Samuli Tolvanen

AGILE TRANSFORMATION PROCESS AT U-BLOX

Bachelor's thesis
Faculty of Information technology and communication sciences
MSc. Nyyti Saarimäki
11/2023

ABSTRACT

Samuli Tolvanen: Agile transformation process at u-blox
Bachelor’s Thesis
Tampere University
Bachelor’s Programme in Computing and Electrical Engineering
11/2023

Software development has grown complex over time. This phenomenon has created a need for new processes to manage the software development process itself. Growing complexity has led to the creation of agile techniques in software development. Recently the usage of agile methodologies like Scrum has increased. The ideas of the original Scrum guide were tailored for team-sized units. However, these principles have recently been adapted to larger organizations. Scrum for larger organizations is called Scaled Scrum. There are multiple different Scaled Scrum frameworks like Scrum of Scrums, Scrumconix, and LeSS have different principles and rules. Each organization has unique needs when it comes to the agile process, so there isn’t an out-of-box solution available.

This thesis examines u-blox, a Swiss semiconductor company that has lately created its agile model based on the Scrum of Scrums. Adaptation of this agile model has been a challenging process. This thesis aims to research the issues in service request processes towards the DevOps Value Unit. Additionally, this thesis focuses on issues of agile and communication.

The main research methods of this thesis are literature review and conducting interviews with employees from the DevOps Value Unit. The review of literature provides insights into general agile topics, while the interviews are conducted to uncover similar themes within the target organization. During the interviews, some key issues came up. The Scrum Master roles at the DevOps Value Unit were not well functioning due to a lack of proper training, the positions are also handled in addition to other duties. Some other difficulties that came up were related to communication tools and their usage in the target organization.

The data gathered from the interviews was compared with literature reviews. A few observations were made afterward: The scrum masters at the DevOps Value Unit needed proper Scrum Master training. It would also be useful to have training on how to use the communication tools in a similar manner with co-workers. Additionally, it would benefit the team to have discussions about agile and the reasons behind the process.

Keywords: Scrum, Scaled Scrum, Scrum of Scrums, Agile, Organizational Transformation, Software Organization, Value Based Organization

TIIVISTELMÄ

Samuli Tolvanen: Ketterän kehityksen muutosprosessi u-blox:lla
Kandidaatintyö
Tampereen yliopisto
Tieto- ja sähkötekniikan tekniikan kandidaattiohjelma: Ohjelmistotekniikka
11/2023

Ohjelmistokehitys on muuttunut monimutkaisemmaksi vuosien aikana. Monimutkaisuuden kasvu on johtanut tarpeelle uusille prosesseille, jotka hallitsevat ohjelmistokehitystä itseään. Tämä tarve on johtanut ketterien menetelmien (eng. agile) syntyyn. Ketterät menetelmät ovat prosesseja, jotka on tarkoitettu ohjelmistokehityksen hallintaan. Erityisesti viime vuosina ketterien menetelmien kuten Scrummin käyttö on yleistynyt. Alkuperäinen Scrum oli tarkoitettu tiimin kokoiselle yksikölle, mutta ketterien menetelmien laajentaminen koko organisaation tasolle on myös yleistynyt. Scrumia joka on tarkoitettu suuremmalle organisaatiolle kutsutaankin laajennetuksi Scrumiksi (eng. Scaled Scrum). Esimerkkejä tällaisista ovat Scrum of Scrums, Scrumconix, ja LeSS. Nämä ovatkin erilaisia oppaita siitä miten ketteryyttä voidaan laajentaa organisaatiotasolle. Yleisellä tasolla voidaan sanoa, että ketterää mallia ei voi suoraan kopioida standardista, vaan se pitää kustomoida erikseen jokaiselle organisaatiolle johtuen niiden erilaisista tarpeista ja vaatimuksista.

Tämän kandidaatintyö tutkii u-bloxia, sveitsiläistä puolijohdealan yritystä, joka on viime aikoina luonut oman ketterien menetelmien prosessinsa, joka perustuu Scrum of Scrumiin. Uuden ketterän menetelmän käyttöönottoprosessi on ollut haastava tutkielman kohdeorganisaatiolle. Tämän tutkielman tutkimuskohteena on DevOps organisaatio ja siihen kohdistuvat vaatimusprosessit ja niiden haasteet. Lisäksi tässä tutkielmassa selvitetään ketteriin menetelmiin perustuvan prosessin toimivuutta ja kommunikaatioon liittyviä haasteita kohdeorganisaatiossa.

Pääasialliset tutkimusmetodit tässä työssä ovat kirjallisuuskatsaus ja kohdeorganisaation työntekijöiden haastattelut. Kirjallisuuskatsauksen lisäksi tämä tutkielma sisältää DevOps kohdeorganisaation työntekijöiden haastatteluja. Kirjallisuuskatsauksen tavoitteena on kuvata ketteryyttä sen eri muodoissa. Tutkielmassa esiintyvät haastattelut liittyvät samoihin aiheisiin. Näissä haastatteluissa nousi esiin muutamia ongelmia. Scrum Masterin rooli nähtiin kohdeorganisaatiossa hankalaksi, sillä nykyisten Scrum Masterien tulee hoitaa roolia muiden työtehtäviensä lisäksi. Muut haastatteluissa esiin nousseet haasteet liittyivät käytössä olleisiin viestinnässä käytettyihin työkaluihin ja niiden käyttöön kohdeorganisaatiossa.

Haastatteluissa kerättyä aineistoa vertaillaan tutkielmassa kirjallisuuskatsauksen tuloksiin. Vertailun tuloksena tehtiin seuraavia johtopäätöksiä: Scrum Masterin rooleissa toimiville työntekijöille tarvittaisiin heidän rooliensa mukaista koulutusta, jotta he voisivat toimia rooleissaan paremmin. Lisäksi olisi hyödyllistä käydä keskusteluja ketteryydestä ja sen motiiveista yleisellä tasolla. Tämä mahdollistaisi kaikille työntekijöille samat tiedot, sekä ymmärryksen ketteryydestä ja prosesseista.

Avainsanat: Scrum, Skaalattu Scrum, Scrum of Scrums, Ketterä, Organisaatiomuutos, Ohjelmisto organisaatio, Arvopohjainen organisaatio.

PREFACE

I want to thank u-blox for the chance to do this thesis. I want to thank my colleagues for their support during the process. This project has been a great opportunity to learn about different u-blox organizations and our ways of working.

This thesis would not have been possible without the help of interviewees who shared their thoughts with me. Special thanks to the PO team. You helped me with your questions and ideas throughout the whole process.

Tampere, 29.11.2023

Samuli Tolvanen

CONTENTS

ABBREVATIONS AND NOTATIONS

Agile leader	Similar to scrum master but acts for larger organization
CI	Continuous integration
CPO	Chief product owner – Product owner for larger organization backlog
Customer	User of the product. Ex. If person creates a ticket for IT, they become customer of IT. At u-blox in this thesis context all customers are internal individuals or organizations
Developer	Member of value team. Creator of value
DevOps	In context of this thesis defines the target organization of the research
PO	Product owner
Value Organization	Agile Organization of Value Units at u-blox
Value Team	Agile team at u-blox
Value Unit	Agile team of teams at u-blox

1. INTRODUCTION

Software development has grown in complexity over time, and it continues to grow in complexity in the future as it has before. Software organizations need to keep up with the pace of changing frameworks and recent technologies. That creates a need for development processes where feedback is gathered about what works, what does not, and how to distribute resources. One solution to these challenges consist of agile development methodologies like Scrum [13].

Making changes to any organization developing software is challenging. These changes are necessary due to growing complexity of software products. Adapting an agile framework into an organization takes much work from everybody since it is a new mindset on how to cooperate and communicate with co-workers. This work must be daily and active while pursuing and adapting new ways of work. This thesis takes a look into agile implementation in an organization that has both software and embedded development functions.

The target organization of this thesis is u-blox. u-blox is a Swiss semiconductor company that has recently leaped towards agile and value-based organization. The first research question is: *How u-blox's agile transformation process is working for the DevOps Value Unit?* This question aims to find out if the agile process itself is working and whether there are any issues. It also investigates incoming requests and how they are made. The second question is: *How does communication work inside the DevOps Value Unit?* This question aims to find out whether the current tools and processes support agile communication with co-workers. The third and last research question is: *What is the meaning of agile at the DevOps Value Unit?* What they see working and what are the issues. Gathered observations from the target organization are being compared to literature and agile theorems to find conflicts and potential errors in the current agile process.

