

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Duc, Anh Nguyen; Khalid, Khan; Lønnestad, Tor; Bajwa, Sohaib Shahid; Wang, Xiaofeng; Abrahamsson, Pekka

Title: How do startups develop internet-of-things systems : a multiple exploratory case study

Year: 2019

Version: Accepted version (Final draft)

Copyright: © 2019, IEEE

Rights: In Copyright

Rights url: <http://rightsstatements.org/page/InC/1.0/?language=en>

Please cite the original version:

Duc, A. N., Khalid, K., Lønnestad, T., Bajwa, S. S., Wang, X., & Abrahamsson, P. (2019). How do startups develop internet-of-things systems : a multiple exploratory case study. In ICSSP '19 : Proceedings of the International Conference on Software and System Processes (pp. 74-83). IEEE. <https://doi.org/10.1109/ICSSP.2019.00019>

How do startups develop Internet-of-things systems- A multiple exploratory case study

Anh Nguyen Duc

University College of South Eastern
Norway
angu@usn.no

Khan Khalid

PAF - Karachi Institute of Economics
and Technology, Pakistan
khalid.khan@pafkiet.edu.pk

Tor Lønnestad

University College of South Eastern
Norway
Tor.Lonnestad@usn.no

Sohaib Shahid Bajwa

Department of Computer Science,
University of Calgary, Canada
sohaibshahid.bajwa@ucalgary.ca

Xiaofeng Wang

Free University of Bolzano,
Italy
Xiaofeng.Wang@unibz.it

Pekka Abrahamsson

University of Jyväskylä
Finland
pekka.abrahamsson@jyu.fi

***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 methodological commonalities among startups who are developing Internet-of-Things products. Using the SEMAT Essence framework, we captured common team compositions, common types of Minimum Viable Products and common way of working in early stage Internet-of-Things startups. We found that startups include various engineering and business competence, but do not cover all of what is needed. The development of Internet-of-Things applications adopts certain speed-favor approaches, i.e. rapid prototyping, iterative development and outsourcing. The finding implies some recommendations for both researchers and practitioners in the area of Internet-of-Things development.**

***Keywords*—Internet-of-Things, case study, hardware-related development, Minimum Viable Products, SEMAT Essence**

I. INTRODUCTION

The emergence of Internet-of-Things has an enormous potential to impact the future society and the way people and business are organized. Internet-of-Things reflects a system of a connected set of anyone, anything, anytime, anyplace, any service, and any network embodying characteristics of physical, networked, software, and of human-interactive systems [1]. With the Industry 4.0 revolution, the adoption and development of connected systems, such as smart grids, autonomous automobile systems, medical monitoring are becoming mainstream [2]. Unlike traditional embedded systems, which are designed as stand-alone devices, the focus is on networking several devices. The number of connected devices available in the worldwide market is approaching 15 billion [3]. According to Gartner's hype cycle [4], by 2020, such technologies will be in 95% of electronics for new product designs.

Internet-of-Things applications are under development and piloted deployment via existing infrastructures and ecosystems of large cooperates, such as Amazon, Cisco, Bosch, and AT&T. There are also startup companies who participate or even lead the development of some segments of Internet-of-Things. For instance, Ayla Networks¹, a startup found in 2010, delivers industry-leading device management and application enablement with current funding of 125 million USD. Once Internet-of-

¹ <https://www.aylanetworks.com/>

Things technologies are standardized and popularized, the next step would be to optimize the development and operation of Internet-of-Things systems that continuously deliver needed values to end-users.

From an architectural perspective, an Internet-of-Things is a comprehensive set of software-relevant elements, from cloud-based storage and data analysis, end-user applications, middleware and hardware devices and their connectivity [5]. Challenges for Internet-of-Things development are mentioned as “something old, something new” [6, 8]. From an Internet-centric view, software components in Internet-of-Things systems need to support all kinds of equipment or devices should interact effectively in order to collaborate in accomplishing the assigned tasks [5]. Besides, the challenge of capturing and analyzing all kinds of information in a coherent manner from heterogeneous and sensor-enabled devices is a new complexity in comparison to traditional embedded systems [5]. It is argued that there is a demand for new practices in developing Internet-of-Thing systems [5, 6]. Jacobson et al. mentioned that while there is no one-size-fits-all approach, the methodology for Internet-of-Things needs various methodological elements from existing development methods [8]. The development of such systems requires competence from software engineering, hardware and electronic engineering, and also domain expertise in various fields, such as medical, autonomous, etc. Larrucea et al. argued that new approaches to standard software engineering techniques are also needed—for example, rethinking configuration management in the context of the extremely dynamic, continuously reconfiguring systems [6].

Given this motivation, this paper attempts at framing some common elements of the Internet-of-Things development from methodological and process perspectives. We would like to understand what are common practices, challenges to the Internet-of-Things development from the Software Engineering paradigm. Such commonalities are then used to identify the key engineering components around which the process of designing and developing Internet-of-Things systems and applications could revolve. As far as we know, only a few existing works investigated the methodological aspect of Internet-of-Things applications [7, 8, 9, 10]. Furthermore, we focused on the startup context. Startups, new companies with limited resources, short operational histories, looking for scalable business models [5, 6, 16] appears as a special context in which traditional product development approaches might not be directly applicable. The need for the product development methodologies to be adapted to a projects scope, magnitude, complexity, and changing requirements in the startup context is generally acknowledged, however, there exists a lack in guidance on how software startups can adapt their process to their situational context [15]. The combination of Internet-of-things and startup contexts led us the Research Question:

RQ: How do startups develop their Internet-of-Thing systems in their early-stages?

Our paper is organized as follows. Section 2 presents related work, Section 3 is our methodology approach. Section 4 presents our findings, Section 5 is the discussion. Section 6 concludes the paper.

II. RELATED WORK

A. Digital startups

The term “startup” has been defined differently across various principles [11-14, 20, 21]. Steve Blank describes a startup as a temporary organization that aims to create high-tech innovative products without having a prior working history [13]. He further highlights that in a startup context, the business and its product should be developed in parallel. Eric Ries defines a startup as a human institution that is designed to create a unique product or service under extreme uncertainty [14]. Rather than a formal company, a startup should be considered as a temporary organizational state, that seeks a validated and scalable business model [21]. Hence, a company with a dozen employees can still be in a startup state to validate a business model or a market.

System and software products can be related to a startup context in different ways [11]. Based on the role of systems and software in their business model, Steininger classified four types of digital startups: IT-facilitated startups, IT-mediated startups, IT-bearing startup and IT-ubiquity startups [11]. In IT-facilitated model, the software systems can be used to facilitate business activities, without contributing directly to the core value creation. An example is the adoption of marketing software to collect and to analyze market feedback. In IT-mediated model, software systems are used to connect the startup with its clients. An e-

commerce startup would buy or develop an e-commerce website system to reach their buyers. In IT-bearing model, the system is used as part of the company's infrastructure and product. This includes companies that develop software, hardware products for the mass market. In IT-ubiquity model, the value creation of the company completely relies on the IT-mediated offering of the software or systems. The companies provide not only the development but also the operation and maintenance of the system or software.

Internet-of-Thing systems, for instance, smart home solutions, need to consider a comprehensive system, from a cloud-based storage and data analysis, end-user applications, middleware and hardware devices and their connectivity, hence representing a mixed hardware-software system. We refer to Internet-of-Things startups as companies whose core value propositions basing on Internet-of-Things applications.

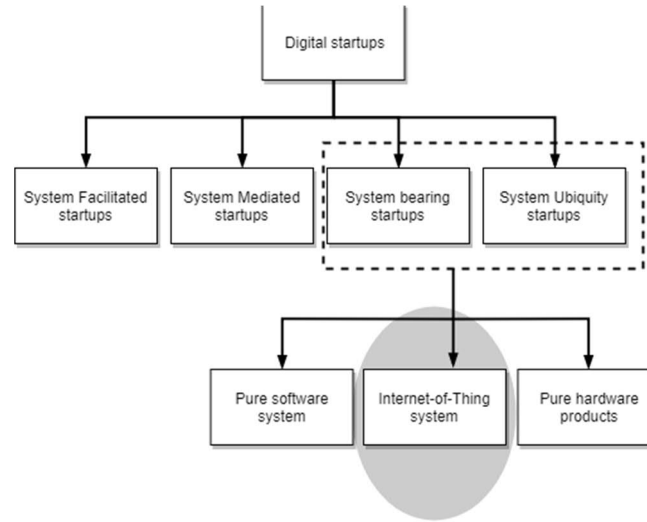


Figure 1: The position of Internet-of-Things startups (adjusted from [11])

B. Minimum Viable Products

Minimum Viable Product (MVP) can present for early-stage product development endeavor. MVP is defined as “a version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort” [14]. MVP is the core artifact of Lean startup methodology, which helps to conduct validated learning, and potentially evolving into final products. Eric Ries initiated the classification of MVP types, which is in common use in the startup communities [14]. 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. The role of MVPs is well understood as a key learning artifact in the software startup context. However, it is not known how MVPs are created and used in hardware-related development, specially Internet-of-Things systems.

Developers in software startups typically prioritize speed over quality when producing MVPs [15, 17]. Bosch et al. advocated for adjusting the Lean startup methodology to accommodate the development of multiple ideas and integrated them when the time for idea validation is too long [19]. Nguyen-Duc et al. studied the roles of MVPs in software startups and different approaches to speed up the process of making MVPs [18]. These studies provide a foundation for us to explore similar engineering perspectives in Internet-of-Things application context.

C. Internet-of-things development

The architecture of Internet-of-Things systems is commonly perceived with three layers (1) Perception Layer, (2) Network Layer and (3) Application Layer [30]. As shown in Figure 2, the **Perception** layer includes devices, objects that gather information from environments, human bodies, buildings, goods, etc. This includes 2-D bar code readers, RFID tags, camera, GPS, mobile phones, sensors and sensor network. The **Network** layer transmits and process data. Examples of technology in Network layer are 4G/5G,

Bluetooth, ZigBee, Wifi, SigFox and LoRa. The **Application** layer uses the processed data from the previous layers, providing the front-end of the whole Internet-of-Things architecture. Although many companies aiming at Internet-of-Things market by producing devices/ products/ services in one of the layers, we refer to Internet-of-Things startups as companies whose core value propositions base Internet-of-Things systems. As the key phenomenon in this paper, we refer to the product development as the process of bringing a product to market, from ideation, conceptualization, requirement elicitation, design to implementation and deployment.

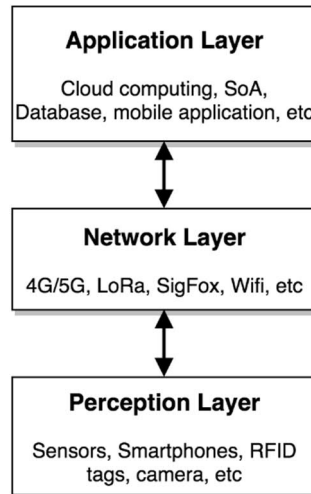


Figure 2: Three-layer architecture of Internet-of-things systems

The adoption of Software Engineering paradigms for the development of Internet-of-Things applications is mentioned in recent publications [7, 9, 10]. Zambonelli proposed a way to explore related engineering areas to identify a general model and methodology for Internet-of-Things software engineering [10]. Harrison et al. described engineering methods and tools for distributed component-based development of cyber-physical systems [7]. Usländer et al. presented a methodology that combined service engineering and Agile development in Internet-of-Things context [9]. From the theoretical perspective, these studies often propose methods without empirical validation. Furthermore, the methods are not specifically discussed in the context of startups. Jacobson et al. mentioned that while there is no one-size-fits-all approach, the methodology for Internet-of-Things needs various methodological elements from existing development methods [8]. Jacobson and colleges proposed a set of a language with a set of notations that helps developers to build their own methodologies [8, 33].

A broader view of the literature about embedded systems reveals possible adoptions of Software Engineering methodologies and tools in hardware-related development [22-25]. Ronkainen et al. described challenges with hard real-time requirements, prototyping, documentation, and test-driven development in Agile hardware development [22]. Greene reported a positive experience of applying Agile approaches in firmware development at Intel [23]. Adopting XP practices, Santos et al. showed a successful software version created for control of a satellite camera [24]. Kaisti et al. conducted a systematic mapping study about the adoption of Agile methodology in embedded systems development [25]. The authors suggested that Agile practices can be used in the embedded domain, but the practices need to be adapted to fit the strictly constrained field of embedded system development.

III. RESEARCH METHODOLOGY

This section describes the design and implementation of case studies that address our RQs.

A. Study Design

Following Yin's case study guideline [26], 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” [27]. A multiple-case study enables researchers to explore differences within and between cases. A large number of cases can help to triangulate findings, which was particularly important since we obtained only one, or at most, three interviews per case. We selected a startup as the unit of analysis and adopted a purposive sampling strategy to recruit cases [28]. As startups often focus on one product or one service from the beginning, in all of our cases, each startup focused on one product development project. There is often difficulty in identifying a real startup case among other similar phenomena, such as pure software startups, SMEs or part-time startups. Therefore, we clearly defined the criteria for our case selection:

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

From previous work [17, 29], we have constructed a contact list of 219 startups. Other co-authors of this work added 22 hardware startups to that list. From the total number of 241 startups, we identified 52 that potentially satisfied our case selection criteria. After contacting the companies via email, eight startups were willing to participate in this research. The first three cases were collected from startups 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 hardware startups.

B. Data collection

The major data source was semi-structured interviews. Semi-structured interviews are a common approach to collect relevant insights into many phenomena in software engineering [27]. The advantage of semi-structured interviews is that it allows improvisation and exploration of the objects studied [27]. Secondary data sources were published information about startups collected from the Internet. Prior to conducting interviews, we obtained as much information as possible regarding the startups’ 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 startups from the beginning. The years of technical experience of interviewees ranged from two years 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. During interviews, we adjusted the order of questions based on the responses and comments of the interviewees.

Table 1: Summary of qualitative data

Id	Profile of interviewees	# of interviews	Avg. years of exp.
Case A	CEO, CTO, hardware developer	3	2
Case B	CTO	1	2
Case C	CTO	2	10+
Case D	CEO	1	7+
Case E	CEO	1	5
Case F	CEO, CTO	2	3
Case G	Hardware developer	1	3
Case H	CEO	1	4
Total		12	

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, besides the questions about prototyping and product development approaches, we also asked participants to look at a list of Agile practices [34] and discuss challenges and opportunities these practices present in connection to hardware development (Agile practices are not discussed within the scope of this work). 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.

C. Data analysis

We adopted a thematic analysis, which is a common data analysis approach for qualitative data [31]. Thematic analysis is defined as “*a method for identifying, analyzing, and reporting patterns (themes) within data*” [31]. We adopted a mixed approach by exploring emerging concepts with a top-down guideline from a pre-defined set of themes. The approach has been applied and published in software engineering research venues [32]. Initially, we attempted to use the five steps of Design Thinking [35] to structure the interview data. This was reflected in the interview guideline and was expected to initiate general themes for our thematic analysis. When all interview transcripts were available, we found that not all Design Thinking steps were captured in all cases. Some cases we had very little information about the definition and ideation of the product. Besides, there were a lot of interesting stories into detail prototyping and product development techniques. This led to a decision of looking for a general framework to capture engineering activities.

We argue that the development of Internet-of-Thing application needs more than traditional software engineering and involves other disciplines. Hence, Software Engineering body of knowledge [36] might not be large enough to capture inter-disciplinary topics. When looking for a general engineering framework, we found SEMAT Essence [33], a language that captures seven essential elements (alphas) of software engineering, those that are integral to all software engineering methods and also applicable to inter-disciplinary engineering [33]. Alphas can be in certain states and are categorized in areas of concern. The seven alphas are Opportunity, Stakeholders, Requirements, System, Work Team and Way of Working. As the first step toward a comprehensive guideline for Internet-of-Things application development, we focus on the four alphas in endeavour dimension:

- Team with a concern of the collaboration of diverse competencies and skills to achieve team effectiveness
- System: made up of software, hardware, and data that provides its primary value by the execution of the software
- Work with a concern about the number of tasks to realize the opportunities. Work needs to be prepared, coordinated and tracked by teams.
- Way of working is about practices, processes, tactics, tools, methods, strategies that are used by the teams to accomplish work.

We scanned again interview transcripts and identified relevant pieces of texts, putting them into one of the three above categories. The texts were then labeled, grouped together under common concepts. Descriptive information associated with each concept was included in the concept description. The connections among concepts were also searched and documented.

D. Case description

Our case description, from Case A to Case H, is summarized in Table 2. Seven startups were characterized as being in the early stages and one startup in the later stage, as described as follows.

Case A is a startup 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 a seed funding under the South Korean government’s incubator program. Team A includes a CEO (business developer), four software developers, and one hardware engineer.

Table 2: Case demographics

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

Case B is a hardware startup 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 startup 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 startup 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 the 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 startup 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 startup developing an intelligent home automation system that allows users to control electric appliances remotely, using an Android app over either GSM or WiFi. 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-startup phase at the time of this study with regard to their financial situation and business development.

Case G is a hardware startup 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, etc. This startup arose from a university project.

During its first two months in an incubation program, they sold three product units. The startup won a national student-based entrepreneurial competition.

Case H is an Internet-of-Things startup 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.

IV. RESULTS

Figure 3 presents common themes among four alphas, system, work, team and way of working. Systems represent the delivered value propositions of the startups, according to three layers of Internet-of-things architecture. Work is summarized into demonstrable types of artifacts, MVPs and launching products. Teams are summarized into themes of required engineering competence (software engineering, hardware engineering, electronic engineering, product design), external competence that teams have an interface with (outsourcing partners, component providers, service providers, manufacturers), entrepreneurial competence (finance and marketing). Way of working emphasizes the common methods and practices to develop MVPs and released products. The detail of each section is presented below.

A. Team –the central catalyst

Digital startups are often initiated by an entrepreneur who seeks for new opportunities and being inspired by trending technologies, such as Internet-of-Things. In some cases, the founder is also a technology expert who would like to realize some of his/ her advanced knowledge into a beneficial business. Whether entrepreneurs are business-oriented or technology-oriented, it is common that startups do not have all needed competence in the beginning. As shown in Table 3, six out of eight cases involve external competence in the development of their MVPs. This takes a form of consultant work, third-party component providers, cloud service providers and manufacturers. In our sample, the major internal engineering competence is software development.

Table 3: Startups' team compositions

Id	External.	# ppl.	Software ppl	Hardware ppl
Case A	Y	6	4	1
Case B	Y	10	6	2
Case C	Y	4	1	2
Case D	Y	85	20	8
Case E	Y	20	15	3
Case F	N	8	6	1
Case G	Y	3	1	1
Case H	N	5	2	2

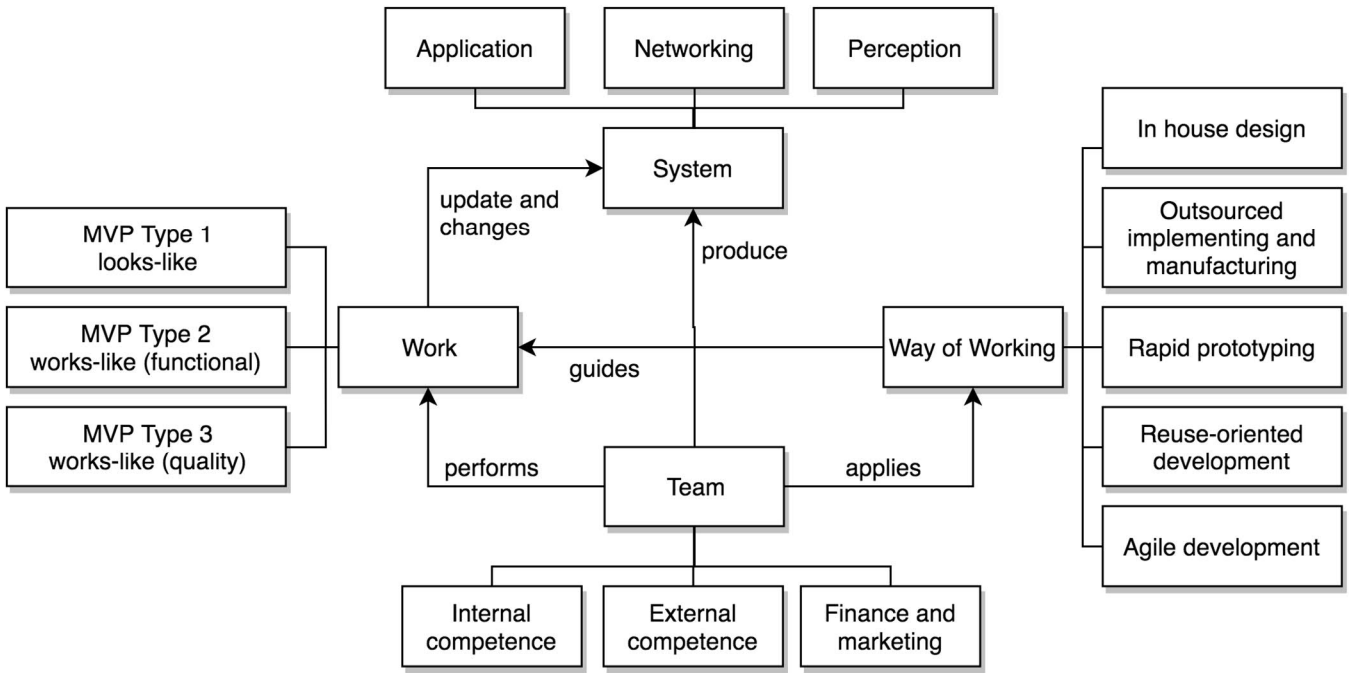


Figure 3: Early-stage endeavors for Internet-of-Things development in a startup context

B. Systems – the visualized visions of Internet-of-things

Systems implement the value propositions of startups and provide products or services for end users. We differentiate between two angles of Internet-of-Things, which are Internet-centric and Things-centric. Startups can obtain competitive advantages by delivering advanced Internet-centric features, i.e. with utilizing and analysis of sensor data. Startups can also compete in the market by providing advanced products or services in the network layer or the perception layer. Table 4 describes the value propositions in our cases. From each case, we also identified the starting point of system development, either with the Application layer or the Perception layer.

Table 4: Startups' team compositions

Id	Value propositions	Start first
Case A	Internet-centric	Application
Case B	Internet-centric	Perception
Case C	Thing-centric	Perception
Case D	Internet-centric	Application
Case E	Thing-centric	Application
Case F	Both	Application
Case G	Thing-centric	Perception
Case H	Thing-centric	Perception

C. Work – incremental implementation of systems

An early representation of work is captured by Minimum Viable Products. We used this concept to abstract technical details of work at the engineering level. We were also able to group MVPs into three types: MVP type 1 – looks-like, MVP type 2- functional works-like and MVP type 3 – quality works-like.

MVP type 1 has a 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)

We observed that MVP type 1 was made for both software and hardware components, i.e. using wireframe and mock-up tools for web/ mobile interfaces and 3-D printers for hardware units. MVP type 1 was used often for design, brainstorming, presenting ideas and mostly throwaway after usage.

MVP type 2 might involve hardware unit design, including sensor integration, chips, circuit design, and system engineering for physical units, and backend development for software components. The MVP would be demonstrable in term of functionalities:

“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).

The acceptable quality MVPs is reached at a so-called Minimum Acceptable Quality to enable the rapid prototyping. We also observed MVP type 2 with complete functionalities and operable with actual data:

“we have done in-house testing and then there was some field testing before the shipment. Then, of course, the most interesting data came from the real scenario” (Case D)

Table 5: Key MVPs in early stage startups

Id	Key MVPs	Duration	Type
Case A	A system-level prototype, which includes aqua environmental tracking sensors (e.g. pH and temperature), a mobile app for an aquaculture farming control system	9 months	Type 1
Case B	A module-level MVP with circuits and covers that can be tested by users	2 months	Type 2
Case C	A system-level MVP with fully integrated, operational camera features	28 months	Type 2
Case D	System-level MVP with sensors and IoT gateways, cloud storage and user interface that is fully functional	24 months	Type 3
Case E	System-level MVP includes tested sensors and integrated circuits, and a mobile app interface	18 months	Type 1
Case F	Single-feature MVP to switch energy saver with a simple data storage in the cloud	3 months	Type 2
Case G	Single-feature MVP with a simple eye tracker and motor controllers	5 months	Type 1
Case H	A module level MVP with a hardware unit	3 months	Type 1

Most of the architectural and functional design is 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)

Comparing to MVP type 2, an MVP type 3 focus on quality attributes, i.e performance of a function, scalability of a database, security of data, etc. **MVP type 3** involves the design and optimization of current MVPs to achieve best acceptable quality products. Non-functional testing is the main 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 the different parts, so they are easy to distinguish. The electronic boys alternate between the code and developing the product itself.” (Case C).

Table 5 presents the key MVPs in the early-stage of our cases. Among multiple MVPs and released prototypes, the selected MVPs were presented due to their perceived importance for business and customer development according to our interviewee.

Some common connections among different types of MVPs were also observed. We have seen that all of our cases started with MVP type 1. Even though looks-like MVPs are continuously created during the prototyping and product development, it is the first bridge between ideation and technical implementation. In most of the cases, we have seen the first focus of MVP development is functionalities. It is common that MVP type 2 coming after MVP type 1. In many cases, the early functions are developed for Application layer, i.e. web or mobile applications. 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 main focus in MVP type 3.

D. Way of Working – the balance of speed and quality

As shown in Figure 3, the development of three types of MVPs reveals a number of common way of working namely in-house design, outsourced implementation and manufacturing, rapid prototyping, reuse-oriented development and Agile development.

Outsourced implementation and manufacturing refer to the use of local contractors or subcontractors in startups' early stages. Examples of local contractors are consultant companies, maker space, student projects and manufacturers. In Case 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 startup, they facilitate the scaling-down activities when they lack funding or change business directions. Some startups 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 to 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).

While outsourcing of MVP elements is typically near-shored, manufacturing of products is cost-wise and from far-shored countries. It is mentioned by most of our cases that the production has happened or will happen in China.

Rapid prototyping: Internet-of-Things startups are beneficial from the advancement in 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. It has happened that it took 10 hours for a 3D printing of a small component. A lot of communication and changes happen already in the computer-based prototypes, i.e. with CAD designs.

Reused-oriented development, refers to the practice that developers utilize ready-made components, whether they are open source or commercial off the shelf, to speed up prototyping and product development. The reusability occurs in both application, network and perception layers. Many of our startups build their first prototype with Arduino and Raspberry Pi, open-source electronic prototyping platforms (Case A, B, C, F, G, H). All of the startups purchased sensor devices like accelerometer, gyroscope, light and touch sensors from third-party providers. Software components are frequently reused in incremental prototypes.

Agile development is the regular release of product versions. The duration of each iteration varies among our startups. 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. Internet-of-Things 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 startups. The CEOs related agility with less upfront planning, short-term driven evolution of the startups. They also mentioned the speed of prototyping, development and fast time-to-market when asked about Agile. Startups stated that full controls of development activities and partnership would prepare themselves to respond to unexpected changes. Some startups 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 software changes very much all the time. We are still trying to find what is the right way to do it” (Case C).

V. DISCUSSION

A. The development of MVPs in Internet-of-Things startups

A majority of tools and engineering methods for hardware-related systems are vendor-specific and designed to support closed-control environment [9]. Hardware components needed to be designed according to strict requirements of physical characteristics and domain-specific regulations. In this way, traditional, waterfall-like models are suitable for the R&D nature of Internet-of-Things applications. We observed that startups have experienced certain durations with focused product design and experimentation. The complexity of hardware design is one of the main barriers that prolong the prototyping duration. In some cases, the co-design of hardware and software components inside the devices and integration of Application, Network and Perception layers lead to long cycles of hardware prototyping. However, besides the hardware-related engineering activities, startups also deliver a system-level MVP that utilize a significant amount of software components. Hardware-relevant components could be contracted to avoid early complex hardware development. However, this also creates extra dependencies in early-stages. i.e. vendor dependency, platform dependency, competence dependency, etc. The amount of dependency seems to be a novel aspect that needs to be addressed with adjusted engineering methods.

Our cases also demonstrate the adoption of Agile methodologies in the context of Internet-of-Things. Agile adoption has been previously reported in the context of embedded systems [22-25]. We also noticed similar challenges with hardware-related development, such as hard real-time requirements, prototyping,

documentation and test-driven development [22]. In a firmware development project, Greene reported the use of 30-day Sprint duration [23]. In startup contexts, we found that Sprint planning is situational.

Usländer et al. described some Agile modeling best practices in his work, including active stakeholder participation, document continuously, iteration modelling and test-driven development [9]. We did not see that startups have utilized these practices in a systematic way. Instead, the MVP development in Internet-of-Things startups seems to be more likely to the MVP development in software startups [5], which is iterative, opportunistic and frequent use of work-around solutions. The immature state of our investigation does not allow a full explanation for this observation. However, we suspect that the nature of startup contexts, which always pose chaotic situations, and the fact that hardware development is not as a core engineering activity in our startups, leading to the less adoption of strict processes and engineering principles that are typically seen in embedded system development.

