 3240.

Consent Form for Participants

I have read the Participant Information Sheet for this study and have had the details of the study explained to me. My questions about the study have been answered to my satisfaction, and I understand that I may ask further questions at any time.

I also understand that I am free to withdraw from the study before the 31st of March 2022 or to decline to answer any particular questions in the study. I understand I can withdraw any information I have provided until the researcher has commenced analysing my data. I agree to provide information to the researchers under the conditions of confidentiality set out on the Participant Information Sheet.

☐ I agree to participate in this study under the conditions set out in the previous page

Video and audio recording

- ○ I agree to my responses to be video and audio recorded
- ○ I prefer that the researcher will not video and audio record my responses

Use of images

- ○ I agree to my images being used
- ○ I prefer that the researcher will not use my images

Demographics

Age

What is your age?

- ○ Under 18
- ○ 18 - 24
- ○ 25 - 34
- ○ 35 - 44
- ○ 45 - 54
- ○ 55 - 64
- ○ 65 or older

Education

What is the highest degree or level of school you have completed?

- ○ High school degree or equivalent (e.g. GED)
- ○ Bachelor's degree (e.g. BA, BS)
- ○ Master's degree (e.g. MA, MS, MEd)
- ○ Doctorate degree (e.g. PhD)

Knowledge and Experience

Digital technologies
Digital technologies are electronic tools, systems, devices and resources that generate, store or process data in digital form (in uniform 0–1 form). Websites, mobile phones, digital televisions, or digital music players are examples of digital technologies.

The following questions will ask about your knowledge, experience and satisfaction with digital technologies. By asking these questions, we are not evaluating your knowledge but understanding your relationship with digital devices and services.

Computers in your work
Which computing platforms do you commonly use?

- ☐ Windows PC
- ☐ Macintosh
- ☐ Linux

Mobile phones and tablets
Which mobile operating systems do you use?

- ☐ Android – Google
- ☐ iOS – Apple
- ☐ Other

Knowledge

How would you qualify your digital technologies knowledge?

	Very poor	Below average	Average	Above average	Excellent
Setup and configure	○	○	○	○	○
Programming (using menus)	○	○	○	○	○
Coding (writing code)	○	○	○	○	○
Following instructions (e.g., manuals or tutorials)	○	○	○	○	○

Experience

How would you consider your level of satisfaction with digital technologies?

	Very dissatisfied	Dissatisfied	Neither	Satisfied	Very satisfied
Mobile phone	○	○	○	○	○
Tablet	○	○	○	○	○
Computer	○	○	○	○	○
Car digital tools (e.g., Bluetooth stereo, maps, etc.)	○	○	○	○	○
Home appliances	○	○	○	○	○
Other (e.g., ATM, online banking, etc.)	○	○	○	○	○

Internet of Things (IoT)

How would you rate your knowledge of IoT technologies?

	Very low	Below average	Average	Above average	Very high
Single-board microcontrollers (e.g., Arduino)	○	○	○	○	○
Cloud services (e.g., databases)	○	○	○	○	○
Sensors (e.g., temperature, proximity, etc.)	○	○	○	○	○
Actuators (e.g., servo motors, smart bulbs, etc.)	○	○	○	○	○

Community

How would you rate your awareness of the following community-based resources?

	Not at all aware	Slightly aware	Moderately aware	Very aware	Extremely aware
Arduino forum	○	○	○	○	○
Stack Overflow	○	○	○	○	○
Reddit	○	○	○	○	○
Other tech forums	○	○	○	○	○
Make: magazine	○	○	○	○	○
Instructables.com	○	○	○	○	○
Hacksters.io	○	○	○	○	○
Maker.io	○	○	○	○	○
GitHub	○	○	○	○	○

Help and assistance

How frequently do you ask for help online?
(beyond "asking Google")

	Never	Rarely	Sometimes	Often	Always
Post a question in a forum	○	○	○	○	○
Chat with experienced people	○	○	○	○	○
Ask on social media (e.g., Twitter, Facebook)	○	○	○	○	○

Identifying problems

How obvious is it for you to spot the cause of an electrical, electronic or digital device malfunction?