This thesis combines literature and theorems with practical implementation of them. Literature has been obtained from the ACM library, Google Scholar, and Andor. Keywords were related to multiple topics and fields. The most common keywords were agile, Scrum, scaled Scrum, software, embedded systems, and communication. The company data is from interviews of company employees and company guidelines about roles and processes.

Section 2 has a description of the organization and some graphs of how the organization works. Section 3 manages multiple topics around agile software organizations in general. Some Scrum-based agile theorems are included as well. Section 4 describes how the company employees see the ongoing agile process and how that compares to company guidelines. Section 5 compares the results of the interview to the literature. Section 6 has a summary and discussion.

2. SCRUM AND AGILE

Modern software development has grown extraordinarily complex over time. Software development processes include many phases that vary from the planning to post-delivery customer support. Complex processes need larger workforces to succeed. While the headcount is growing, managing the workforce becomes more difficult. The complexity creates a need for new agile ways to work and manage the development process. This has led to the development of new agile methodologies in software development. The most popular of these is some implementation of the original Scrum [13].

This section mainly consists of agile theorems and a description of agile based on the literature. Section 2.1 introduces the role of scrum and agile in a software organization. Section 2.2 talks about the scaling of the Scrum to the organizational level. Section 2.3 identifies challenges in agile for embedded systems. Section 2.4 is an extensive analysis of the role of communication in agile.

2.1 Scrum and agile methods in a software organization

Many of the agile methodologies that exist today originate from ideas of the original Scrum Guide. The Scrum guide is based on empiricism and lean-based thinking. Empiricism asserts that knowledge comes from experience and making decisions based on observations [13]. The idea in Scrum is that all development should happen iteratively, and all work done aims to add value to the customer. Scrum introduces all kinds of new concepts and tools for software development. It introduces new roles such as product owner and Scrum master. Fulfilling customer needs iteratively is the goal of Scrum [12]. This requires making well-defined requirements and requests to the developing team.

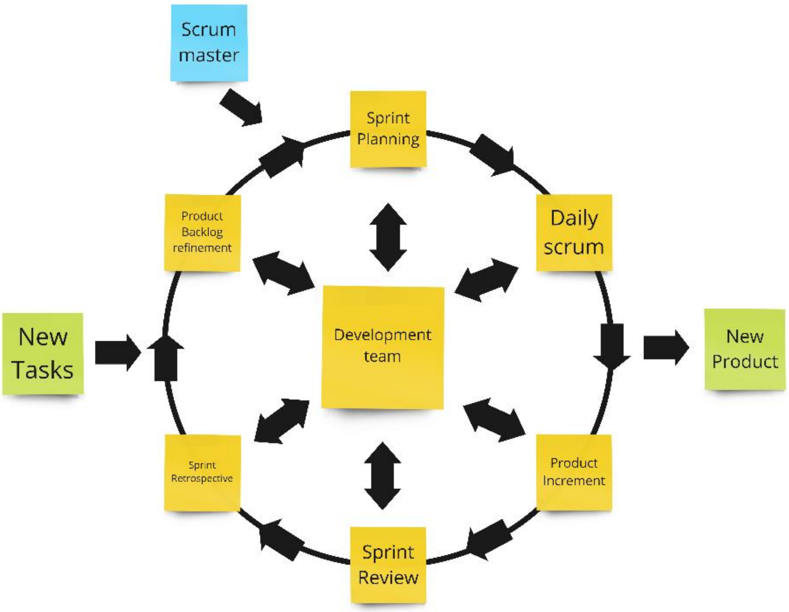

Figure 2.1: *The iterative nature of Scrum. Own illustration based on [13]*

One major challenge in software development is the tendency for projects to grow excessively large and become unmanageable. Described in Figure 2.1 the Scrum method introduces a cyclical model that reduces tasks into better manageable sizes [13]. This approach has proven beneficial for developers. It allows them to concentrate on a specific, smaller piece of the whole project at a time. This also makes their work more manageable [6]. It provides them with a concrete goal to work towards, which helps support motivation even in larger projects. The model that Scrum introduces is scalable up to an organizational level.

2.2 Scaled Scrum

The original Scrum guide is designed and intended for a single development team, but since modern-day software projects are growing in complexity and team count, there is a demand for an agile model for larger organizations [2]. The Scrum implementation used throughout the whole organization is called Scaled Scrum. There are multiple existing scaled Scrum frameworks. The literature recognizes Scrum of Scrums, Scrumconix, LeSS, Nexus, and SAFe as the main frameworks of Scaled Scrum in large organizations [10]. These scaled Scrum frameworks are extending the idea of the original Scrum guide by adding some rules and roles that make it easier to apply Scrum rules to software organizations of varied sizes and working cultures.

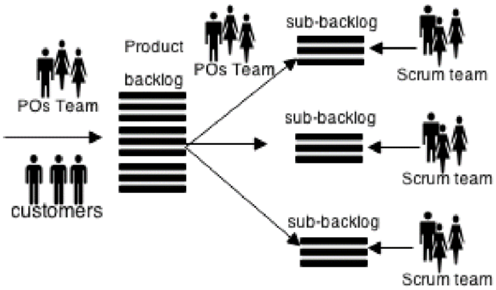

Figure 2.2: *Division of workloads according in Scrum of Scrums [1]*

Scrum of Scrums is one of these scaled Scrum guides. Scrum of Scrums extends the original idea of value creation from the team and developer level to the organizational level making the whole agile process run at every level of the organization [4]. Scrum of Scrums introduces several new roles and extends some older ones. One of the roles introduced is the Chief Product Owner (CPO), who delegates diminishing of some larger workloads to smaller ones. The teams have their Products Owners (POs) who take the smaller tasks and put them into team backlogs [1]. This is illustrated in the Figure 2.2 where the work is divided into team backlogs. This model makes the feedback loop reach the top of the organization. The model allows measuring the progress of given tasks at every level of the organization. One major drawback of this model is that it requires a lot of communication and coordination [1]. If those things are not working, the process will eventually break.

When implementing a scaled agile framework for an organization, it is necessary to consider that every organization is different. That is why existing ready-made Agile framework solutions are not available. Assuming that using only some parts will supply early payback will not get reliable results [4]. One good practice is to take some existing models as a building block and use it as a base model. After the selection of the base model, an organization may build its Agile model on top of the base model. However, implementing agile still requires much work and commitment to get the process working. For example, a software consulting company making full-stack applications has dissimilar needs from the agile process than a semiconductor manufacturer making firmware for their developed chips.

2.3 Embedded systems and agile

The development of embedded systems significantly diverges from traditional software development. Embedded systems are inherently more complex. Therefore, developing embedded solutions requires special knowledge in many fields [8]. This complexity increases the number of developers and teams involved in the process, thereby adding layers of complexity to the development process [5]. Consequently, there is an amplified need for development processes that effectively manage and streamline the development of these systems.

One of the main difficulties in implementing an agile process for embedded systems is that the design processes tend to be relatively slow for chips [8]. That causes difficulties in synchronizing the sprints with the firmware development. An added thing to consider with embedded applications is the testing. The testing process requires specialized testing infrastructure, software, and hardware. The testing process must be synchronized with the development, creating an additional challenge. These are the biggest challenges in the agile for embedded systems. As for all agile implementations, communication, and the processes around it, determine if an agile way of working will be functional in an embedded development organization [8].

2.4 Communication in agile

Communication is crucial for agile process to work. Communication is especially important because, everything from implementing the simplest component to some elevated level needs some level of communication to work. Communicating about needs and giving feedback is needed in order to be agile. At the team level several communication channels are needed when developers are integrating their separate work together. The most important channel of communication is face-to-face and real discussions. It is supported by several other communication tools such as code repositories, daily meetings, and smart boards for distant teams [15]. Challenges in communication can lead dysfunctional agile process since the heart of agile is communication.

2.4.1 Challenges in agile communication

Communication is fundamentally easier in smaller organizations. For large organizations, there are several challenges in communications. In large global organizations, the most common problems are typically related to time zones, project coordination and management, and organizational culture [16]. For international organizations usage of native languages can be a problem that prevents engagement of some people within the organization [14]. Remote work can cause difficulties in terms of communication. The challenges are mainly caused due to missing non-verbal communication and facial emotions in remote meetings [9].

Working across the globe is the main reason for having remote meetings. Remote meetings can cause difficulties and those may be mitigated by having the right tools and processes for communication in place. Another major challenge is related to organizational factors, such as communication tools. If an organization does not have any relevant tools and processes for communication, the communication process may become dysfunctional. That may lead to project delays and failures [16]. If an organization has the right tools and employees use them as intended, that usually leads to the success of the agile process itself.

2.4.2 Communication tools for agile organizations

Many large organizations have adopted social networking tools to make communications easier by doing business in multiple countries. A few examples of these tools are Slack, Teams, and Viva Engage. In addition to these tools communication processes in agile organizations may be managed through Kanban boards like Jira, Trello, or Miro. Software development teams usually have a version control system. These version control systems may have some communication tools integrated as well to help communication around the code. Using social networking tools enhances communication between team members when using group chats over one-to-one messaging [14]. That helps the communication process and prevents the same questions from being asked again.

Almost all open-source applications are developed and maintained by volunteers. These groups do not have similar hierarchical software organizations compared to companies, but they have developed processes where face-to-face communication is not necessary. These groups have adopted several different communication channels such as Internet Relay Chats (IRCs) to support their communication [11].

As stated above, communication tools are helping to mitigate communication-related problems in teams. Successful usage of communication tools can lead better team experience for the employees [9]. It is still important not to import communication tools and processes to solve problems not related to communication but also to making sure that the tools and processes are working as intended. This requires from employees to know how to use the tools and have a similar understanding of how to use them.

2.4.3 Communication with customer

While communication between team members is important, so is the communication between the customer and the developing team. It is generally known that it is customer’s responsibility to make good and clear requirements. The customer is expected to be part of the process and support the developing team [12]. The customer’s responsibility is to be committed to the process. The team is responsible for consulting the customer often enough to get the maximum output and ideas from the customer.

If the customer is not very committed to the project the risk of failures and misunderstandings increases since the developing team does not have access to the information, it needs [7]. The agile framework necessitates customers out there for the development team to answer their questions and needs. Feedback is expected from customers during the process and lacking it may be one of the reasons for project failures [16]. The key to have a successful project requires fair communication from all participants.

3. AGILE TRANSFORMATION PROCESS

As a background information to the agile transformation process, the target company u-blox is a Swiss-based semiconductor company founded in 1997. The number of employees was approximately 1300 personnel in 2022 [18]. The company operates in 16 countries across the globe. The primary focuses of u-blox are cellular, short-range, and positioning solutions and the services related to these fields. The main market focuses are consumer, automotive, and industrial markets [17]. Recently, u-blox's positioning product center underwent a significant organizational change. While the positioning has been moving away from the traditional corporate line structure, it has adopted a more agile and process-oriented value organization. This notable change resulted in the creation of a new DevOps Value Unit, which is the central focus of this research.

This chapter explains the target organization. Section 3.1 is about the background of the change. In Section 3.2 explains how the company has been organized after the agile transformation process. Section 3.3 describes the focus of this thesis: DevOps Value Unit, and its functionalities and provided services. It has graphs about the main client organizations of the DevOps Value Unit. Section 3.4 illustrates how the request processes should work.

3.1 Motives for the transformation

Before the summer of 2023, the u-blox positioning product center used a traditional corporate line organization. In this line of organization, different smaller development teams were working as part of some larger teams. The larger teams were composed around related topics, like testing and testing support functionalities. This line organization did not follow agile principles on the organizational level. Communication between employees was going through several communication channels. Direct messaging was a common method to reach a co-worker when there was a need for something. Larger tasks and goals came from the management to the team specialized in the topics. The developing team implemented the wanted features and functionalities according to the requests of the superiors.

The transformation was done to improve the feedback loops and to gather data about the performance of the teams. These changes were meant to improve the productivity of the organization in general. The model also equips the company organization to work better as the company grows.

The transformation to a value-based organizational structure altered this significantly. After the transformation process, feedback loops and task monitoring processes are now in use. At the same time, a process for managing tickets and defining new levels of tickets was introduced. Along with the value organization transformation process, several new agile organization-related roles were introduced. The goal of this transformation was to improve the firmware and hardware development process and to make them more transparent for the higher-level management.

3.2 Towards value-based organization

The organizational reform has led to a transition from a line organization to a value-based organizational structure. The model of this new agile implementation is based on Scrum of Scrums. The new organizational structure is adopted by positioning organization that includes various Value organizations. Each of these organizations consists of multiple Value Units. *Value Units* usually tend to have some common topic they are working around. Value Units typically consist of value teams. Value teams are where developers conduct their daily work. This structure emphasizes the importance of value creation at every level, from the overall organization to the individual developer level.

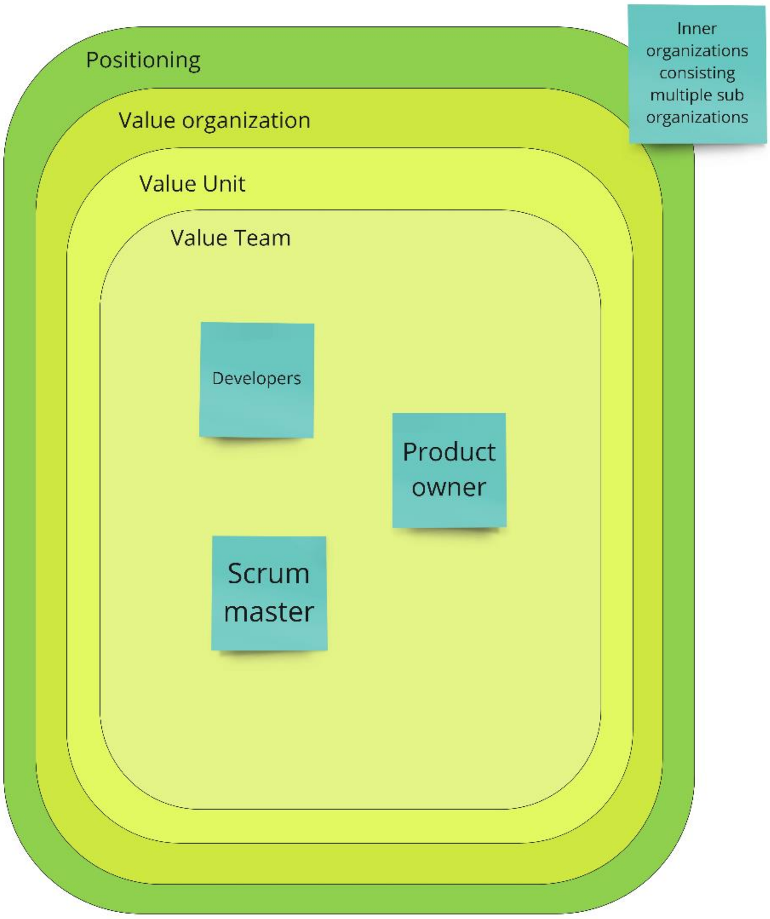

Figure 3.1: *The new positioning value organization and its structure*

Figure 3.1 presents the organizational hierarchy at u-blox positioning. Value Organizations make strategies and direct large-scale issues in alignment with their mission and vision. Value Units create value in specific areas, contributing to the larger goals set by their parent organization. These Value Units consist of value teams that are working around related topics. In these value teams, developers collaborate to solve individual problems and create solutions. Effectively piece by piece turning strategy into real value. Thus, every level of the organization has a vital role in delivering value for the company.

3.3 DevOps Value Unit

Due to the big organizational change, new value teams were formed, and roles of old teams altered significantly. A new DevOps Value Unit was formed from existing supporting teams to provide development support to whole whole Positioning product center. The new DevOps Value Unit consist of 3 new value teams that were picked from the old-line organization based on the services and products they provided.

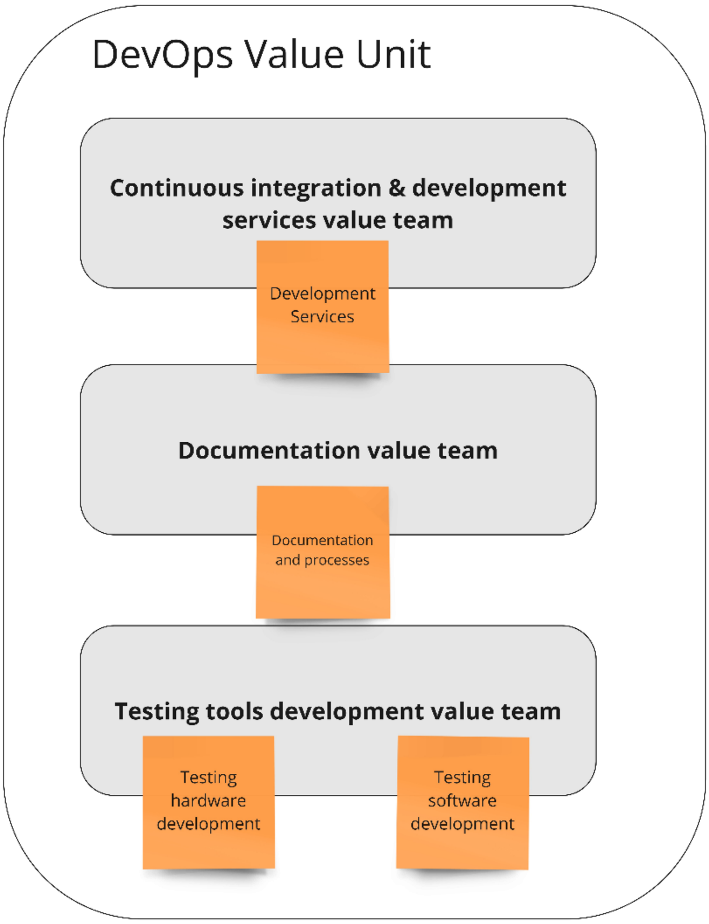
DevOps Value Unit
Continuous integration & development services value team
Development Services
Documentation value team
Documentation and processes
Testing tools development value team
Testing hardware development
Testing software development

Figure 3.2: *DevOps Value Unit, its value teams, and their provided services*

The reorganization has altered the communication dynamics between teams, leading to changes in service request processes from other teams. This change raises a need for process and service requests analysis from DevOps Value Unit point of view to better fulfil needs of other Value Units. Figure 3.2 presents services DevOps Value Unit is supplying to positioning. This analysis aims to recognize challenges in current agile implementations in DevOps Value Unit and investigates service requests and expectations towards it.

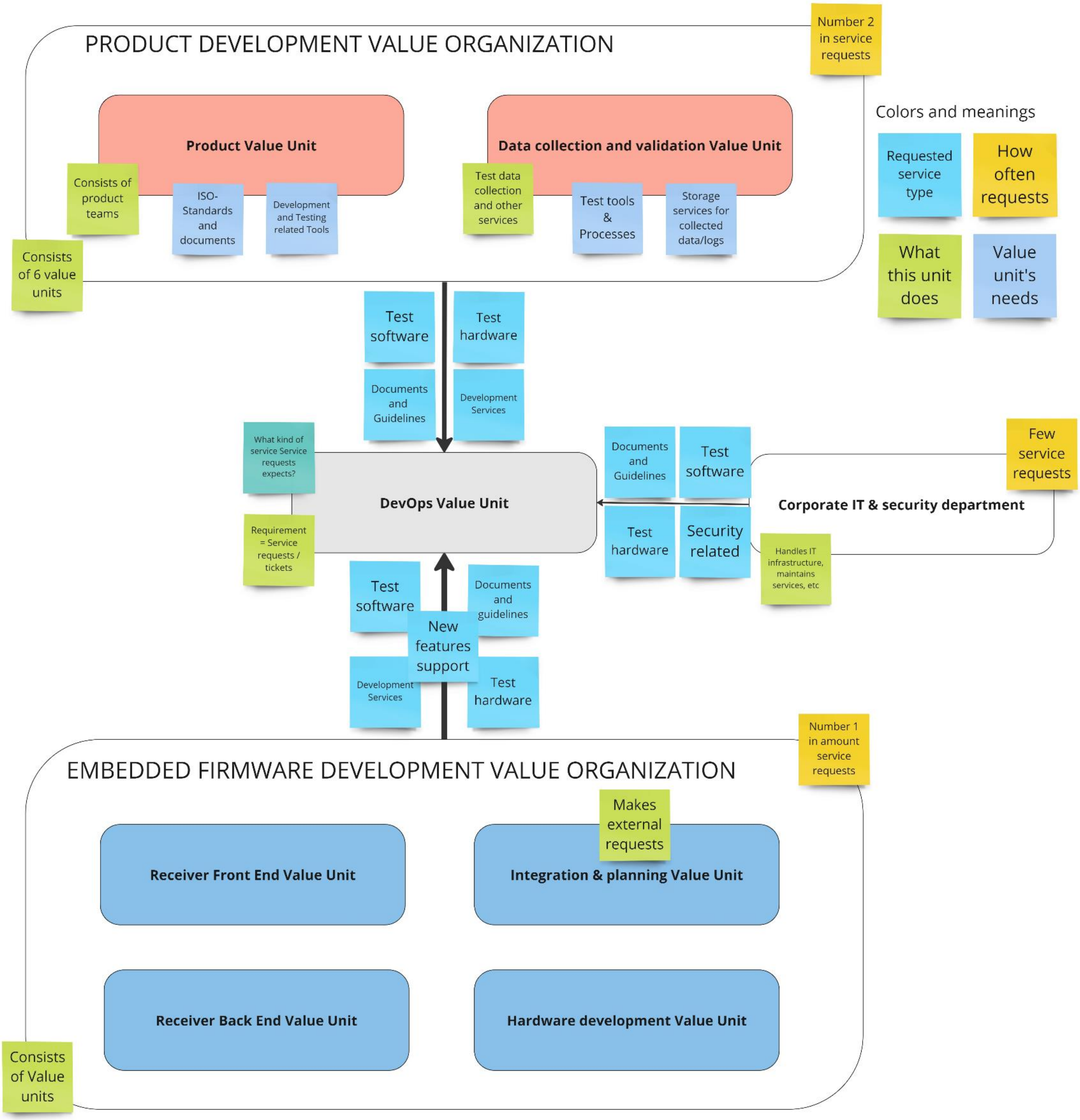
PRODUCT DEVELOPMENT VALUE ORGANIZATION
Number 2 in service requests
Product Value Unit
Data collection and validation Value Unit
Consists of product teams
ISO-Standards and documents
Development and Testing related Tools
Test data collection and other services
Test tools & Processes
Storage services for collected data/logs
Consists of 6 value units
Colors and meanings
Requested service type
How often requests
What this unit does
Value unit's needs
Test software
Test hardware
Documents and Guidelines
Development Services
What kind of service Service requests expects?
DevOps Value Unit
Requirement = Service requests / tickets
Documents and Guidelines
Test software
Test hardware
Security related
Few service requests
Corporate IT & security department
Handles IT infrastructure, maintains services, etc
Test software
Documents and guidelines
New features support
Development Services
Test hardware
Number 1 in amount service requests
EMBEDDED FIRMWARE DEVELOPMENT VALUE ORGANIZATION
Makes external requests
Receiver Front End Value Unit
Integration & planning Value Unit
Receiver Back End Value Unit
Hardware development Value Unit
Consists of Value units

Figure 3.3: *Service requests towards DevOps Value Unit*

As seen in Figure 3.3, the DevOps Value Unit supports numerous other Value Units within the positioning organization. The primary client is the Embedded firmware development value organization, which consists of various Value Units and teams. This organization develops positioning platforms, needing that both software and hardware testing keep pace with the product development process.

The Product Development Value Organization has requirements from the customer's product perspective. The product Value Unit's support requests are often aligned with several ISO-standards that need to be fulfilled. Data collection and validation Value Unit requires testing for software compatibility and Development services for their operations. u-blox IT department has several requests towards DevOps Value Unit. These requests are typically aligned with security standards and therefore these are meant for all teams. All recognized Value Units need documentation services from the DevOps Value Unit. Their documentation needs are used to share knowledge about current services and ongoing processes.

3.4 Service request flow towards DevOps Value Unit

u-blox's new agile model requirements and decisions have a certain hierarchy. According to this model, larger Deliverables are being sliced into smaller pieces. At the end of this process developer will take over some very small sub-tasks that have been sliced multiple times from bigger goals. At the same time, there is the feedback process running in reverse order. Below is presented a graph of how the initiatives are being processed into smaller pieces.

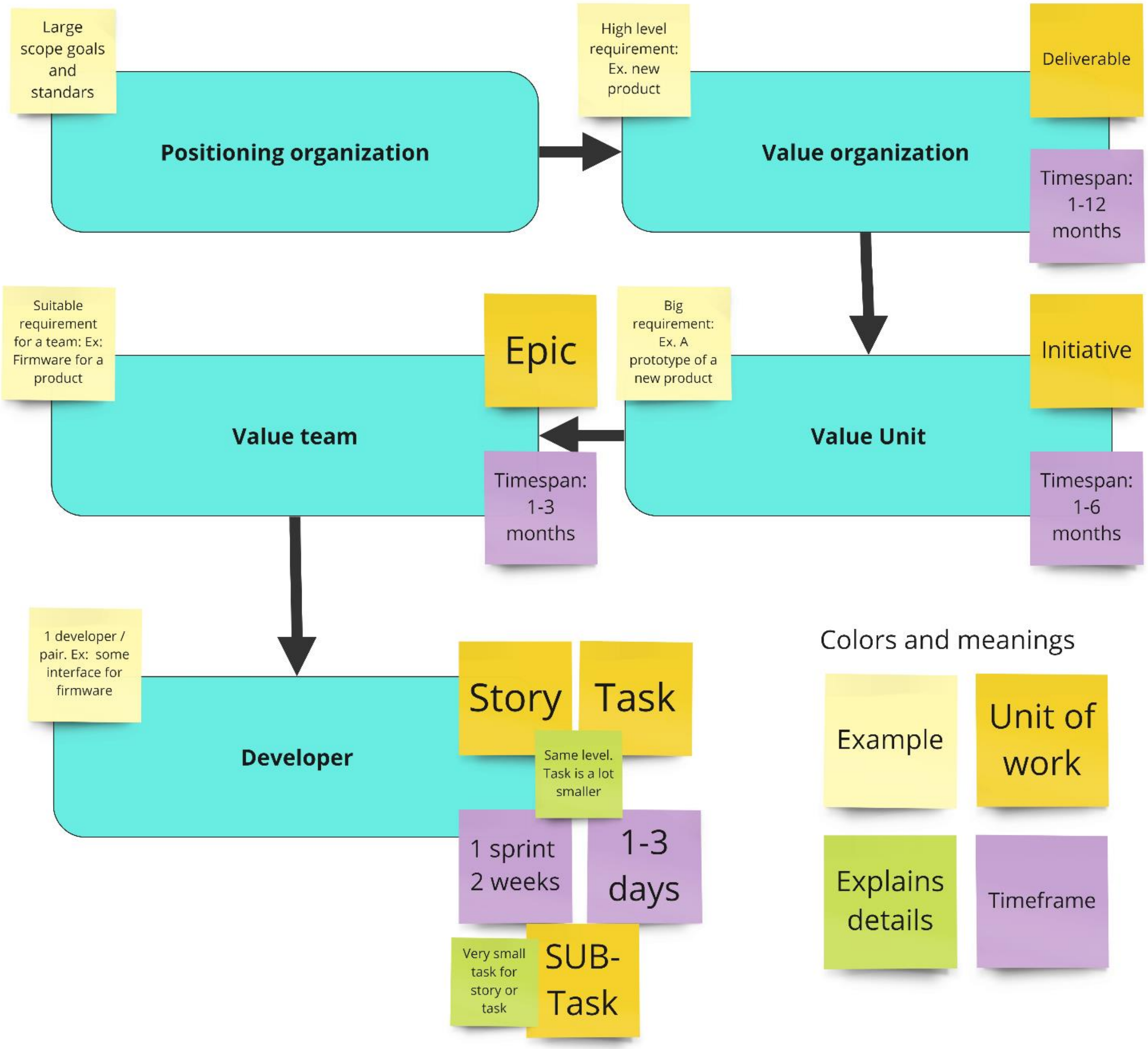
Large scope goals and standars
Positioning organization
High level requirement: Ex. new product
Value organization
Deliverable
Timespan: 1-12 months
Suitable requirement for a team: Ex: Firmware for a product
Value team
Epic
Timespan: 1-3 months
Big requirement: Ex. A prototype of a new product
Value Unit
Initiative
Timespan: 1-6 months
1 developer / pair. Ex: some interface for firmware
Developer
Story
Task
Same level. Task is a lot smaller
1 sprint 2 weeks
1-3 days
Very small task for story or task
SUB-Task
Colors and meanings
Example
Unit of work
Explains details
Timeframe

***Figure 3.4:** Positioning task levels in each organization*

Figure 3.4 presents the typical very high-level flow of how the request process works. It can be seen from the figure that typically the positioning organization sets certain high-level projects, these are typically new product releases for example. These product releases can be reduced into deliverables, which are smaller than whole products, but still excessively big and high level. *Deliverables* can be broken down into initiatives. *Initiatives* are typically something that can be integrated with other initiatives to make a deliverable. An example of an initiative could be a user story that describes one thing on a high level that the user can do. Initiatives are broken into epics, which are typically sized for a team. *Epics* are logical steps toward initiatives, and they can hold technical details. An example of an Epic could be the first release of a small software product. Epics are broken into stories or tasks. *Stories* are typically larger than tasks and more descriptive, as *tasks* are addressing a single minor problem. They both can hold subtasks that are the smallest units of work. When it comes to the DevOps Value Unit it only takes in initiatives, as it is not part of any value organization.

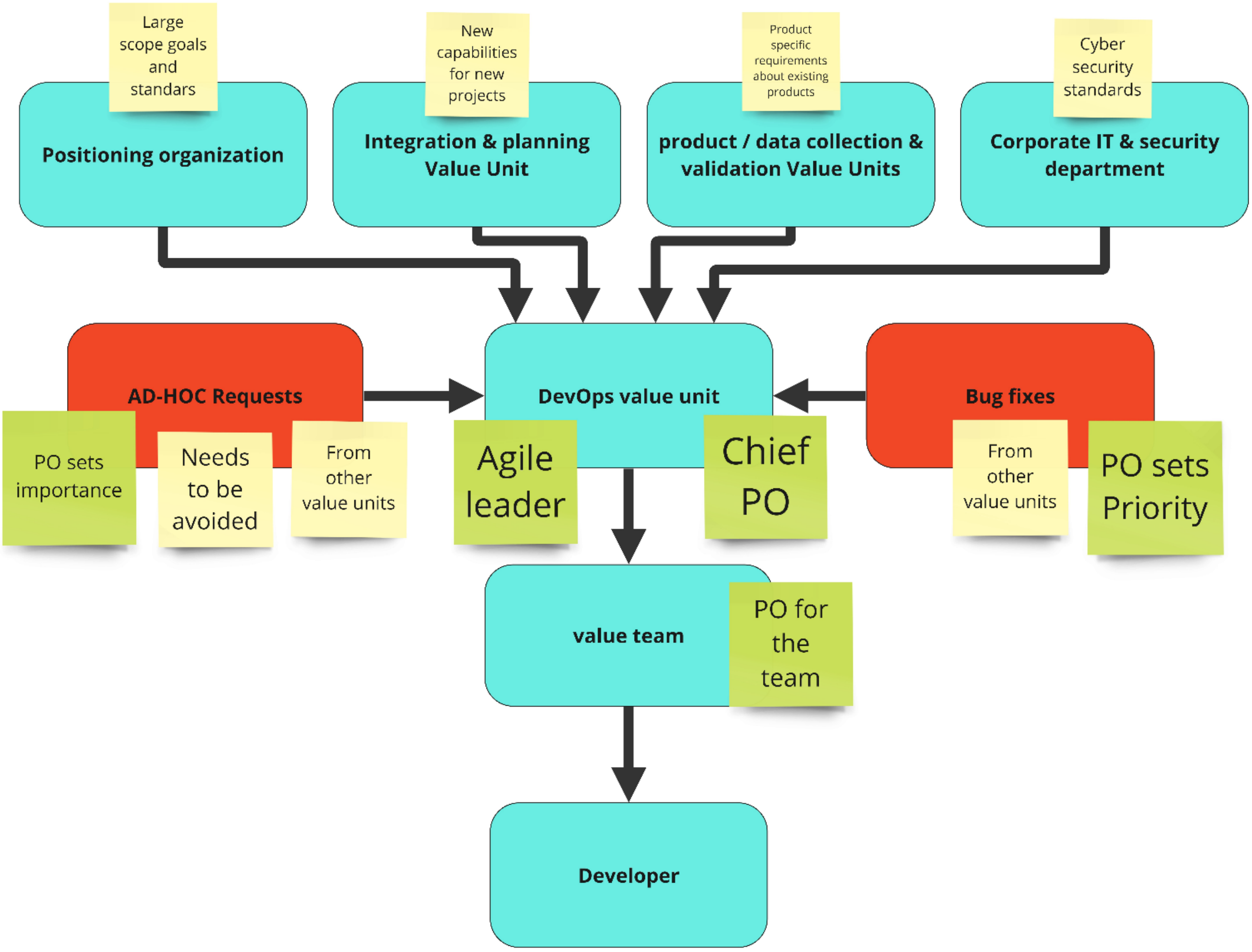

Figure 3.5: *DevOps Value Unit's initiatives and requests from outside*

Figure 3.5 Describes requests that the DevOps Value Unit receives. This picture describes the process as it is described in agile planning. The ideal situation would be that all requests to teams would go through POs. This would make sure that the team is focused on the most important tasks. Ideally, all the requests would come to the Value Unit level where POs can prioritize and delegate workloads to the right teams. This model would minimize incoming ad-hoc requests, thereby reducing the workload that developers are facing. One additional benefit is that the model keeps the developers focused when there is a need to focus on only one task at a time. It would also help to distribute resources more effectively, leading to better working organization.

4. AGILE PROCESS AT THE DEVOPS VALUE UNIT

This chapter contains interviews of u-blox employees collected approximately 5 months after the change. The interviews have been collected in one-on-one meetings with selected DevOps Value Unit's employees from all teams. The interviews were conducted in English and Finnish. The Finnish interview contents and quotes are being translated for this thesis. The questions used in the interview can be found at Appendix A. The questions aim to find out where the requests and requirements come from, and do they follow agile practices. The other question topics are related to communication tools in use at u-blox. The last section in question bank is related to current agile implementation at u-blox.

This chapter is divided into sections based on the themes described above. Section 4.1 is about the background of the interviewees. Section 4.2 describes how the interviewees feel about the current request process. Section 4.3 contains answers from communication related tools. Section 4.4 is about agile processes and how the interviewees see the meaning of agile and its roles. Last Section 4.5 concludes other topics and thoughts based on the interviews.

4.1 Background of the interviewees

All the interviewees were from the DevOps Value Unit. The number of interviewees was seven in total. The interviewees came from the different teams inside the DevOps Value Unit. Interviewees had been working relatively long in the IT industry, and the shortest working background was approximately 8 years of experience. The average working history in years was around 14 years' worth of experience. Employment time at u-blox was from 1 year up to 10 years.

All the interviewees had utilized at least some Agile model before joining u-blox, meaning they all had had some experience with other Agile models than the current one at u-blox. Used Agile models varied between different implementations of the scrum, kanban, and waterfall models.

The roles of interviewees in their teams varied a lot. All the Scrum Master roles in teams were labeled as “Acting” Scrum masters since the Scrum masters did not obtain any official Scrum master training. Typically, they were mainly working as developers in their relative teams. The other roles present were managerial, product owner, and developer roles.

4.2 Service request processes and customer interaction

According to the process described in Chapter 2.3, requests typically follow predefined procedures. As for the real-world development processes, it gets more complicated. Usually, the development tasks to developers go through the organization and PO before the developers take over the development tasks in the sprint planning ceremony. Based on the data collected from the employees, this process does not always work as intended. Some small requests go through other channels like Teams, even though they should go through Jira. Typically, these tasks are small improvement needs or urgent bug fixes. Sometimes developers put these ad-hoc in the backlog, but there are cases where these tasks are being worked on without putting them in the backlog or notifying the PO.

Good characteristics of tickets and requests typically consist of use cases instead of very niche technical ready-made solutions. One developer described a well-constructed request as “I would say poor requirements are very specific technical implementation details. Good ones are just use cases.” Generally, good requests have some description of the background of the working environment and what is the wanted solution in the end. Currently, some of the requested topics are very technical solutions, even though the developers usually know the working environment better and can produce the best solution to the problem.

Typically, at the DevOps Value Units' development processes, the client is not present when the team takes over a new task. However, this happens in cases where the ticket created by the customer does not have all the necessary information or if the background of the request is unclear. In those cases, the developing team typically contacts the customer for more information on the requested topic. At the developer level, the developers are not usually engaging in many discussions with the customers regularly. The product owner and managerial levels typically have more customer interaction since the higher-level initiatives are being discussed at that level.

The request breakdown is typically managed internally within the team, and therefore the client is not present in this phase. The team knows the problem and forms the logical steps itself. The breakdown phase is typically easier to do in the teams working with software, but the testing hardware team has more difficulties with task breakdown since they cannot deliver a part of their product that would increase value. This process is still continuously improved. In addition to team level breakdown meetings at the Value Unit level there is also some task breakdown done. This happens in DevOps Value Unit's backlog refinement meetings where PO's along with agile leader are going through tasks and setting priority order for tasks before putting them to developing team's backlogs.

4.3 Communication and Collaboration tools

Currently, u-blox has multiple tools and communication channels in use. These channels consist of ticket management channels like Jira, communication tools such as Teams, and version control system git. These tools are in use by various organizations throughout the whole positioning organization. Other Microsoft provided services are also in use for communication. These include Viva Engage, SharePoint, and others. U-blox has internally developed and hosted a wiki service as the old knowledge base for storing information. All this has caused the data and information to be divided into many separate places.

The interviewees were all stating that the current tools were not good or bad. Most employees felt that the communication tools available did support their work. The employees were experiencing difficulties with some communication tools in use. These difficulties are mainly related to two communication tools: Teams and Jira.

The difficulties related to Teams were mostly about the channel functionality Teams offers. Employees felt that the channels were dysfunctional and that not all the teams used them similarly. Some of the development teams used channels as the primary way of communication and as forums for different topics. Some other development teams did not use them at all. Teams channels were said to have ineffective search functionalities. Due to these issues in search, a developer commented: “I have the people sending direct messages me with common problems that have been answered in the Teams channels, but they can't find that thread because of the poor functionality of Teams.” Some of these problems originate from a lack of education and from people using the same functionalities for different purposes.

The issues identified in Jira originated from a large set of tools and functionalities it offers. The considerable number of functionalities results in complex menus. For managing basic tickets from the developer's point of view this complexity can be problematic. One developer commented on Jira’s search functionality: “The search is working, but not. In some cases, I was unable to find the ticket that I was looking for.” However, the problems with Jira are not recognized by the managerial and PO level since they have used more

Jira’s complicated user interface, and they have more experience using its features than the developers.

A couple of interviewees felt that adopting Slack to replace the usage of Teams channels would help to communicate better. Most of interviewees thought that there were enough tools in use but the education and processes on how to use these tools were insufficient. The interviewees also stated that the company having offices in multiple locations can cause some issues due to the lack of face-to-face communication. The lack of education and usage guidelines has led to situations where the same tools have different usages in different contexts.

4.4 Agile roles and the meaning of agile

This section oversees agile-related questions. Section 4.4.1 is related to the scrum-related roles and their functionalities. Section 4.4.2 contains responses about the essence of agile itself. The other topic is whether the interviewees have felt u-blox's agile implementation useful for their team.

4.4.1 Agile roles

This section looks at whether the employees feel they are aware of u-blox’s defined role descriptions. These descriptions describe the responsibilities of each role existing in u-blox’s agile implementation. Nearly all the interviewees felt that they knew what the role descriptions at u-blox are. One interviewee was not familiar with the role descriptions but wanted more education about the roles. Few other responders felt also that the roles are not very defined and there is some possibility of misunderstandings due to the employees having different thoughts about the roles. The interview focused primarily on two main roles the developers work with. These roles are Scrum Master and PO.

The scrum master role was seen as more problematic than the PO role. As stated before, the DevOps Value Unit does not have any scrum masters with proper training. All these current scrum masters are labeled as “acting” scrum masters. This leads to a situation where the employees have different expectations about the scrum master role. Currently, the developers see the scrum master role’s responsibilities are facilitating discussions and organizing meetings. Additionally, the scrum masters themselves said that “understanding what the team is working on and resolving the conflicts and blockers that they're having.” The scrum masters feel that their job is to protect their team and guarantee that their team is not being disturbed during the sprints.

For the PO role, the interviewees saw it as well functioning. The PO role was seen from the developer's point of view as a gatekeeper for incoming requests. The key responsibilities of the PO were stated to be backlog grooming, task prioritization, and the process maintainer along with the scrum master. The only difficulties identified by the people developing testing hardware at the beginning since their way of working is so hardware intensive, and that was causing some minor difficulties with their PO but otherwise, the role has been seen as well-functioning and meaningful. The POs have received PO training about their roles and responsibilities to better enable them to do their work.

4.4.2 The meaning of agile

The agile model of u-blox divides opinions. Some of the interviewees see it as well functioning and enabling model that improves current processes and clarifies the rules. Other interviewees see it as a blocker for their work. The key issue with the current model is that it is not very versatile. Complaints arose from these topics: the sprint length is fixed, the development teams do not have enough autonomy, and the hardware-related tasks are hard to break down. The general opinion about the current model is that it is a good basic model working well enough despite its problems.

People tend to have their own understanding of what agile means. The one question that was asked was about the meaning of agile. For some of the interviewees agile meant “Ability to react to changes and gather feedback on what works and what does not work” and for some others “Products to be delivered with quality and in time.” Generally, it can be said that the meaning of agile was somewhat different for everybody. The common themes around the topics are related usually to the collaboration aspect of working. The employees' varying thoughts about the meaning of agile are likely rising from a lack of discussion about the meaning of agile.

To further improve employees understanding about agile and the roles in use they were asked if they saw any kind of education fitting to further improve their own and co-workers understanding about agile or agile roles. Proper training for the scrum masters was mentioned a couple of times and it was seed the most important training. The interviewees also wanted more education on Scaled Scrum and some general discussions about agile and the meaning of agile.

4.5 Observations about communication

During the interview process, additional observations were made about various topics. Some of the interviewees did not have a similar understanding of the agile topics even though they worked in the same team. This is probably caused by a lack of discussions around these topics

The need for education was raised in conversations a couple of times. The issues with Jira could be fixed with education on how it should be used at the developer level. It would probably also be useful to have discussions about the agile processes at u-blox. This would help to get everybody on the same page about the meaning of agile.

5. RESULTS

The agile model in use at u-blox is based on the scaled implementation of Scrum. This implementation is the base of u-blox's model called Scrum of Scrums. It extends the idea of the original Scrum guide to the whole organizational level [4]. This section compares the results of the interview from Section 4 and the organization working model from Section 3 with the theories from the literature that can be found in Section 2. The comparisons are organized similarly to the results of interviews. This chapter answers the research questions presented in Chapter 1.

Section 5.1 compares u-blox's model with theoretical Scrum of scrums and identifies agile processes in organization using Scrum. The section also investigates issues in the current process and whether the current process follows good practices. It also answers to the first research question: *How u-blox's agile transformation process is working for the DevOps Value Unit?* Section 5.2 compares agile communication tools and processes with the existing ones and tries to identify challenges in communication and answers to the second research question: *How does communication work inside the DevOps Value Unit?* Section 5.3 explains what is agile based on the interviews and literature. It also provides answer to the third research question: *What is the meaning of agile at the DevOps Value Unit?*

5.1 RQ1: How u-blox's agile transformation process is working for the DevOps Value Unit?

The current service requests towards DevOps come from the other Value Units and organizations. Many of the service requests come from the stakeholders presented in Figure 3.3. The new service requests go to the DevOps Value Unit backlog. After that, the priority of the DevOps backlog is defined in the Value Unit-level backlog refinement meetings. Based on the predefined priority in the DevOps backlog, the team POs pick their tasks from there and assign them to the team during sprint planning ceremonies.

Typically, the agile request handling process for developing teams is very similar to the original Scrum. According to the original Scrum guide, these requirement processes come from a stakeholder to the PO, who puts them into the developing team's backlog in priority order [13]. The process at the teams in DevOps is quite like this. The tasks for the team come through the PO. Typically, the tasks are assigned to developers during the sprint ceremonies. For a developer working in a development team the lack of knowledge about the Value Unit level processes is not an issue.

One issue comes from the habits of the old organization. Occasionally, the stakeholders of DevOps Value Unit put their tasks and bug reports straight into the development team backlog. This behavior violates the described scrum process and diminishes the POs' control over the backlog [13]. Second issue is that the development teams are doing some work and bug fixes outside the backlog. Luckily these ways of working are not very common but a reminder of the old organization.

One way to address these issues in the processes is having trainings about the agile. These trainings could be useful for the whole positioning organization. Alongside these trainings it would be useful to have discussions reflecting how the processes used to work in the past and how they are wanted to be in the future. The trainings could help to facilitate these discussions. After all it takes time for the new processes to establish well.

The current processes towards DevOps Value Unit work well at a high level. This higher-level management consists of agile Leaders, CPO, and POs. These are as well described in Scrum of Scrums [1]. At the DevOps Value Unit these employees have received training about their roles. DevOps refinement organization seems to work well in general. The higher-level management is easier due to fewer people working at the higher level in the organization's hierarchy. This improves communication and knowledge sharing at this level.

5.2 RQ2: How does communication work inside the DevOps Value Unit?

Communication and knowledge sharing are necessary for any functional agile process. Working communication in an organization involves using different tools and processes around it [3]. These communication processes can be improved and developed further. It is known that large transformation processes do not always work very smoothly, but they tend to improve over time. The usage of communication tools can be challenging if there are no agreed guidelines on how to use these tools. In the end, the goal of these tools and processes is to make communication so easy that it leads to more knowledge being shared.

5.2.1 Communication tools and processes

The communication processes can be supported with tools. These tools can be used to support different types of communication including task-related communication, messaging, and video calls. These tools can support communication in software organizations and thereby help the agile processes to work better [14]. Additionally agile requires communication processes alongside the tools. These processes are meant to enable communication by offering knowledge of how to communicate and with whom.

As u-blox is a medium-sized company operating multiple locations across the globe there are some challenges in communication. Different teams use different platforms for communication. The development teams might work in multiple locations and therefore cannot have daily organic discussions. Due to the number of nationalities present, there are different customs about how to communicate. These challenges are typical challenges for agile organizations to face while having operations in multiple countries and cultures [15]. Some of these issues arose from the discussions with employees. Further information can be found in Section 4.3. Source [15] identifies face-to-face communication as important from the agile point of view. Face-to-face communication was also mentioned during the interviews where the interviewees identified it as an important forum for daily communication.

5.2.2 Communication issues

The issues identified during the interview process were not very severe. A few minor issues were identified with different tools. The most problematic tool was Teams and especially Teams channels. The Teams channels were identified as dysfunctional features. The channels caused some problems for some of the developers. The problems with Teams channels at the DevOps Value Unit were similar to the source [14]'s descriptions with slack problems. Co-workers were using direct messages over the channels. That makes the channels even more dysfunctional and, in the end, leads to growing dissatisfaction over Teams.

Additionally, there are some issues with requirements managing service Jira at the developer level. These issues were mostly caused by the complexity of the tool. These issues could be addressed by having some Jira training about the functionalities at the developer level. The developers might not need all the features Jira offers in their daily work, but these potential pieces of training could give them a better understanding of the tools and thereby increase their motivation towards their work.

To address the issues in communication tools and processes it is important to talk about them. It is important to ask if one does not know something and ask for guidance. Additionally, it could help to have discussions about the tools that the team uses and discuss the motivation and processes with different tools. These things usually seem obvious and useless, but they can increase employee happiness when the employees have a better understanding and visibility of the processes.

5.3 RQ3: What is the meaning of agile at the DevOps Value Unit?

Agile has many meanings and a unique meaning for everybody. Employees come from different backgrounds, and they have used various implementations of agile before. The Scrum model is so versatile that it exists in multiple variants in many software-developing organizations. Due to its popularity, the employees tend to have some idea of what it means. Employees' perception of agile usually originates from their earlier experiences of various implementations of agile models. This leads to situations where employees may have different opinions and thoughts about the meaning of agile. At every implementation of agile there is something that works and something that still needs to be fixed.

5.3.1 Success stories

At the DevOps Value Unit, the current agile implementation has proven to be a useful and working tool for the employees despite the challenges. Section 4.4.1 speaks of the PO role and points out that it works well according to POs and developers. The literature sees PO as a bridge between customers and the developing team [1]. PO's role at the DevOps works very similarly to this. This is probably one of the reasons why the PO role is felt to be so high functioning. Source [1] also describes the roles of the POs for the team and CPO roles that are in use at the DevOps Value Unit. Generally, task management works very well, and these processes are on a good path forward.

There were not too many complaints about the current agile model. There is still room for improvement in various topics. The current model was said to be well-functioning. The expectations of the employees matched with their knowledge of the model. Generally, it can be said that the transformation process has gone well for the DevOps Value Unit. There have not been any major challenges, though the transformations are never easy.

5.3.2 Challenges and Issues

Along with the transformation processes, a couple of challenges have been showing up. The testing hardware team had some challenges. This is caused due to their hardware-related tasks. These tasks can be difficult to split into very small pieces. This problem is also recognized by the source [8] that the development processes with hardware can be relatively slow. This issue was identified as a large problem at the beginning of the agile transformation process. The feelings about the agile itself are improving within the testing hardware team since they have gained more knowledge about the process. This is a good example of how the development processes can be improved over time. It has shown that these processes take time and hard work.

The Scrum master role at the DevOps Value Unit has had difficulties due to the lack of proper scrum master training. As mentioned in Sections 4.1 and 4.4.1 all the current Scrum masters of DevOps Value Unit are temporary. This means that they are handling their Scrum Master duties alongside their other tasks. It is recognized by the managerial level at the DevOps Value Unit that education is needed for the role. The Scrum Master education would better enable the Scrum Master role’s full potential as a coach and mentor for the developing team.

5.3.3 What are the employee's thoughts about the agile?

Due to the employees having multiple opinions about what agile means, it is necessary to discuss what agile means at the team level. These discussions could handle topics such as the role of the Scrum master in the team or how the developers see u-blox’s agile model working for their team. It could be useful to have similar discussions at the DevOps Value Unit. The goal of these discussions would be to gain a better understanding of the thoughts of other employees. Discussions like these have not been a common custom. Therefore, having them could further increase the understanding of the u-blox’s agile model.

According to literature agile and Scrum mean many things. The original Scrum guide states that an agile implementation consists of these core values: commitment, focus, openness, respect, and courage. These base principles mean that the development teams and the customer are committed to the process and achieving goals together. The team is focused on the task, and they are open and communicate if they have issues. Respect means that the developers treat each other as professionals. Courage means that the team members dare to work through problems. [13]

The interviewees stated in Section 4.4.2 that the meaning of agile and Scrum is the ability to react to changes and gather feedback. These values are in line with the original Scrum guide’s ideas about the agility of the process. Additionally, one of the interviewees said that agile is related to the ability to deliver products that are high quality and delivery is on time. This is also one core value in Scrum. Generally, the agile values of interviewees and the literature are similar. agile means many things and it has unique meaning for everyone.

6. CONCLUSIONS

This thesis researched agile in an embedded developing organization. The main methods of this research are literature reviews and interviews of the employees of the target organization. This data was compared and based on the observations following results were constructed.

In conclusion, it can be said that the current agile implementation at u-blox is working well for the DevOps Value Unit. The current process is heavily based on Scrum of Scrums implementation. This similarity enabled comparison between the models. The current agile process at u-blox is well constructed. The process contains many elements from the Scrum of Scrums and therefore these processes are almost identical in terms of roles and ideas. The process seems to be working for the DevOps Value Unit despite the few challenges the current process has.

The key challenges identified in this thesis are mainly related to communication and the usage of communication tools. One issue is that some of the tools are not easy to use and that the employees use the same tools differently depending on the team. Another problem was that occasionally new tasks are put into a developing team's backlog without consulting the PO. Some of the work is still being done without the backlog. That is still a minor issue to be solved. A need for education for the Scrum Master role was mentioned, by multiple employees during the interviews. The current process will improve over time if there are open discussions about the process and problems.

In this thesis, there are a few limitations that constrain the generalizability of the conclusions. The research was conducted in one company and in one Value Unit which limits generalizability a lot. Another limiting factor is the relatively small number of interviewees.

Further research in the context of this thesis could be to escalate the research into other Value Units and the whole u-blox organization. This could provide more insight into the processes and perceptions of agile at u-blox. The next step would be taking some similar embedded developing organizations and making comparisons between different companies and their agile implementations

One of the main conclusions was that the meaning of agile is in the end defined by the team using the agile model. The only thing to consider is that everyone is on the same page when they talk about agile implementations. The process itself can change and improve since doing so is agile, but only as long as everyone is kept on board with these changes. In the end, in the world of information technology, the only constant is change.

REFERENCES

[1] AlMutairi, A. M., & Qureshi, M. R. J. (2015). The Proposal of Scaling the Roles in Scrum of Scrums for Distributed Large Projects. International Journal of Information Technology and Computer Science, 7(8), 68–74. https://doi.org/10.5815/ijitcs.2015.08.10

[2] Philipp Diebold, Anna Schmitt, and Sven Theobald. (2018). Scaling agile: how to select the most appropriate framework. In Proceedings of the 19th International Conference on Agile Software Development: Companion (XP '18). Association for Computing Machinery, New York, NY, USA, Article 7, 1–4. https://doi.org/10.1145/3234152.3234177

[3] Dühring, L., & Zerfass, A. (2021). The Triple Role of Communications in agile Organizations. International Journal of Strategic Communication, 15(2), 93–112. https://doi.org/10.1080/1553118X.2021.1887875

[4] Ebert, C., & Paasivaara, M. (2017). Scaling agile. IEEE Software, 34(6), 98–103. https://doi.org/10.1109/MS.2017.4121226

[5] Kaisti, M., Rantala, V., Mujunen, T., Hyrynsalmi, S., Könnölä, K., Mäkilä, T., & Lehtonen, T. (2013). agile methods for embedded systems development - a literature review and a mapping study. EURASIP Journal on Embedded Systems, 2013(1), 15–15. https://doi.org/10.1186/1687-3963-2013-15

[6] Kautz, K., Johanson, T. H., & Uldahl, A. (2014). The Perceived Impact of the agile Development and Project Management Method Scrum on Information Systems and Software Development Productivity. AJIS. Australasian Journal of Information Systems, 18(3). https://doi.org/10.3127/ajis.v18i3.1095

[7] Korkala, M., Pikkarainen, M., & Conboy, K. (n.d.). Distributed agile Development: A Case Study of Customer Communication Challenges. In agile Processes in Software Engineering and Extreme Programming (Vol. 31, pp. 161–167). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-01853-4_21

[8] Könnölä, K., Suomi, S., Mäkilä, T., Jokela, T., Rantala, V., & Lehtonen, T. (2016). agile methods in embedded system development: Multiple-case study of three industrial cases. The Journal of Systems and Software, 118, 134–150. https://doi.org/10.1016/j.jss.2016.05.001

[9] Lisa L. Lacher and Cydnee Biehl. (2019) Does Social Sensitivity Impact Virtual Teams? In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19). Association for Computing Machinery, New York, NY, USA, 36–42. https://doi.org/10.1145/3287324.3287409

[10] Pardo Calvache, C. J., Chilito Gomez, P. R., Viveros Meneses, D. E., & Pino Correa, F. J. (2019). Scrum+: A scaled Scrum for the agile global software development project management with multiple models. Revista Facultad de Ingeniería, 93, 105–116. https://doi.org/10.17533//udea.redin.20190519

[11] Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. Empirical

Software Engineering : an International Journal, 13(3), 303–337. https://doi.org/10.1007/s10664-008-9065-9

[12] Ramesh, B., Cao, L., & Baskerville, R. (2010). agile requirements engineering practices and challenges: an empirical study. Information Systems Journal (Oxford, England), 20(5), 449–480. https://doi.org/10.1111/j.1365-2575.2007.00259.x

[13] Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Scrumguides.org. Retrieved September 5, 2023, from https://scrumguides.org/docs/scrum-guide/v2020/2020-Scrum-Guide-US.pdf#zoom=100

[14] Viktoria Stray, Nils Brede Moe, and Mehdi Noroozi. (2019). Slack me if you can! using enterprise social networking tools in virtual agile teams. In Proceedings of the 14th International Conference on Global Software Engineering (ICGSE '19). IEEE Press, 101–111. https://doi.org/10.1109/ICGSE.2019.00031

[15] Yagüe, A., Garbajosa, J., Díaz, J., & González, E. (2016). An exploratory study in communication in agile Global Software Development. Computer Standards and Interfaces, 48, 184–197. https://doi.org/10.1016/j.csi.2016.06.002

[16] Sofiia Yermolaieva. (2020.) Communication Challenges in agile Teams from The Communication Theory Prospective. In Proceedings of the 2020 European Symposium on Software Engineering (ESSE '20). Association for Computing Machinery, New York, NY, USA, 88–95. https://doi.org/10.1145/3393822.3432327

[17] u-blox annual report. (2022). Retrieved October 31, 2023, from https://content.u-blox.com/sites/default/files/documents/Annual-Report-2022.pdf

[18] u-blox. (2023). Home. u-blox. Retrieved September 13, 2023, from https://www.u-blox.com/

APPENDIX A: QUESTIONS USED IN THE INTERVIEW

Questions:

Background questions:

- How many years of experience do you have in this industry?
- Have you utilized any Agile methodologies other than the one used at u-blox?
- Describe shortly what is your role in the team?

Interview questions:

Requests/Tickets:

I. Who makes requests towards you/your team?
II. What is a good requirement? What kind of information is included?
III. With whom you will talk about requests you get? Are there any end users present when the request is made?
IV. What kind of discussions are being held with the person who made the request? Meeting once, regularly, or never?
V. If the request is big enough, how it is being broken down and is the requesting party present in this?

Communication related tools:

VI. Do the u-blox tools support your work? Jira, git teams for ex.
    a. If not what kind of problems, you have faced/seen?
VII. Do you have any tools or practices in mind that would support your communication with co-workers?

Agile at U-Blox:

VIII. Are you familiar with u-blox's agile role descriptions?
    a. If yes, do you see any differences how they are implemented in practice?
IX. How do you think u-blox's agile model fits in your team?
X. How do you see the scrum master role in your team?
XI. How do you see PO's role in your team?
XII. What is agile? Describe with your own words or what does it mean for you?
XIII. What kind of education would be suitable to get better understanding of the agile and its processes?