B. Threats to validity

There are several threats to validity worth discussing [27]. One internal threat to validities is the bias in the data collection, as the data might not represent the comprehensive case. Most of our cases are represented by one interview. In order to mitigate this threat, we selected CTO and CEO as interviewees, who have insights about their startups. For Case A and Case C, we have follow-up clarification via interview and emails. We also use other types of data sources, such as documents and observations to increase our understanding of the cases (Case F, G and H). A construct validity threat is the possible inadequate descriptions of identified concepts. We tried at our best to collect contextual information about the startups, from social media and personal contacts. When analyzing data, the coding process of interview transcripts was assisted by the authors' prior knowledge about prototyping and software development. For generalization validity, our case is characterized by Pakistani and Scandinavian startups. They are at early stages with bootstrap financing models. Hence, the observation might not yet be generalized to other types of startups, for example, internal cooperate startups, venture capital invested startups, and other European startups and American startups. Hence, the observed challenges with prototyping and hardware development might not be directly applied to such contexts, although analytical generalization may be possible in similar contexts.

VI. CONCLUSIONS

Internet-of-Things applications are predicted to be the next technological revolution with a huge impact on all industries and businesses. Existing software will likely to be modified and much new software will be developed as a part of Internet-of-Things systems. Methodologies to develop these systems for the mass market would be needed by companies in various application domains. This paper reveals state-of-practice on how startup companies are making their MVPs. Such commonalities would be a foundation to identify the key engineering components around which the process of designing and developing Internet-of-Things systems are consolidated.

Using the SEMAT Essence framework, we captured common team compositions, common types of MVPs and common way of working in early stage Internet-of-Things startups. We found that startups include various engineering and business competence, but do not cover all of what is needed. This leads to extra dependencies when producing MVPs. Three types of MVPs were found with the increase of functionalities and quality. Similar to software startups, the development of Internet-of-Things applications adopts certain speed-favor approaches, i.e. rapid prototyping, iterative development and outsourcing.

Our study contributes to the software startup and embedded system literature in several ways. We portray how startups initiate their Internet-of-Things development by evolving several MVPs. The comparison to software startups reveals the areas that need methodological adjustments to align with hardware-related characteristics of the whole Internet-of-Things systems. To the best of our knowledge, this is one of the first empirical work focusing on Internet-of-Things development processes and practices. The empirical investigation of emerging startups is also beneficial for practitioners who involve in hardware startups. They can be aware of possible practices and risks during the early-stages of product development.

Future work could go beyond the comparison between software startups and hardware startups. While there are many practices and processes are applying in software startup contexts, for example, Lean startup, they can be considered in an Internet-of-Things startup context as well. Moreover, a similar investigation can apply for a larger set of cases, which helps to increase the generality of our findings.

REFERENCES

- [1] E. A. Lee, "Computing foundations and practice for cyber-physical systems: a preliminary report," Tech. Rep. UCB/EECS-2007-72, University of California, Berkeley, May 2007
- [2] H. Lasi, P. Fettke, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, Oct. 2014.
- [3] G. Tomas, R. Peter, "Agile adoption at Ericsson hardware product development", Conference NFF, Reykjavik, Iceland, 2013
- [4] Gartner hype cycle 2018 [Accessed 01.02.2018] <https://www.gartner.com/smarterwithgartner/gartner-top-strategic-predictions-for-2018-and-beyond/>
- [5] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29(7), pp. 1645–1660, Sep. 2013.
- [6] X. Larrucea, A. Combelles, J. Favaro, and K. Taneja, "Software Engineering for the Internet of Things," *IEEE Software*, vol. 34 (1), pp. 24–28, Jan. 2017.
- [7] R. Harrison, D. Vera, and B. Ahmad, "Engineering Methods and Tools for Cyber-Physical Automation Systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 973–985, May 2016.
- [8] I. Jacobson, I. Spence, and P.-W. Ng, "Is There a Single Method for the Internet of Things?," *Commun. ACM*, vol. 60, no. 11, pp. 46–53, Oct. 2017.
- [9] T. Usländer and T. Batz, "Agile Service Engineering in the Industrial Internet of Things," *Future Internet*, vol. 10, no. 10, p. 100, Oct. 2018
- [10] F. Zambonelli, "Key Abstractions for IoT-Oriented Software Engineering," *IEEE Software*, vol. 34, no. 1, pp. 38–45, Jan. 2017.
- [11] D. M. Steininger, "Linking information systems and entrepreneurship: A review and agenda for IT-associated and digital entrepreneurship research," *Information Systems Journal*, vol. 1(1), 2018
- [12] A. Ghezzi, "Digital startups and the adoption and implementation of Lean Startup Approaches: Effectuation, Bricolage and Opportunity Creation in practice," *Technological Forecasting and Social Change*, Sep. 2018.
- [13] S. Blank, *The four steps to the epiphany*. 1st edn. CafePress, 2015
- [14] E. Ries, *The Lean Startup: How Constant Innovation Creates Radically Successful Businesses*. Penguin Group, London; 2014
- [15] J. Pantiuchina, M. Mondini, D. Khanna, X. Wang, and P. Abrahamsson, "Are Software Startups Applying Agile Practices? The State of the Practice from a Large Survey," in *Agile Processes in Software Engineering and Extreme Programming*, 2017, pp. 167–183.
- [16] G. Coleman and R. V. O'Connor, "An investigation into software development process formation in software startups," *Journal of Ent Info Management*, vol. 21(6), pp. 633–648, Oct. 2008.
- [17] A. Nguyen-Duc, X. Wang, and P. Abrahamsson, "What Influences the Speed of Prototyping? An Empirical Investigation of Twenty Software Startups," in *Agile Processes in Software Engineering and Extreme Programming*, 2017, pp. 20–36.
- [18] A. N. Duc and P. Abrahamsson, "Minimum Viable Product or Multiple Facet Product? The Role of MVP in Software Startups," in *Agile Processes, in Software Engineering, and Extreme Programming*, 2016, pp. 118–130.
- [19] J. Bosch, H. Holmström Olsson, J. Björk, and J. Ljungblad, "The Early Stage Software Startup Development Model: A Framework for Operationalizing Lean Principles in Software Startups," in *Lean Enterprise Software and Systems*, 2013, pp. 1–15.