	Not at all	Slightly	Moderately	Very	Extremely
Household appliances	○	○	○	○	○
Smart appliances	○	○	○	○	○
Audiovisual systems	○	○	○	○	○
Remotely controlled devices (e.g., garage door)	○	○	○	○	○
Computers	○	○	○	○	○
Wireless devices (e.g., laptop, tablet, mobile phone)	○	○	○	○	○
Other devices	○	○	○	○	○

Your ID Number

How to withdraw

Your unique participant ID number is **${e://Field/Random%20ID}**.

If you would like to withdraw from the study, contact us before the 31st of March

2022, mentioning your number. The researcher will then remove that number and your participation from the study.

Powered by Qualtrics

Appendix E

Posters

This appendix presents the posters used to recruit participants for the user studies described in Chapters 6 and 7. These posters were designed to provide potential participants with key information about the studies, including the purpose, participation requirements, and how to get involved. By sharing these materials, we aim to offer insight into our recruitment process and ensure transparency in participants' engagement in the research.

we need YOU

to help us better understand the IoT

We are researching Internet of Things (IoT) technologies with non-experts, and we invite you to participate in our user study.

No previous experience is required. But if you are curious about technologies that we see gaining presence in our lives, from smart homes to self-tracking tools, you will find the study interesting.

You will receive a book store voucher as compensation for your time.

Use the QR code to fill in your details, and we will get in contact with you.

From an Idea to a Prototype
Tomás García Ferrari | SCMS
tomas.garciaferrari@waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

do you WANT

...to use sensors and record your own data?

We are researching Internet of Things (IoT) technologies with non-experts, and we invite you to participate in our user study.

No previous experience is required. But if you are curious about technologies that we see gaining presence in our lives, from smart homes to self-tracking tools, you will find the study interesting.

You will receive a book store voucher as compensation for your time.

Use the QR code to fill in your details, and we will get in contact with you.

From an Idea to a Prototype
Tomás García Ferrari | SCMS
tomas.garciaferrari@waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

would you LIKE

...to use sensors to automate your home?

We are researching Internet of Things (IoT) technologies with non-experts, and we invite you to participate in our user study.

No previous experience is required. But if you are curious about technologies that we see gaining presence in our lives, from smart homes to self-tracking tools, you will find the study interesting.

You will receive a book store voucher as compensation for your time.

Use the QR code to fill in your details, and we will get in contact with you.

From an Idea to a Prototype
Tomás García Ferrari | SCMS
tomas.garciaferrari@waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

we need YOU

to help us identify where an IoT project fails

We are researching Internet of Things (IoT) technologies with non-experts, and we invite you to participate in our user study.

No previous experience is required. But if you are curious about technologies that we see gaining presence in our lives, from smart homes to self-tracking tools, you will find the study interesting.

You will receive a book store voucher as compensation for your time.

Use the QR code to fill in your details, and we will get in contact with you.

Break This Glass
Tomás García Ferrari | SCMS
tomas.garciaferrari@waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

do you WANT

...to isolate a problem in a sensor-based IoT project?

We are researching Internet of Things (IoT) technologies with non-experts, and we invite you to participate in our user study.

No previous experience is required. But if you are curious about technologies that we see gaining presence in our lives, from smart homes to self-tracking tools, you will find the study interesting.

You will receive a book store voucher as compensation for your time.

Use the QR code to fill in your details, and we will get in contact with you.

Break This Glass
Tomás García Ferrari | SCMS
tomas.garciaferrari@waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

would you LIKE

...to spot a failure in an home automation project?

We are researching Internet of Things (IoT) technologies with non-experts, and we invite you to participate in our user study.

No previous experience is required. But if you are curious about technologies that we see gaining presence in our lives, from smart homes to self-tracking tools, you will find the study interesting.

You will receive a book store voucher as compensation for your time.

Use the QR code to fill in your details, and we will get in contact with you.

Break This Glass
Tomás García Ferrari | SCMS
tomas.garciaferrari@waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato