future internet

MDPI

Article

Minimum Viable Products for Internet of Things Applications: Common Pitfalls and Practices

Anh Nguyen-Duc [1,*], Khan Khalid [2], Sohaib Shahid Bajwa [3] and Tor Lønnestad [1]

[1] Business School, University of South Eastern Norway, 3800 Bø i Telemark, Norway; Tor.Lonnestad@usn.no
[2] Karachi Institute of Economics and Technology, P.O. Box 75190 Karachi, Pakistan; khalid.khan@pafkiet.edu.pk
[3] University of Calgary, 2500 University Drive NW, Calgary, AB T2N 1N4, Canada; sohaibshahid.bajwa@ucalgary.ca
* Correspondence: angu@usn.no

Received: 28 January 2019; Accepted: 14 February 2019; Published: 18 February 2019

Abstract: Internet of Things applications are not only the new opportunity for digital businesses but also a major driving force for the modification and creation of software systems in all industries and businesses. Compared to other types of software-intensive products, the development of Internet of Things applications lacks a systematic approach and guidelines. This paper aims at understanding the common practices and challenges among start-up companies who are developing Internet of Things products. A qualitative research is conducted with data from twelve semi-structured interviews. A thematic analysis reveals common types of Minimum Viable Products, prototyping techniques and production concerns among early stage hardware start-ups. We found that hardware start-ups go through an incremental prototyping process toward production. The progress associates with the transition from speed-focus to quality-focus. Hardware start-ups heavily rely on third-party vendors in term of development speed and final product quality. We identified 24 challenges related to management, requirement, design, implementation and testing. Internet of Things entrepreneurs should be aware of relevant pitfalls and managing both internal and external risks.

Keywords: Internet of Things applications; prototyping; hardware start-ups; Minimum Viable Products; case study

1. Introduction

In recent years, with the emergence of Internet of Things, societies have become extremely interconnected. Advances in hardware development and the availability of powerful but very inexpensive integrated chips enable application of Internet of Things to not only industry, manufacturing but also every aspect of our daily life. With the Industry 4.0 revolution [1], the adoption and development of connected systems, such as smart grids, autonomous automobile systems, medical monitoring are becoming mainstream. Unlike traditional embedded systems, which are designed as stand-alone devices, the focus of Internet of Things is on networking several devices [2]. The number of connected devices available in the worldwide market is approaching 15 billion [3]. According to Gartner's hype cycle, by 2020, such technologies will be in 95% of electronics for new product designs.

Internet of Things is typically a complex system in several dimensions as they embody characteristics of physical, networked, software and of human-interactive systems [4]. Physical and software components are closely intertwined in Internet of Things, each specializing in different architectural and operational scales. As an emerging area with the overlapping and integration of multiple fields of science and engineering, the development of Internet of Things requires software and system engineers, computer scientists and network professionals to collaborate closely with experts in

various fields such as automation and control, civil engineering, mechanical engineering and biology. Consequently, Internet of Things development is closely related to software engineering paradigm, processes and practices. It has been much focus on the applications, use cases of Internet of Things, less investigated in how Internet of Things are developed.

Internet of Things is also a huge business opportunity as an increasing number of start-ups are in the quest for successful Internet of Things business models. Start-up companies are participating in this technological revolution actively, even as major driving forces in some industry sectors. Start-ups are new companies with limited resources, short operational history, often looking for scalable business models [5,6]. Start-ups appear as a special context in which traditional product development approaches might not be directly applicable. The global movement of start-ups calls for the attention of practitioners and researchers in the quest for development methodologies that are suitable for start-ups' business objectives, as well as their unique engineering environments [7]. Start-ups adopt certain approaches to develop their products and to some extent, have a direct impact on business objective and activities. For instance, empirical studies on software start-ups reveal common phenomena in the development of software products, such as agile development, evolutionary prototypes, customer involvement, technical debt and the neglect of quality [8–12]. Recent advances in hardware prototyping (i.e., 3D printing and hardware development kits), the development of hardware-related products can be more agile and iterative [8,13]. For instance, a higher development pace and greater flexibility are reported to be facilitated by using Agile methodologies in hardware projects at Ericsson [3].

To the best of our knowledge, the body of research examining Internet of Things development in the context of start-ups is very limited. To address this research gap, we aimed to explore the state-of-practice of hardware-related product development. Previous knowledge about Minimum Viable Products (MVPs) in software start-ups helped us to compare and contrast key engineering activities in early-stage start-ups [11]. Consequently, we derived two research questions (RQs) from the research objective:

1. RQ1: How do start-ups prototype their Internet of Things applications?
2. RQ2: What are the challenges associated with Internet of Things development in start-up contexts?

The paper is organized as below; the next Section is about Related work. After that, we present the research methodology. Then findings for RQs are presented and discussed. Finally, the conclusion ends the paper.

2. Related Works

2.1. Software Engineering Aspects of Internet of Things

The emergence of the Internet of Things is bringing us connected devices that are an integral part of the physical world. Internet of Things systems, for instance, smart home solutions, need to consider a comprehensive system, from cloud-based storage and data analysis, end-user applications, middleware and hardware devices and their connectivity, hence representing a mixed hardware-software system [14]. The generic architecture of an Internet of Things system proposed by Taivalsaari and Mikkonen [15] largely defines the elements of any Internet of Things: client devices; gateways, backend cloud, including databases and other storage systems; analysis applications and end-user applications. Zambonelli et al. generalize common features of CPS systems that are relevant to software engineering into stakeholders and users, requirements, groups and coalitions, avatars and Smart things [16–18]. Fortino et al. propose an agent-based approach in modelling Internet of Things systems [17]. As seen in Figure 1, Internet of Things development is conceptualized in a water-fall approach, with analysis, design and development. We adopt this general view on common elements in an Internet of Things system, which gives us flexibility when looking at various systems.

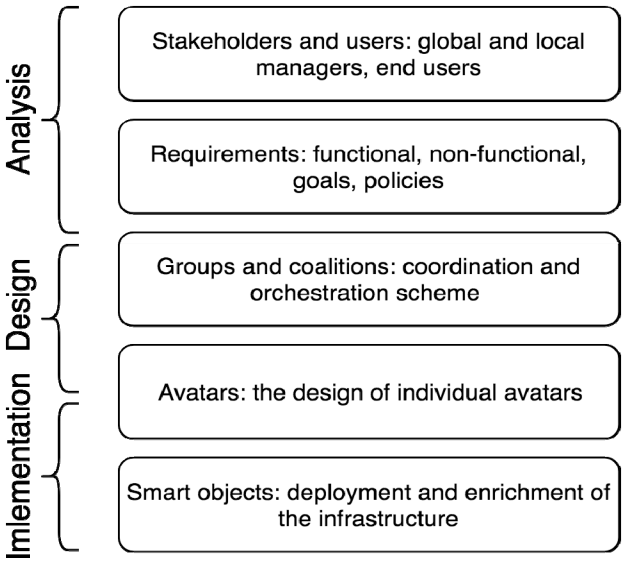

Figure 1. Conceptual elements of an Internet of Things system (adapted from [16]).

2.2. Internet of Things Start-Ups

The term "start-up" has been around with increasing frequency among practitioners and researchers in the last five years. Steve Blank describes a start-up as a temporary organization that aims to create high-tech innovative products without having a prior working history [19]. He further highlights that in a start-up context, the business and its product should be developed in parallel. Eric Ries defines a start-up as a human institution that is designed to create a unique product or service under extreme uncertainty [8]. Rather than a formal company, a start-up should be considered as a temporary organizational state, that seeks a validated and scalable business model [20]. Hence, a company with a dozen employees can still be in a start-up state to validate a business model or a market. Start-ups are designed to grow fast while small and medium-sized enterprises (SMEs) focus on doing stable business for years [21]. Among several ways of describing start-up life cycles, Ries' characterizes start-ups in three steps, which are a problem–solution fit, product–market fit (or the early stages) and scale-up (or the later stages) [8]. We focused on early-stage start-ups, that is typically a small entrepreneurial team developing software-intensive products with scalable business models. A series of prototypes and beta product can be observed in this early phase.

The phrase "hardware start-ups" has been used in several publications to refer high-tech start-ups that focus on delivering products and/or services to individuals, households and industry through physical electronic devices [22,23]. Hardware-related products, for instance, smart home solutions, require the producer to consider a comprehensive system, from a cloud-based storage and data analysis to end-user applications, middleware and hardware devices and their connectivity [24] and thus represent a mix of hardware and software systems. We refer to Internet of Things start-ups as companies that develop and deliver customer value based on Internet of Things applications. Internet of Things products are normally regulated by domain-specific standards, regulations and fixed-requirements. Quality attributes, such as performance, robustness, safety and security, are more critical to the success of an Internet of Things product.

2.3. Start-Up Product Development Methodologies

The need for the product development methodologies to be adapted to a projects scope, magnitude, complexity and changing requirements in start-up context is generally acknowledged, however, there exists a lack in guidance on how software start-ups can adapt their process to their situational context [25]. Start-ups perform experiments in the problem-solution domain, which often ended up with launching a product that greatly differs from the original idea [11,22].

This learning experience is always associated with the construction of Minimum Viable Product (MVP), a representative proxy of the final product, to validate either new technology or to elicit customer requirements [8,11]. MVP is a core of Lean start-up concept [8], which might or might not be developed directly into the final product (as shown in Figure 2). Eric Ries initiates the classification of MVP types [8], which is in common use in the start-up communities, as shown in Figure 2. For instance, an MVP can be a short animation that explains what your product does and why users should buy it. It also can be a user interface that looks like a real working product but the actual business process is manually carried on (Wizard of Oz). A concierge MVP is a manual service that consists of exactly the same steps users would go through with the product.

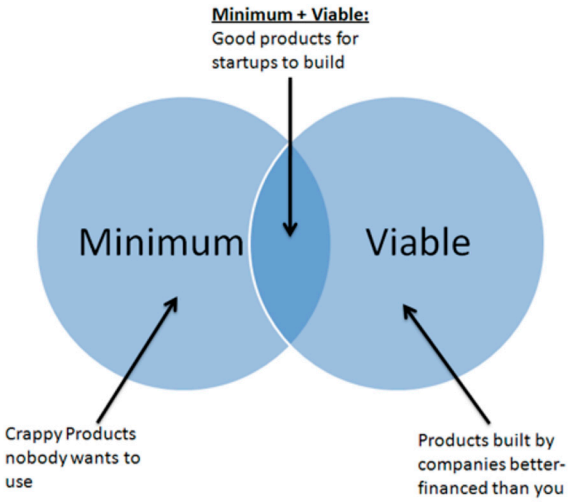

Figure 2. The definition of Minimum Viable Product (MVP).

Sone research has been done on how start-ups developed software products that reflect various types of software MVPs and MVP prototyping practices [5,6,11,20,25–28]. Nguyen-Duc et al. studied the roles of MVPs in software start-ups and different approaches to speed up the process of making MVPs [6,11]. Developers in software start-ups typically prioritize speed related to agile practices rather than quality related ones [25,27]. Paternoster et al. summarized a pattern of software start-ups that quickly built a product in an evolutionary fashion and learned from the users' feedback [5]. Bosch et al. advocated for adjusting the Lean start-up methodology to accommodate the development of multiple ideas and integrate them when the time for their testing and validation is too long [26]. Giardino et al. presented a greenfield model to explain the priority of start-ups to develop product versions as soon as possible [28]. The ability to verify product-market fit in the early phase is traded off for technical debt in product side in later phases.

2.4. State-of-Practice Hardware Product Development

Hardware-related product development has several characteristics where it differs from pure software development. Compared to software development, hardware product development takes longer [22]. For instance, prototyping and demonstrating a software prototype might occur over time on the order of weeks or days. For hardware products such as consumer electronics, development of a demonstrable prototype might take many months and years after the associated technologies have been proven [22]. Advanced technologies and tools for modelling, simulation has enabled the adoption of shorter cycles of hardware-related product development [29,30]. Research has shown several positive experience using Agile methodologies in embedded systems [31–35]. Greene reported a positive experience of applying Agile approaches in firmware development at Intel [31]. Cordeiro et al. proposed a new platform-based design using Agile approaches that substantially reduced development time in three hardware development projects [32]. Adopting XP practices,

Santos et al. showed a successful software version created for control of a satellite camera [33]. Kaisti et al. conducted a systematic mapping study of the adoption of Agile methodology in embedded systems development [34]. Ronkainen et al. described challenges with hard real-time requirements, prototyping, documentation and test-driven development in Agile hardware development [35]. The authors suggested that Agile practices can be used in the embedded domain but the practices need to be adapted to fit the more constrained field of embedded product development [35].

3. Research Methodology

This section presents our research methodology (Section 3.1), approach to data collection and analysis (Section 3.2) and a description of the cases investigated in this study (Section 3.3).

3.1. Study Design

Following Yin's case study guideline [36], we conducted a multiple-case exploratory study. Exploratory case studies are selected for the purpose of "finding out what is happening, seeking new insights and generating ideas and hypotheses for new research" [37]. A multiple-case study enables researchers to explore differences within and between the cases. A large number of cases can help to triangulate findings, which were particularly important since we obtained only one or at most, three interviews per case. We selected a start-up as the unit of analysis and adopted a purposive sampling strategy to recruit cases [38]. As start-ups often focus on one product or one service from the beginning, in all of our cases, each start-up focused on one product development project. There is often difficulty in identifying a real start-up case among other similar phenomena, such as pure software start-ups, SMEs or part-time start-ups. Therefore, we clearly defined the criteria for our case selection:

(1) a start-up that has at least two full-time members, so product development is not an individual activity;
(2) a start-up that operates for at least six months, so their experience can be relevant;
(3) a start-up that has at least a first running prototype, so its engineering practices are a relevant topic; and
(4) a start-up whose product or prototype is composed of Internet of Things components

From previous work [6,11,20,27], we have constructed a contact list of 219 start-ups. Other co-authors of this work added 22 hardware start-ups to that list. From the total number of 241 start-ups, we identified 52 that potentially satisfied our case selection criteria. After contacting the companies via email, eight start-ups were willing to participate in this research. The first three cases were collected from start-ups at which the first author had a personal relationship, which made it easier to collect in-depth material to increase our understanding of the cases. We did not select cases due to their adoption of prototyping or agile practices. Hence, these cases represent an unbiased sample of real situations occurring in Internet of Things start-ups.

3.2. Data Collection and Analysis

Figure 3 summarizes our approach to data collection and analysis. The major data source was semi-structured interviews. Semi-structured interviews are a common approach to collecting relevant insights into many phenomena in software engineering [37,39]. The advantage of semi-structured interviews is that it allows improvisation and exploration of the objects studied [37]. Secondary data sources were published information about start-ups collected from the Internet. Prior to conducting interviews, we obtained as much information as possible regarding the start-ups' products, markets, customers and business models. As shown in Table 1, we conducted a total of twelve interviews with interviewees from eight companies. All interviewees are CEOs, CTOs or hardware engineers that have been with the start-ups from the beginning. The years of technical experience of the interviewees ranged from two to more than 10 years. The interview guide consisted of five sections: (1) business

background, (2) idea visualization and prototyping, (3) product development, (4) challenges and lessons learned and (5) adoption of development paradigms. The interview guide is presented in Appendix A. During interviews, we adjusted the order of questions based on the responses and comments of the interviewees.

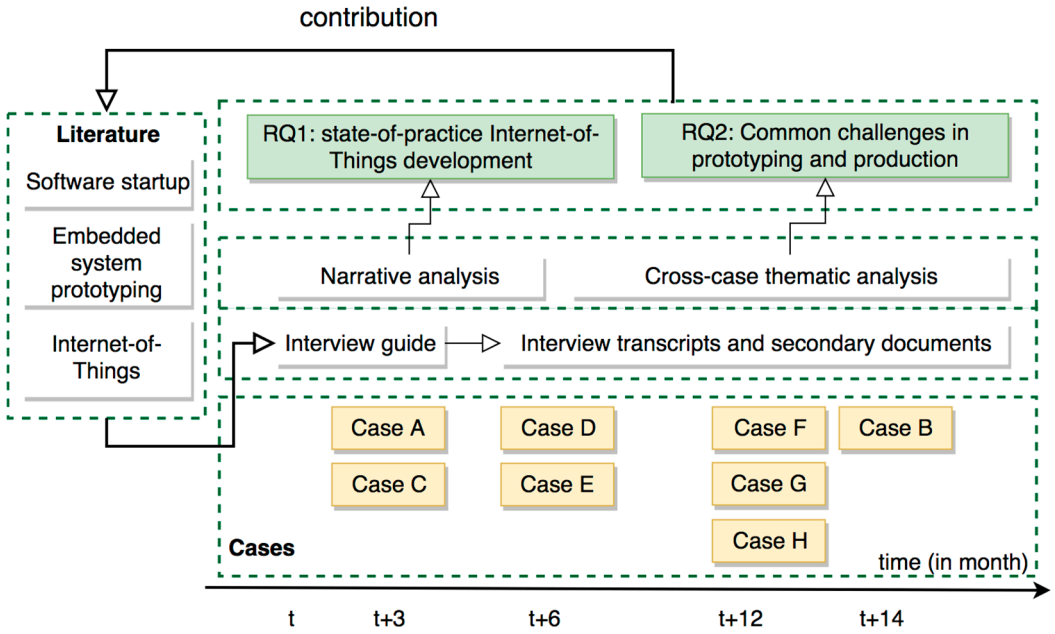

Figure 3. Research approach.

Table 1. Summary of qualitative data.

Id	Profile of Interviewee(s)	# of Interviews	Avg. Years of Exp.	Other Documents
Case A	CEO, CTO, hardware developer	3	2	Yes
Case B	CTO	1	2	Yes
Case C	CTO	2	10+	Yes
Case D	CEO	1	7+	No
Case E	CEO	1	5	No
Case F	CEO, CTO	2	3	Yes
Case G	Hardware developer	1	3	No
Case H	CEO	1	4	No
Total		12		4

Data were first collected from Case A and Case C to provide an initial basis for the development of the research questions. In these cases, the first author conducted follow-up interviews with both the CEO and the CTO. Due to the personal relationship of the author with the companies, we had access to other documents, including business model canvases, business plans and development contracts.

Six months later, we collected data from Case D and Case E. This was done collectively by the first author. In these cases, we conducted follow-up interviews, asking participants to look at a list of Agile practices and discuss challenges and opportunities these practices present in connection to hardware development. After one year, we collected three more cases, Case F, Case G and Case H, with interviews conducted by the second author. Data for the final case, Case B, was collected by interviews conducted by two master's students from the Norwegian University of Science and Technology (NTNU). All interviews were conducted face-to-face and with the participation of co-authors of this study.

We adopted two approaches for analysis of the data: narrative and thematic analysis. To address RQ1, a narrative summary of interview transcripts and online documents relating to the cases was created to tell a story about each case [40]. The first and second authors interpreted and described stories of how start-ups prototype. To address RQ2, we adopted a thematic analysis, which is a common data analysis approach for qualitative data [41]. Thematic analysis is defined as "a method for identifying, analysing and reporting patterns (themes) within data" [41]. We applied a process of thematic synthesis, which has been applied and published in software engineering research venues [42]:

1. Initial reading: We read through the transcribed interviews to generate initial ideas and identify possible patterns in the data. All interviews were transcribed shortly after they were conducted to ensure the actual meaning of interviewees' answers was preserved.
2. Coding: We scanned textual data to search for challenges facing start-ups. The final interpretation of case descriptions and results were reviewed and adjusted by other co-authors.
3. Translating codes into themes: A theme can be seen as a way of grouping initial codes into a smaller number of sets to create a meaningful whole of unstructured codes [8]. Since we were interested in understanding prototyping practices and challenges in Internet of Things start-ups, we mainly focused on creating themes describing concerns of prototyping (RQ1). We identified a total of 23 unique themes from our codes.
4. Creating higher-order themes: The generated themes were further explored and interpreted to create a model of higher-order themes. We used terminology from SWEBOK knowledge area [43] to guide the coding of second-order themes when organizing prototyping challenges in Internet of Things start-ups. We found a total of four higher-order themes.

3.3. Case Description

Our case description, from Case A to Case H, is summarized in Table 2. Our sample consists of start-ups in Norway, Finland and Pakistan. Seven start-ups were characterized as being in early stages and one start-up in a later stage, as described below.

Table 2. Case demographics.

Id	Product	Year	# ppl.	Country	Current Stage
Case A	Fish farm tracking system	2016	6	Norway	Early stage
Case B	Networks of connected camera	2016	10	Norway	Early stage
Case C	Under water drone	2013	4	Finland	Early stage
Case D	Tracking devices for shipment	2013	85	Finland	Later stage
Case E	Muscle operation measure	2011	20	Finland	Early stage
Case F	Smart home solution	2015	8	Pakistan	Early stage
Case G	Smart wheelchair	2016	3	Pakistan	Early stage
Case H	Smart home devices	2016	5	Pakistan	Early stage

- Case A is a start-up in an aquaculture domain founded in late 2016. At the time of the investigation, Case A was finalizing their first prototype—an environmental tracking and monitoring solution for fish farming. The targeted users were small-to-medium farms in South-East Asian countries. The company had received seed funding under the South Korean government's incubator program. Team A includes a CEO (business developer), four software developers and one hardware engineer.
- Case B is an Internet of Things start-up founded in early 2016. It offered an enhanced device that is integrated into a network of cameras. Started by three students from NTNU, Norway, the company now has ten employees, five of whom work on product development. The start-up acquired seed funding from several Norwegian funding schemes.

- Case C is a hardware company founded in 2013. The product is an underwater drone that can be used for fishing and aquaculture research. The product consists of a camera and a web-based controller. The business model is B2C; at the time of this study, its focus was on marketing to individual fishers and researchers. The company had four members of staff; the CTO is also a mechanical engineer. The CEO and the head of marketing took care of purchasing of components, subsystems, PCs and accessories as needed.
- Case D is a start-up founded in 2013. The company started as a mobile app development with 20 employees. This team is a spin-off from an international enterprise. At the time of this study, they had 85 members of staff who mainly worked to develop Internet of Things solutions. The company started with a first product that provided tracking devices for expensive shipments. Employing a B2B model, Case D had revenue of c.a. 8 million Euro in 2014.
- Case E is a grown start-up founded in 2011. Developed from work that formed the basis of a master's thesis, the company provides equipment for measuring muscle operation under normal training conditions. The CEO is a muscle specialist who possesses domain knowledge. The company has sold their products to several hospitals and gyms.
- Case F is a start-up developing an intelligent home automation system that allows users to control electric appliances remotely, using an Android app over either GSM or Wi-Fi. Started in 2015 by a team of five college students, after one year they had eight full-time members of staff. The company was in a pre-start-up phase at the time of this study with regard to their financial situation and business development.
- Case G is a hardware start-up that offers an eye-tracking-based control system for smart wheelchairs. The product adopts Electroencephalogram signal processing and Internet of Things technologies. The wheelchair can be controlled and moved by looking at a screen. The business model is B2C, with a focus on the general public, disabled students, elderly people and so forth. This start-up arose from a university project. During its first two months in an incubation program, they sold three product units. The start-up won a national student-based entrepreneurial competition.
- Case H is an Internet of Things start-up which was founded in 2015. Positioned in the smart home sector, Case H has two major products: Smartic and Smart Board. Smartic is a plug-and-play device that can be connected to any electrical device to automate it. Smart Board, on the other hand, replaces conventional switchboards in smart houses and automates all electrical devices. The business model is B2C, serving home and office users. At the time of this study, there were five employees in the company, including a CEO (business developer), three software developers and one hardware engineer.

4. Results

This section presents our findings for RQ1: How do Internet of Things start-ups build their MVPs in the early stages? (Section 4.1), RQ2: What are the challenges associated with MVP engineering in Internet of Things start-ups? (Section 4.2).

4.1. RQ1: How Do Internet of Things Start-Ups Build Their MVPs in the Early Stage?

4.1.1. Key MVPs from Internet of Things Start-Ups

The interviewee was asked about the MVPs that are critical for their start-ups in the early stage. This does not mean each start-up produces only these MVPs. Many MVPs can be created but start-ups only focused on describing one MVP that is most important for them at their early stages. The role of the MVPs can be to acquire funding, to attract customers or to recruit team members and so forth. Table 3 presents the descriptions of these major MVPs in our eight hardware-start-ups. For each MVP, we also specify the software/hardware components were developed, the time used to complete the MVPs and grouping them into three groups.

Table 3. Characteristics of major validated MVPs in Internet of Things start-ups.

Id	Key MVPs	Hardware Approach	Software Approach	Prototyping Duration	Type
Case A	The MVP include prototype, which includes aqua environmental tracking sensors (e.g., pH and temperature), a mobile app for an aquaculture farming control system	"Lego composing"	incremental in-house development	9 months	Type 1
Case B	The MVP assembled with circuits and covers that can be tested by users	Fully in-house design, 3rd party components, ad hoc and discrete	Simplified software development	2 months	Type 2
Case C	The MVP includes the fully integrated, operational camera	In-house design	Incremental development, Open-source components	28 months	Type 2
Case D	The MVP includes sensors and IoT gateways, cloud storage and user interface that is fully functional	In-house design, sub-contracting	Incremental software development 2 week Sprint	24 months	Type 3
Case E	The MVP includes tested sensors and integrated circuits and a mobile app interface	Outsourced sensor development,	In-house development	18 months	Type 1
Case F	The MVP includes a single feature to switch energy saver with a simple data storage in the cloud	3rd party components, in-house assembling	In-house development	3 months	Type 2
Case G	The MVP includes a simple eye tracker and motor controllers	Mock-up hardware	In-house development	5 months	Type 1
Case H	The MVP consists of a single hardware unit	3rd party components, in-house assembling	In-house development	3 months	Type 1

MVP creation duration varies from three months to 28 months. The process started since ideas were agreed among the team until the MVP is ready for its purpose. Several activities are included in the prototyping process, from research and development (R&D) activities, market and customer research, preparation of tools and necessary components, physical design, implementation and testing. The varied durations depend on the MVPs themselves and contextual factors:

- In Case H, the MVP was done in a short time (3 months). The major MVP was quite simple with no physical unit design. The significant amount of time was spent on conceptual design and sketching features. The team ordered available electronics units and integrated into the software part. The team focused on a front-end design of the software product. The integration was based on a 3rd party service provider.
- In case C, the MVP was complete in a very long time (28 months). The prototyping process is lengthy due to the complexity of the hardware design and the part-time nature of the participation of the founders. The core part of the MVP is the design of the camera. Besides many architectural elements were bought, other elements were designed in-house, which takes time. The MVP heavily depended on open-source components for both hardware and software parts.

Based on the similarities in objectives and methodologies of the development of MVPs, we classify the Internet of Things start-ups' MVPs into three types: piecemeal MVPs (MVP type 1), functional MVPs (MVP type 2) and non-functional MVPs (MVP type 3), as shown in Figure 4. An MVP can be used as the basis to develop the next MVP or incremental prototyping.

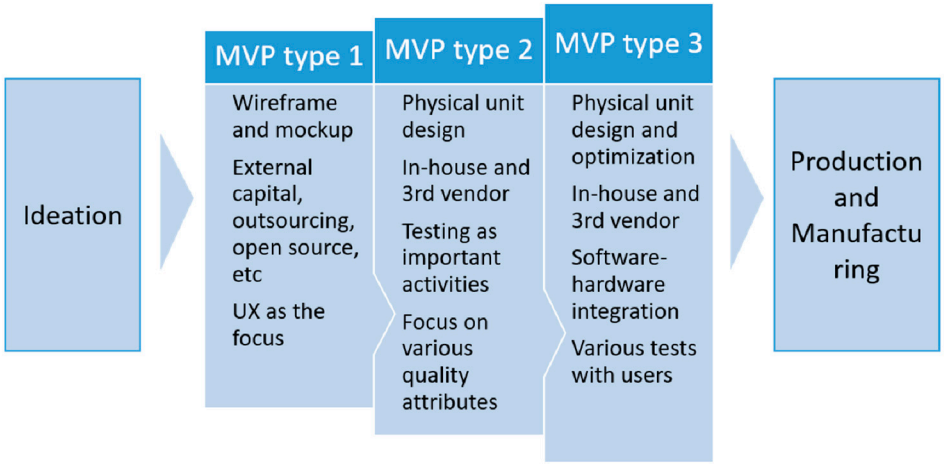

Figure 4. Iterative constructions of MVPs in Internet of Things start-ups.

MVP type 1 has user experience (UX) similar to the final product but the actual functionalities are manually carried on. This type is found in Case E, G and H. Type 1 could be seen as a simulation of desired products. As internal development effort is usually limited due to the lack of available competencies or funding, MVPs type 1 typically utilize rapid prototyping, local contractors and ready-made components. For instance, CTO in Case E mentions:

> *"we outsourced the sensor development so, we did not have that kind of skills and, I do not think we would have had the equipment either, to go for that"* (Case E)

MVP type 1 is the first step after the ideation phase and due to the coarse-grained illustration, it would be easy to iterate. For such MVPs, UX is an important part, for instance:

> *"And according to the feedback what they have been providing, we have been developing the solution during the whole springtime. We have been developing, quite a lot of functionalities, based on the user feedback. And we have also made quite a lot of usability-related changes and modifications in the software."* (Case E)

Evolving from MVP type 1, Internet of Things start-ups progress to create MVP type 2. The transition can be illustrated as in Case F, when the team moving from an MVP with an only single feature to a functional MVP:

> *"It was nothing more than switching on/off of an energy saver using SMS from our mobile phone. But the current product has a ton of features including controlling, grouping, timers, electricity consumption monitors, motion sensors, cameras."* (Case F)

MVP type 2 involves hardware unit design, including sensor integration, chips, circuit design and system engineering. The product would be complete from functional aspects. Quality attributes of the final products are considered. Most of the hardware designs are done in-house but the activity might also involve outsourcing or contractors. This type is found in Case A, B, D and F, as illustrated in the quote below:

> *"The development is close to the hardware, so you need to know the entire structure of the hardware. When you are on engine control, you need to understand how the electrical circuit is built to write the code structure"* (Case B)

Functional testing is an important activity for this type of MVP. Testing can be done at the unit level, component level, system level and user level.

> *"Q: you make finished physical products every time you prototype? A: Yes. Then they are ready so that a non-technical user can test them. We run some tests ourselves, just watching if it works or not... it is important that it functions all together so that actual users can provide feedback."* (Case B)

"It helped us a lot in proper testing in a live environment and getting an idea of daily usage by a customer. (Case F)

MVP type 2 would be functionally complete and operable with actual data:

"what could be done in-house testing and then there was some field testing ourselves before the shipment. Then of course, the most interesting data came from the real shipment" (Case D)

The transitions from MVP type 2 to type 3 involves the shift of effort from hardware development to software development. When a hardware unit as a core value proposition is complete and validated, the added value components, such as, data visualization and processing is more focused. Quality attributes are the focus in MVP type 3. MVP type 3 involves the design and optimization of current MVPs to achieve the best acceptable quality products. The acceptable quality MVPs is reached at a so-called Minimum Acceptable Quality to enable the rapid prototyping. Non-functional testing is the focus for MVP type 3. The integration of software and hardware elements was done and tested regressively, as shown in Case 3:

"They [software components] are at the same time so tightly depending on our hardware that it is not possible to understand the stuff without a deep understanding of the whole." (Case C)

At this step, the focus is on the integration of hardware and software elements to provide the best values:

"Different people work on different parts, so they are easy to distinguish. The electronic boys alternate between the code and developing the product itself." (Case C)

4.1.2. Engineering Approaches at Early-Stages

As shown in Figure 5, the development of three types of MVPs reveals a number of common engineering approaches, namely (iterative development, outsourcing, third-party components and rapid prototyping).

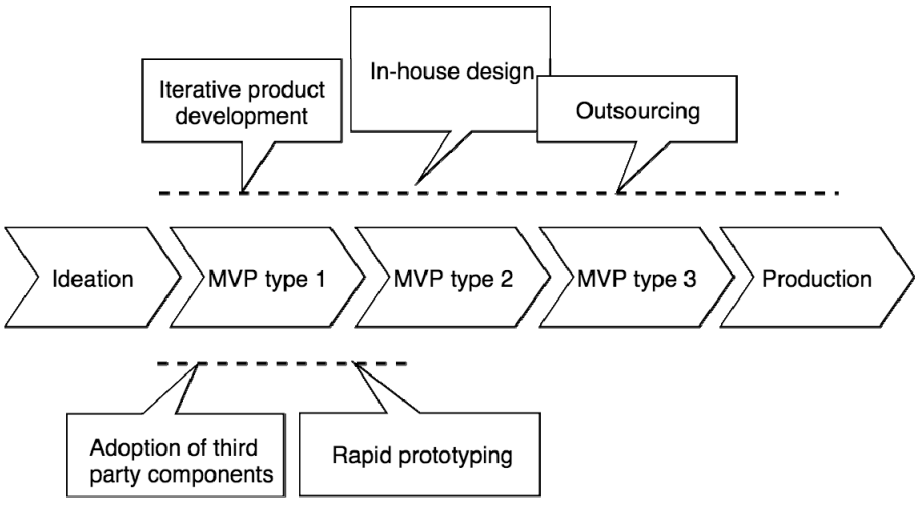

Figure 5. Engineering approaches in early-stage Internet of Things start-ups.

Iterative development is the regular release of product versions. The duration of each iteration varies among Internet of Things start-ups. Hardware development requires a longer duration than software development does and a one- to four-week sprint structure is probably insufficient to make reasonable progress in a hardware development project. It is also difficult for many hardware projects to release working products frequently, though it is a high priority according to standard Agile principles. Hardware development involves shipping physical units, which is costly and

time-consuming. Delivering an MVP type 2 or type 3 to a customer after each iteration and then adapting the design to the customer's changing specifications could dramatically increase the cost of a project. In our cases, throughout multiple iterations, MVP type 2 and type 3 does not change much from the initial design.

Iterative development was mentioned as a mean to achieve agility for Internet of Things start-ups. The CEOs related Agility with less upfront planning, short-term driven evolution of the start-ups. They also mentioned the speed of prototyping, development and fast time-to-market when asked about Agile. Start-ups stated that full controls of development activities and partnership would prepare themselves to respond to unexpected changes. Some start-ups also highlighted the importance of team collaboration over defined processes. The adoption of certain Agile practices or approaches might be different between the development of hardware and software elements:

> *"Our electronics are relatively simple, while SW changes very much all the time. We are still trying to find what is the right way to do it"* (Case C)

Outsourcing refers to the use of local contractors or subcontractors in start-ups' early stages. An example of local contractors is consultant companies, maker space, student projects and manufacturers. In case of D, skilled contractors were hired to achieve a quick start with a functional, (and normally physical) prototype. As mentioned by in Case E, utilizing external resources enables the capacities of speeding up product experimentation and development. Furthermore, as contractors are not an integral part of the start-up, they facilitate the scaling-down activities when they lack funding or change directions. Some start-ups use local contractors while some others hire offshore vendors. Making use of local vendors can be a feasible option, contributing to prototyping speed:

> *"The local one delivered very quickly. It is critical that the component from China comes on time, especially since we next week will display the latest iteration in England. Every time it's just a matter of time. If we could do everything internally, we would have saved a lot of time on sending, it would have been great"* (Case E)

Adoption of third-party components, whether they are open source or commercial off the shelf. Hardware development can take advantage of ready-made components to speed up prototyping and product development. Internet of Things start-ups utilize both off-the-shelf hardware components and open source components and libraries. Many of our start-ups build their first prototype with Arduino and Raspberry Pi, open-source electronic prototyping platforms (Case A, B, C, F, G, H). All start-ups purchased sensor devices like accelerometer, gyroscope, light and touch sensors from third-party providers. The adoption of existing components also enables the practice of reusing in incremental prototypes.

Rapid prototyping: Internet of Things start-ups are beneficial from the advancement of prototyping technologies, such as 3D printing and CAD simulation tools. Almost all of our cases own or acquire 3D printing services, which enable them to make many physical prototypes in a short time.

> *"We solicit components, test different things and form factors, using a lot of 3D printing. We have invested in a better 3D printer so we can use the sensors on patients in the hospital. The cheaper printers make rough surfaces that will not be approved for medical use. This is a thing we otherwise would have to order from someone else but that would take 3 weeks, so we removed it and invested in the better printer for our mechanical development."* (Case C)

It is also noted that even though 3D printing can be used for every physical component of a product, the printing takes time. In one case, it took 10 h for a 3D printing of a small component. A lot of communication and changes happen already in the computer-based prototypes, that is with CAD designs.

4.2. RQ2: What Are the Challenges Associated with MVP Engineering in Internet of Things Start-Ups?

The challenges relating to hardware prototypes were merged into five themes: (1) Management, (2) Requirement, (3) Design, (4) Implementation and (5) Testing. Table 4 presents the thematic map, with the 24 MVP challenges, designated C01 through C24. The challenges are also mapped to temporal dimensions of start-ups, whether they are relevant to early stages (i.e., MVP type 1, type 2) and later stages (MVP type 3).

Management concerns the aspects of finance, market and human resources during prototype development. MVPs are typically integrated from various components. Component use and reuse are common in software development but are even more prominent in hardware production, as it is more difficult to produce physical elements in-house. Consequently, the selection and evaluation of third-party vendors who provide satisfactory components are essential (C01). Multiple aspects need to be considered, that is the technical quality of the components, their cost and supporting policies of the vendors. The commitment of vendors to the speed (C08) and quality of delivery, for instance, needs to be proven over time. Given that a large number of hardware manufacturers are based in China, finding a suitable vendor is not straightforward for start-ups in other countries.

Table 4. Classifying challenges with Internet of Things MVPs.

Management	C01 - Evaluation and selection of third party vendors C03 - Acquiring suitable hardware engineers C06 - Insufficient funding for full-scale prototyping C07 - Pressure of time to market C08 - Delay due to late shipping from third-party vendors	C04 - Marketing the product with limited budget C09 - Underestimating manufacturing cost
Teamwork	C05 - Team communication	
Requirement	C02 - Finding actual needs and demands C24 - Fuzzy and evolving domain-specific constraints	
Design	C10 - Choosing right hardware platforms C13 - Openness and access to external support of the design C14 - Compliance with quality standards C15 - Dependence on third-party quality	C11 - Manufacturing cost effective design C17 - Finding the right balance between profitable and reliable hardware modules
Implementation	C12 - Dealing with complex hardware design C18 - Complexity of integrating software and hardware C19 - Unavailability of hardware components C20 - Finding the right open-source software components	
Testing	C21 - Testing non-functional attributes C22 - Real-world environment for beta testing	C23 - Quality assurance of evolving hardware, firmware and software parts
	Early stages	Later stages

Constraints on human resource are relevant in various stages of Internet of Things start-ups. Finding an appropriate hardware engineer is crucial for the long-term development of the company (C03). The competence of the hardware team not only determines the capabilities of the company but also influences the speed and quality of the products it produces. Another important dimension is finance, as we observe that our cases initiated their businesses by self-funding. Hardware development requires more resources, in terms of both human and financial investment, than software start-ups do. Many start-ups lack sufficient funding for the prototyping phases (C06). This lack can become even more of an issue when close to manufacturing. Start-ups operate under the pressure of speed, which is also a threat to quality-driven development (C07):

> *"There are new launches from big companies also like Samsung. Then there are some, many kinds of players are launching now (their own stories) . . . so there will be competition"* (Case D)

With MVP type 3, the managerial concerns shift to the relationship with marketing and manufacturing. Start-ups need to validate their business model, to test the market feedback on their products (C04). They also need to decide the cost of their products. There is a risk here as the manufacturing cost will determine the product cost (C09).

Requirements are typically challenging when hardware product in nature is quality-focused but the start-up context requires also the product development change-prone under multiple influences. Ensuring hardware products comply with regulatory requirements is a complex and ever-changing process. The process implies a significant administrative component that affects the design, development and testing of a hardware component. Requirement management might involve domain experts, as in the healthcare sector in Case E:

> *"Since this is a medical company, there are very strong requirements for ISO certification and standards. So we will now introduce a process management tool to describe and create processes. Now it's very ad-hoc. We are not so many either, so there is no need to have a rigid system or system at all. Because of the medical/regulatory, we must follow certain processes that must be in place"* (Case E)

In a start-up context, the uncertainty in start-ups' environment probably requires them to be more agile than is suggested in existing Agile approaches and frameworks. Uncertainty could come from market factors, that is uncertainty about customers' demands and level of satisfaction with products and from technological factors, that is requests to upgrade products due to new technologies. In most of our cases, start-ups defined what features to include using market-driven approaches. Hence, a set of requirements became more or less clear after performing market research.

Design: Internet of Things applications often involve much upfront design. In the context of a start-up, the design of a hardware-based product involves consideration of both its hardware and software components. A relevant design issue can be, for instance, the time required to choose a suitable hardware platform (C10). Complexity might arise from the fact that start-ups need to provide a product with more advanced features than the existing products in the market (C12). In Case A, the engineers took a long time to discuss and select an Internet of Things solution among available platforms, that is Arduino, Raspberry, ESP8266, Beaglebone Black, Particle.io and Intel Edison. This selection is made even more difficult by the fast rate of change in hardware technology.

For start-ups that build their products from scratch, the complexity of design is much more daunting. Architectural decisions depend on the selected components for central processing technologies, sensors and IoT nodes. Decisions sometimes also depend on the efficiency and cost of manufacturing such components (C11). Sometimes, architectural decisions need to be made in uncertain situations:

> *"Many such things make it time-consuming in the process but it is hard to know what is going to mix things up later. I think it is better to just take a decision, cause even if you spend a lot of time planning what choice to make, it does not necessarily make the product better."* (Case B)

Start-ups that use open-source components need to consider the degree of openness and the level of support available from the open-source community (C13):

> *"The basic software, it is open source NuttX-based, so in principle, . . . we will do contributions to . . . NuttX community. So, to get the open device you just take the NuttX (release) and make it run there. But then if you need, configuration engine or any more specific API then you need to contact . . . "* (Case B)

Last but not least, hardware design constraints can also come from the fact that some domains, such as healthcare and aerospace, require that products and product development procedures follow strictly defined process models and satisfy compliance standards (C14).

Implementation involves both constructions of hardware and software components. Some start-ups (e.g., Case D, F) struggled with finding suitable open-source software components for system-level design (C20). Some others (A, C, E, G) described problems with finding the balance between profitable hardware modules and hardware quality (C17). Prototyping is sometimes paused due to the lack of availability of required components (Case F, G). Some start-ups also reported facing issues with feature creep due to the design of a multi-purpose product (C16):

> *" . . . It is a type of reference design with the maximum number of features. You can then drop off things. Also partially you can extend it but it is not in the integrated package then. There are means to extend the device but it is external"* (Case C)

Testing is a relevant theme related to early-stage hardware prototyping. Many non-functional attributes need to be assured even at the prototyping stage, such as reliability, robustness and performance (C21). Moreover, the lab environment in which hardware prototypes are made often does not represent the actual operating environments. Testing seemed to be done in an ad hoc manner in most cases. For instance, the CTO of Case B mentioned:

> *"We are testing the subsystems ourselves. We have no structured system for testing. You make something, then you test if it works. Then you deliver the whole system of components to a user and ask if they like the product or if there are any bugs etc. Then we map the feedback back to the subsystems, work on them and try again."* (Case B)

In Case A, the first prototype was a failure as its connectivity required a Wi-Fi network. This was not practical for fish farms, which lacked such a network connection. Testing at the system level was also a challenge when many components were locked in third-party vendors. Quality assurance also needs to consider the evolution of both hardware, firmware and software parts (C23). CTO of Case D mentioned their testing practices for the first prototype:

> *" . . . when upstreaming the testing responsibility so whenever some working product was done so the makers tested it to some degree...Overall maybe you could say, 10 to 20 percent for the first prototypes"* (Case D)

5. Discussions

5.1. Software and Hardware in Internet of Things Applications

In this work, several challenges when creating MVPs in Internet of Things start-ups were recognized. Although not all of the challenges appear in every start-up, they occur in at least three of our cases, which can be argued to be a hardware start-up MVP issues. The challenges occurred in all types of product engineering areas, that is management, requirement, design, implementation and testing. Due to the nature of start-up stages, the identified challenges are mainly found in early stages. However, we also found challenges in transforming and managing MVPs in the later stages.

It is noted that not every companies in the Internet of Things domain focusing on physical products. Many start-ups provide digital services on top of existing Internet of Things platforms, leveraging business-driven analysis of internal data and operational insights [17]. In this way, software prototyping techniques are valid and relevant to Internet of Things development. Together with knowledge about software MVPs [11,12,44,45], the study adds to the understanding of MVPs development in Internet of Things start-ups. In general, we found that Internet of Things MVPs are developed under a similar organizational context, that is limited budget, multiple dependency, prone-change and high-risk environments. Common issues with making MVPs were, for instance, difficulties in finding the right set of user feedback [46], team communication [11], the pressure of time to market [11], openness and access to external support for design, non-functional testing and technology dependencies [11].

We found that the inclusion of hardware parts in Internet of Things products infers an additional complexity and dependency in the technical, financial and resource-related aspects. Hardware MVP is also taken a relatively long time to achieve its purpose. Even with the advancement in prototyping technologies, it is still a high-risk activity to create MVPs for desired quality attributes, such as performance, reliability, safety and security. Design issues, that is choosing hardware platforms, finding suitable hardware components, integrating software and hardware design and ensuring compliance with quality standards are more prominent in hardware-related products than in software

systems. Funding is a more critical issue in hardware start-ups, causing a higher entry threshold for hardware innovators. Software development in start-ups is lightweight and speed-driven [5]. Many MVPs were done to interact with external stakeholders and proofs-of-concept, without being actually functional. Open-source components, simulating functionalities and focusing on UX parts are all practices that come with risks [11,47]. Hardware MVPs depend more on the quality of the hardware engineers on the start-up's team. Due to the long R&D process, it is more demand on documentation to keep important knowledge inside the start-up when a hardware engineer leaves. Dependent on hardware vendors also makes it difficult for hardware start-ups to fully control quality and the time to release their products. This would be a more critical issue than that in software start-ups.

5.2. Implications for Practitioners

Internet of Things entrepreneurs should be aware of three types of MVPs and various approaches to implementing them. Start-ups evolve with the transformation from one MVP to another, which is closer to the final functional product. However, our evidence does not show a systematic way for this transformation. Entrepreneurs should be aware of available means in hand and their associated risks. For Internet of Things MVPs, it typically takes a long time to complete, hence the consequent of the risks would be more severe for the survival of Internet of Things start-ups.

As MVP prototyping is an essential activity for early-stage development of start-ups, entrepreneurs should leverage advanced technologies in prototyping and carefully plan the prototyping iterations. Customer involvement should be encouraged even in early-stage activities. Entrepreneurs should consider a systematic way of working that enables efficient engineering activities in a dynamic and uncertain environment. For start-ups who recruit contractors, the possible consequences of external dependencies on engineering processes and practices should be considered.

When entrepreneurs develop their value propositions based on a hardware/software product, they can be aware of relevant issues as shown in Figure 6. Besides the common challenges for high-tech start-ups, software development should be aware of matters such as user-centred design, feature creeps and technical debts. When the start-up develops a value based on both software and hardware, they would face with a more comprehensive set of issues. In such a case, it is needed to identify which parts of the system providing core values. The core elements should be managed in-house with internal competence. The elements that are secondary could be delayed or outsourced to avoid challenges.

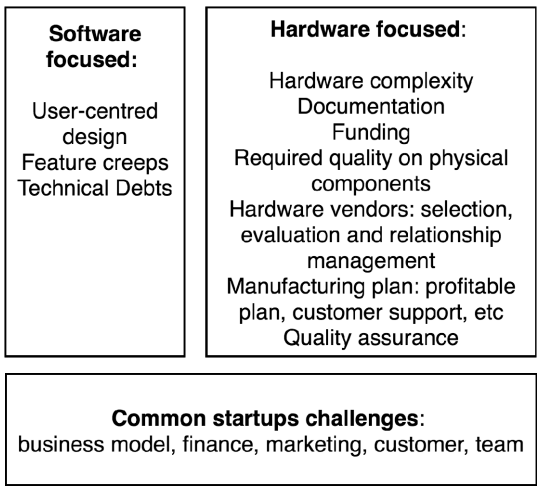

Figure 6. A comprehensive view of challenges for developing Internet of Things MVPs.

5.3. Threats to Validity

There are several threats to the validity of this study that are worth discussing [37]. There is a concern that our sample does not provide insight from actual Internet of Things start-ups. Physical

cameras and smart home devices are typical examples of modern hardware products. We included start-ups in stages from a pre-start-up phase (a few people doing prototypes) to a post-start-up phase (a case with 85 employees). Although some prototypes had a focus on the software parts, that focus reflects the value proposition of their products. Our cases came from three different countries across two continents and offered a variety of observations.

Another internal threat to validity is bias in the data collection, as the data might not represent the comprehensive aspects of the cases. Most of our cases are represented by one interview, which might not represent the complete scenario in our study context. In order to mitigate this threat, we selected CTOs and CEOs as interviewees where possible, who we assumed should have the best understanding of their start-ups. For Case A and C, we performed follow-up clarification via interviews and emails. We also used online documents and websites to increase our understanding of the cases (Case F, G, H). We reached data saturation by finding no new information emerged from adding new cases [36]. Our data collection and analysis were done iteratively over one year of research. The final cases (Case B, F, G, H) did not bring much new insight, which implied that the explanatory power of the core category was fulfilled.

A threat to construct validity is the possibly inadequate descriptions of identified concepts. We tried to collect contextual information about the start-ups using online documents and materials. When analysing the data, the coding process of interview transcripts was assisted by the authors' prior knowledge about prototyping and software development.

As far as external validity is concerned, our cases are characterized by Scandinavian and Pakistani start-ups in early stages with bootstrap financing models. Hence, the findings might not be generalizable to other types of start-ups, for example, internal corporate start-ups, start-ups with venture capital investment and European and North American start-ups. Hence, the observed challenges with prototyping and hardware development might not be directly applicable to such contexts, through analytical generalization may be possible in similar contexts.

6. Conclusions

A considerable number of start-ups deals with hardware components in their value propositions, such as connected devices, wearable devices and robotics. Like software development, hardware-related product development also requires practices and processes. Start-ups provide a unique context where traditional engineering and practices need to be reconsidered and validated. Towards a guideline for engineering high-tech products for start-ups, the first explorative step is to reveal the state-of-practice and concerns that existing Internet of Things start-ups face with.

We classified early stage start-ups' deliveries into MVP type 1, type 2 and type 3. Various approaches are found to implements these MVPs, including iterative development, rapid prototyping, outsourcing, third-party components and in-house design. Start-ups move forwards by transforming from Type 1 to Type 2 and from Type 2 to Type 3. However, there are no common paths for such transformation. Among eight Internet of Things start-ups, we gathered 24 common concerns when developing their MVPs. The concerns belong to management, requirement design, implementation and testing areas. Particularly, Internet of Things start-ups heavily rely on third-party vendors in term of development speed and final product quality. The complexity of desired products needs to be matched with available competence and appropriate management strategies.

To the best of our knowledge, this work is among the first studies of hardware product development in start-up companies. The popularity of Internet of Things start-ups and the current nascent stage of research in this field call for greater attention and further empirical research. We have identified several directions for future work. Future research can investigate the applicability of software MVP approaches in hardware-related product development. We plan to perform a survey to confirm our findings on a larger scale and we call for further collaborative case study research in this direction. Second, we found that Internet of Things start-ups tend to apply Agile practices but in an ad-hoc manner. Environmental factors characteristic of start-ups and their impact on Internet of

Things development must be better understood. Future work should provide a framework or a guideline to apply Agile hardware product development in the context of start-ups. Last but not least, our study barely touched upon the concern of balancing speed and quality in hardware-related product development. An investigation of how start-ups successfully achieve this balance throughout their evolution will be beneficial to early-stage Internet of Things start-ups.

Author Contributions: Conceptualization, A.-N.D.; methodology, A.N.-D.; investigation, A.N.-D., K.K.; resources, T.L.; writing—A.N.-D., K.K.; writing—review and editing, A.N.-D., T.L., S.S.B.; visualization, A.N.-D.

Funding: This research received no external funding.

Acknowledgments: We thank hardware start-ups for sharing their insights with us.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Interview Guideline

Section 1: Business background

Q.1.1: Describe your product?
Q.1.2: Describe your company? Brief history, current head counts, departments etc.
Q.1.3: What is your software development methodologies, processes, environments and tools

Section 2: Idea visualization and prototyping

Q.2.1: When did the idea came into your mind?
Q.2.2: How did you built the first prototype?
Q.2.3: What was your learning from the prototyping?
Q.2.4: Is the initial idea and the current product same? In terms of product, finances, team etc.
Q.2.5: When and where did you first launch your product?

Section 3: Product development and launching

Q.3.1: When the actual development started?
Q.3.2: Were the customers involved during the product development?
Q.3.3: How the current product is different from the prototype?

Section 4: Challenges and lessons learnt

Q.4.1: What were the three biggest challenges?
Q.4.2: What would you do differently?

Section 5: Adoption of paradigm

Q.5.1: Which is your preferable model for Internet of Things software development such as water fall, agile, etc?
Q.5.2: What are Agile practices you have used in your companies? Do they work?
Q.5.3: How do you balance between the speed of development and quality of the product?

References

1. Lasi, H.; Fettke, P.; Feld, T.; Hoffmann, M. Industry 4.0. *Bus. Inf. Syst. Eng.* **2014**, *6*, 239–242. Available online: https://aisel.aisnet.org/bise/vol6/iss4/5 (accessed on 15 February 2019). [CrossRef]
2. Casadei, R.; Fortino, G.; Pianini, D.; Russo, W.; Savaglio, C.; Viroli, M. Modelling and simulation of Opportunistic IoT Services with Aggregate Computing. *Future Gener. Comput. Syst.* **2019**, *91*, 252–262. [CrossRef]
3. Gustavsson, T.; Rönnlund, P. Agile adoption at Ericsson hardware product development. In Proceedings of the 22nd NFF Nordic Academy of Management Conference, Reykjavik, Iceland, 1–23 August 2013.

4. Lee, E.A. *Computing Foundations and Practice for Cyber-Physical Systems: A Preliminary Report*; Technical Report No. UCB/EECS-2007-72; University of California, Berkeley: Berkeley, CA, USA, 21 May 2007.
5. Paternoster, N.; Giardino, C.; Unterkalmsteiner, M.; Gorschek, T.; Abrahamsson, P. Software development in startup companies: A systematic mapping study. *Inf. Softw. Tech.* **2014**, *56*, 1200–1218. [CrossRef]
6. Nguyen-Duc, A.; Seppänen, P.; Abrahamsson, P. Hunter-gatherer Cycle: A Conceptual Model of the Evolution of Software Startups. In Proceedings of the 2015 International Conference on Software and System Process, Tallinn, Estonia, 24–26 August 2015; pp. 199–203.
7. Unterkalmsteiner, M.; Abrahamsson, P.; Wang, X.F.; Nguyen-Duc, A.; Shah, S.; Bajwa, S.S.; Baltes, G.H.; Conboy, K.; Cullina, E.; Dennehy, D.; et al. Software Startups: A Research Agenda. *e-Inf. Softw. Eng. J.* **2016**, *10*, 89–123. [CrossRef]
8. Ries, E. *The Lean Startup: How Constant Innovation Creates Radically Successful Businesses*; Penguin Group: London, UK, 2014.
9. Giardino, C.; Bajwa, S.S.; Wang, X.; Abrahamsson, P. Key Challenges in Early-Stage Software Startups. In Proceedings of the Agile Processes in Software Engineering and Extreme Programming, Helsinki, Finland, 25–29 May 2015; pp. 52–63.
10. Batova, T.; Clark, D.; Card, D. Challenges of lean customer discovery as invention. In Proceedings of the 2016 IEEE International Professional Communication Conference (IPCC), Austin, TX, USA, 2–5 October 2016; pp. 1–5.
11. Nguyen-Duc, A.; Wang, X.; Abrahamsson, P. What Influences the Speed of Prototyping? An Empirical Investigation of Twenty Software Startups. In Proceedings of the Agile Processes in Software Engineering and Extreme Programming, Cologne, Germany, 22–26 May 2017; pp. 20–36.
12. Seppänen, P.; Liukkunen, K.; Oivo, M. Little Big Team: Acquiring Human Capital in Software Startups. In Proceedings of the International Conference Product-Focused Software Process Improvement, Innsbruck, Austria, 29 November–1 December 2017; pp. 280–296.
13. Larman, C.; Basili, V.R. Iterative and incremental developments: A brief history. *Computer* **2003**, *36*, 47–56. [CrossRef]
14. Jacobson, I.; Spence, I.; Ng, P.-W. Is There a Single Method for the Internet of Things? *Queue* **2017**, *15*, 20. [CrossRef]
15. Taivalsaari, A.; Mikkonen, T. A Roadmap to the Programmable World: Software Challenges in the IoT Era. *IEEE Softw.* **2017**, *34*, 72–80. [CrossRef]
16. Zambonelli, F. Key Abstractions for IoT-Oriented Software Engineering. *IEEE Softw.* **2017**, *34*, 38–45. [CrossRef]
17. Fortino, G.; Russo, W.; Savaglio, C.; Shen, W.; Zhou, M. Agent-Oriented Cooperative Smart Objects: From IoT System Design to Implementation. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 1939–1956. [CrossRef]
18. Collin, T. A Methodology for Building the Internet of Things. Available online: https://www.slideshare.net/IoTMethodology/a-methodology-for-building-the-internet-of-things-42112202 (accessed on 15 February 2019).
19. Blank, S. *The Four Steps to the Epiphany: Successful Strategies for Startups That Win*; BookBaby: Pennsauken, NJ, USA, 2005.
20. Nguyen-Duc, A.; Shah, S.M.A.; Ambrahamsson, P. Towards an Early Stage Software Startups Evolution Model. In Proceedings of the 2016 42nd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Limassol, Cyprus, 31 August–2 September 2016; pp. 120–127.
21. SBA Startups & High-Growth Businesses | The, U.S. Small Business Administration | SBA.gov. Available online: www.sba.gov (accessed on 15 February 2019).
22. Chen, E. *Bringing a Hardware Product to Market: Navigating the Wild Ride from Concept to Mass Production*; Penguin Group: London, UK, 2015.
23. Di Resta, R.; Forrest, B.; Vinyard, R. *The Hardware Startup: Building Your Product, Business and Brand*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2015.
24. Li, Y. An Integrated Platform for the Internet of Things Based on an Open Source Ecosystem. *Future Internet* **2018**, *10*, 105. [CrossRef]
25. Pantiuchina, J.; Mondini, M.; Khanna, D.; Wang, X.; Abrahamsson, P. Are Software Startups Applying Agile Practices? The State of the Practice from a Large Survey. In Proceedings of the Agile Processes in Software Engineering and Extreme Programming, Cologne, Germany, 22–26 May 2017; pp. 167–183.