- [20] S. M. Sutton, "The Role of Process in a Software Start-up", *IEEE Softw.*, 17(4), pp.33–39. <https://doi.org/10.1109/52.854066>, 2000.
- [21] Unterkalmsteiner M, Abrahamsson P, Wang X, Nguyen-Duc A, Shah S, Bajwa S, Shahid, Baltes GH, Conboy K, Cullina E, Dennehy D, Edison H, Fernandez-Sanchez C, Garbajosa J, Gorschek T, Klotins E, Hokkanen L, Kon F, Lunesu I, Marchesi M, Morgan L, Oivo M, Selig C, Seppänen P, Sweetman R, Tyrväinen P, Ungerer C, Yagüe A. 2016. Software Startups : A Research Agenda. *e-Informatica Software Engineering Journal*, 10 (1), 89-123. doi:10.5277/e-Inf160105
- [22] J. Ronkainen and P. Abrahamsson, "Software Development under Stringent Hardware Constraints: Do Agile Methods Have a Chance?," in *Extreme Programming and Agile Processes in Software Engineering*, 2003, pp. 73–79.
- [23] B. Greene, "Agile methods applied to embedded firmware development," in *Agile Development Conference*, 2004, pp. 71–77.
- [24] D. dos Santos, I. N. da Silva, R. Modugno, H. Pazelli, and A. Castellar, "Software Development Using an Agile Approach for Satellite Camera Ground Support Equipment," in *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, 2007, pp. 71–76.
- [25] M. Kaisti *et al.*, "Agile methods for embedded systems development - a literature review and a mapping study," *J Embedded Systems*, vol. 2013, no. 1, p. 15, Nov. 2013.
- [26] R. K. Jin, *Case study research: Design and methods (4th Ed.)*. Thousand Oaks, CA: Sage. 2009
- [27] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empir Software Eng*, vol. 14, no. 2, p. 131, Dec. 2008.
- [28] L. A. Palinkas, S. M. Horwitz, C. A. Green, J. P. Wisdom, N. Duan, and K. Hoagwood, "Purposeful sampling for qualitative data collection and analysis in mixed method implementation research," *Adm Policy Ment Health*, vol. 42, no. 5, pp. 533–544, Sep. 2015.
- [29] A. N. Duc, R. Jabangwe, P. Paul, and P. Abrahamsson, "Security Challenges in IoT Development: A Software Engineering Perspective," in *Proceedings of the XP2017 Scientific Workshops*, New York, NY, USA, 2017, pp. 11:1–11:5.
- [30] M. Wu, T. Lu, F. Ling, J. Sun and H. Du, "Research on the architecture of Internet of Things," in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, 2010, vol. 5, pp. V5-484-V5-487.
- [31] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, Jan. 2006.
- [32] D. S. Cruzes and T. Dyba, "Recommended Steps for Thematic Synthesis in Software Engineering," in *2011 International Symposium on Empirical Software Engineering and Measurement*, 2011, pp. 275–284.
- [33] I. Jacobson, P.-W. Ng, P. E. McMahon, I. Spence, and S. Lidman, "The Essence of Software Engineering: The SEMAT Kernel," *Commun. ACM*, vol. 55, no. 12, pp. 42–49, Dec. 2012.
- [34] Agile Alliance, "Subway Map to Agile Practices", [Accessed 14.03.2019] <https://www.agilealliance.org/agile101/subway-map-to-agile-practices/>
- [35] T. Brown, and K. Katz, "Change by design", *Journal of product innovation management*, vol. 28(3), 381-383, 2011
- [36] I. C. Society, P. Bourque, and R. E. Fairley, *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*, 3rd ed. Los Alamitos, CA, USA: IEEE Computer Society Press, 2014