26. Bosch, J.; Holmström Olsson, H.; Björk, J.; Ljungblad, J. The Early Stage Software Startup Development Model: A Framework for Operationalizing Lean Principles in Software Startups. In Proceedings of the International Conference Lean Enterprise Software and Systems, Galway, Ireland, 1–4 December 2013; pp. 1–15.
27. Nguyen-Duc, A.; Weng, X.; Abrahamsson, P. A Preliminary Study of Agility in Business and Production: Cases of Early-stage Hardware Startups. In Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Oulu, Finland, 11–12 October 2018; pp. 51:1–51:4.
28. Giardino, C.; Paternoster, N.; Unterkalmsteiner, M.; Gorschek, T.; Abrahamsson, P. Software Development in Startup Companies: The Greenfield Startup Model. *IEEE Trans. Softw. Eng.* **2016**, *42*, 585–604. [CrossRef]
29. Hugues, J.; Zalila, B.; Pautet, L.; Kordon, F. From the Prototype to the Final Embedded System Using the Ocarina AADL Tool Suite. *ACM Trans. Embed. Comput. Syst.* **2008**, *7*, 42. [CrossRef]
30. Slomka, F.; Dorfel, M.; Munzenberger, R.; Hofmann, R. Hardware/software codesign and rapid prototyping of embedded systems. *IEEE Des. Test Comput.* **2000**, *17*, 28–38. [CrossRef]
31. Greene, B. Agile methods applied to embedded firmware development. In Proceedings of the Agile Development Conference, Salt Lake City, UT, USA, 22–26 June 2004; pp. 71–77. [CrossRef]
32. Cordeiro, L.; Mar, C.; Valentin, E.; Cruz, F.; Patrick, D.; Barreto, R.; Lucena, V. An Agile Development Methodology Applied to Embedded Control Software Under Stringent Hardware Constraints. *ACM SIGSOFT Softw. Eng. Notes* **2008**, *33*, 5:1–5:10. [CrossRef]
33. Dos Santos, D.; da Silva, I.N.; Modugno, R.; Pazelli, H.; Castellar, A. Software Development Using an Agile Approach for Satellite Camera Ground Support Equipment. In *Advances and Innovations in Systems, Computing Sciences and Software Engineering*; Elleithy, K., Ed.; Springer: Dordrecht, The Netherlands, 2007; pp. 71–76.
34. Kaisti, M.; Rantala, V.; Mujunen, T.; Hyrynsalmi, S.; Könnölä, K.; Mäkilä, T.; Lehtonen, T. Agile methods for embedded systems development—A literature review and a mapping study. *EURASIP J. Embed. Syst.* **2013**, 15. [CrossRef]
35. Ronkainen, J.; Abrahamsson, P. Software Development under Stringent Hardware Constraints: Do Agile Methods Have a Chance? In Proceedings of the Extreme Programming and Agile Processes in Software Engineering, Genova, Italy, 25–29 May 2003; pp. 73–79.
36. Aberdeen, T. Yin, R. K. (2009). Case study research: Design and methods (4th Ed.). Thousand Oaks, CA: Sage. *Can. J. Action Res.* **2013**, *14*, 69–71.
37. Runeson, P.; Höst, M. Guidelines for conducting and reporting case study research in software engineering. *Empir. Softw. Eng.* **2008**, *14*, 131. [CrossRef]
38. Palinkas, L.A.; Horwitz, S.M.; Green, C.A.; Wisdom, J.P.; Duan, N.; Hoagwood, K. Purposeful sampling for qualitative data collection and analysis in mixed method implementation research. *Adm. Policy Ment. Health Ment. Health* **2015**, *42*, 533–544. [CrossRef] [PubMed]
39. Hove, S.E.; Anda, B. Experiences from conducting semi-structured interviews in empirical software engineering research. In Proceedings of the 11th IEEE International Software Metrics Symposium (METRICS'05), Como, Italy, 19–22 September 2005. [CrossRef]
40. Dixon-Woods, M.; Agarwal, S.; Jones, D.; Young, B.; Sutton, A. Synthesising qualitative and quantitative evidence: A review of possible methods. *J. Health Serv. Res. Policy* **2005**, *10*, 45–53. [CrossRef] [PubMed]
41. Braun, V.; Clarke, V. Using thematic analysis in psychology. *Qual. Res. Psychol.* **2006**, *3*, 77–101. [CrossRef]
42. Cruzes, D.S.; Dyba, T. Recommended Steps for Thematic Synthesis in Software Engineering. In Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement, Banff, AB, Canada, 22–23 September 2011; pp. 275–284.
43. Society, I.C.; Bourque, P.; Fairley, R.E. *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*, 3rd ed.; IEEE Computer Society Press: Los Alamitos, CA, USA, 2014; ISBN 978-0-7695-5166-1.
44. Eisenmann, T.; Ries, E.; Dillard, S. Hypothesis-Driven Entrepreneurship: The Lean Startup. Harvard Business School Background Note 812-095. December 2011. Available online: https://www.hbs.edu/faculty/Pages/item.aspx?num=41302 (accessed on 15 February 2019).
45. Olsson, H.H.; Bosch, J. The HYPEX Model: From Opinions to Data-Driven Software Development. In *Continuous Software Engineering*; Springer International Publishing: Cham, Switzerland, 2014; pp. 155–164. [CrossRef]

46. Rissanen, O.; Münch, J. Transitioning towards Continuous Delivery in the B2B Domain: A Case Study. In Proceedings of the Agile Processes in Software Engineering and Extreme Programming, Helsinki, Finland, 25–29 May 2015; pp. 154–165.
47. Duc, A.N.; Jabangwe, R.; Paul, P.; Abrahamsson, P. Security Challenges in IoT Development: A Software Engineering Perspective. In Proceedings of the XP2017 Scientific Workshops, Cologne, Germany, 22–26 May 2017; pp. 11:1–11:5.