CHALMERS
UNIVERSITY OF TECHNOLOGY

Feasibility analysis of Scaled Agile Framework for a high-tech company

Master's thesis in the Master's Programme International Project Management

PENPISUT POOWANAVIROCH
SONNATAS CHAISORN

Department of Architecture and Civil Engineering
Division of Construction Management
CHALMERS UNIVERSITY OF TECHNOLOGY
Master's Thesis ACEX30-18-62
Gothenburg, Sweden 2018

MASTER’S THESIS ACEX30-18-62

Feasibility analysis of Scaled Agile Framework for a high-tech company

Master’s Thesis in the Master’s Programme International Project Management

PENPISUT POOWANAVIROCH
SONNATAS CHAISORN

Department of Architecture and Civil Engineering
Division of Construction Management
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2018

Feasibility analysis of Scaled Agile Framework for a high-tech company

Master's Thesis in the Master's Programme International Project Management

PENPISUT POOWANAVIROCH

SONNATAS CHAISORN

Examensarbete ACEX30-18-62
Institutionen för arkitektur och samhällsbyggnadsteknik
Chalmers tekniska högskola, 2018

Department of Architecture and Civil Engineering
Division of Construction Management
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
PURI, R. (2015) Bottom up vs. Top down: How to scali agile. [Online image] Available from: https://thenextweb.com/insider/2015/09/22/bottom-up-vs-top-down-how-to-scale-agile/ [Accessed 12/06/18].
Department of Architecture and Civil Engineering
Göteborg, Sweden, 2018

Feasibility analysis of Scaled Agile Framework for a high-tech company

Master's thesis in the Master's Programme International Project Management

PENPISUT POOWANAVIROCH

SONNATAS CHAISORN

Department of Architecture and Civil Engineering
Division of Construction Management
Chalmers University of Technology

ABSTRACT

Agile project management has recently become very popular within software development industry and now a lot of large organization tend to adopt agile in the same trend. While Agile itself is not meant for the large organization. Then, large-scaled Agile has been introduced in order to overcome the problem. Our aim for this study is to analyze the feasibility to apply, one of the most famous, Large-Scaled Agile Framework (SAFe) to one of a big high-tech company which already went through the Agile transformation. We conducted our research by using the interview as our main method of data collection. We aim to gather data from people from a different role and different level within the organization in order to get the diversity of information. Then we conducted data analysis to identify their current challenges with their Agile. We found out that SAFe is a very well defined framework consists of many useful concepts. SAFe also gave us a better understanding and enlighten us on how to scale up Agile within a large organization. Lastly, our finding indicated that nowadays large organization with agile methodology are still having trouble with applying Agile with high-management level.

Key words: Project Management, Non-Traditional Project Management, Agile,
Scaled Agile Framework, Large-Scaled Agile

Contents

Preface

After two years of study our Master degree at Chalmers University of Technology. This Master thesis is our last chapter and brings us to the end of our student life. Without many supports, we could not make this far.

First of all, we would like to thanks the Avancez, IPOET (International Programme Office for Education and Training), and Adlerbert Foreign Student Hospitality Foundation scholarships that support us throughout these valuable two years.

Second, we would like to thank our case study company, our company supervisors (Annika Rosendahl and Grace Wang), and all the interviewees for participating in the study.

Third, we would like to thank Professor Petra Bosch for being the best supervisor for our master thesis.

Forth, we would like to thank all professors that have been teaching us during these two years throughout our master study.

Fifth, we would like to thank all of our IPM's friends that fulfill our master study life with all the fun, joy, and crazy time together. Without all of you studying at Chalmers would not be the same.

Sixth, we would like to thank Thitipong Sansanayuth, Juntima Nawilaijaroen, and all of the Thai friends in Gothenburg that support and help us during these two years.

Lastly, we would love to thank our families for supporting us from the first day that we decided to study and travel 8420 kilometers away from home until the end of our Master degree.

Göteborg June 2018

Penpisut Poowanaviroch

Sonnatas Chaisorn

1 Introduction

In this section, first we explain the background of our master thesis. Second, we present aim and research questions. Third, we describe the delimitation of our study. Lastly, the structure of the paper is described.

1.1 Background

Due to high competition in the software development market, more pressure is placed on organizations to deliver software faster, be more flexible due to changes of client needs, and have high quality at the same time (Paasivaara et al., 2018). Otherwise, these organizations will become extinct, no matter their size, intelligence, or strength (Leffingwell 2015). To adopt Agile development methods have become more and more popular in software development organizations around the world, in both small and large organizations (VersionOne, 2016). The Agile method is known for empowering the employees, allow for change to serve uncertainty in clients need, focuses on business value, focuses on users, enables reducing lead times, improves quality, and helps organizations adapt faster (Livermore 2008; Petersen and Wohlin 2010). In addition, Agile software development is flexible and fast due to the iteration development and has fast and close feedback from clients (Stettina and Hörz, 2015).

The Agile software development method is originally developed for small teams and projects which does not fit for large organizations. However, the trend of adopting Agile in large organizations has increased significantly during the recent years (Leffingwell 2007; Scheerer, Hildenbrand, and Kude 2014). Referring to the largest relapse survey on Agile transformation, the State of Agile Survey (VersionOne, 2016), 43% of the self-selected respondents worked in development organizations for more than 50% of teams using Agile, and 62% of 4000 respondents came from an organization with over hundred people in software development industry (Paasivaara, 2017). Even if the survey is not scientific, it illustrates that a large number of big organizations that use Agile as their method to manage their projects. However, transforming a large organization to Agile development is challenging (Paasivaara et al., 2013). Dikert et al. (2016) presented a systematic literature review, which showed the lack of research studies on how to conduct a successful Agile transformation in large organizations. The results from the systematic literature review showed that only six scientific papers studied large-scale Agile transformation, with almost 90% of papers were experience reports written by specialists (Paasivaara et al., 2018).

Large software development organizations often have a significant number of big projects, and most of them are globally distributed where they need to coordinate and collaborate between different development sites. Scaling an Agile project method is not easy, and many challenges arise due to collaboration and communication between Agile teams, as well as the challenge of projects distributed over different time zones and culture (Leffingwell, 2007).

Many Agile specialists introduced scaling Agile frameworks to manage Agile in a large organization such as Scaled Agile Framework (SAFe) (Leffingwell, 2018), Large-scale Scrum (LeSS) (Larman and Vodde, 2018), and Disciplined Agile Delivery (DAD) (Ambler, 2012). All of them have great success stories presented on their web pages. However, these scaling frameworks are still lacking empirical scientific studies on how to be implemented, what kind of challenges are present, and how to overcome those challenges. As reported by the State of Agile Survey (VersionOne, 2016), the most popular scaling framework among these three is the Scaled Agile Framework (SAFe). In this case, our study focuses mainly on how large-scale organizations work with large-scale Agile methods (e.g., SAFe).

1.2 Aim and Research questions

The aim of this paper is to start to fill the gap of large-scale Agile research and literature. By studying the possibility of how scaled Agile works with scaling the Agile framework and answer the following question;

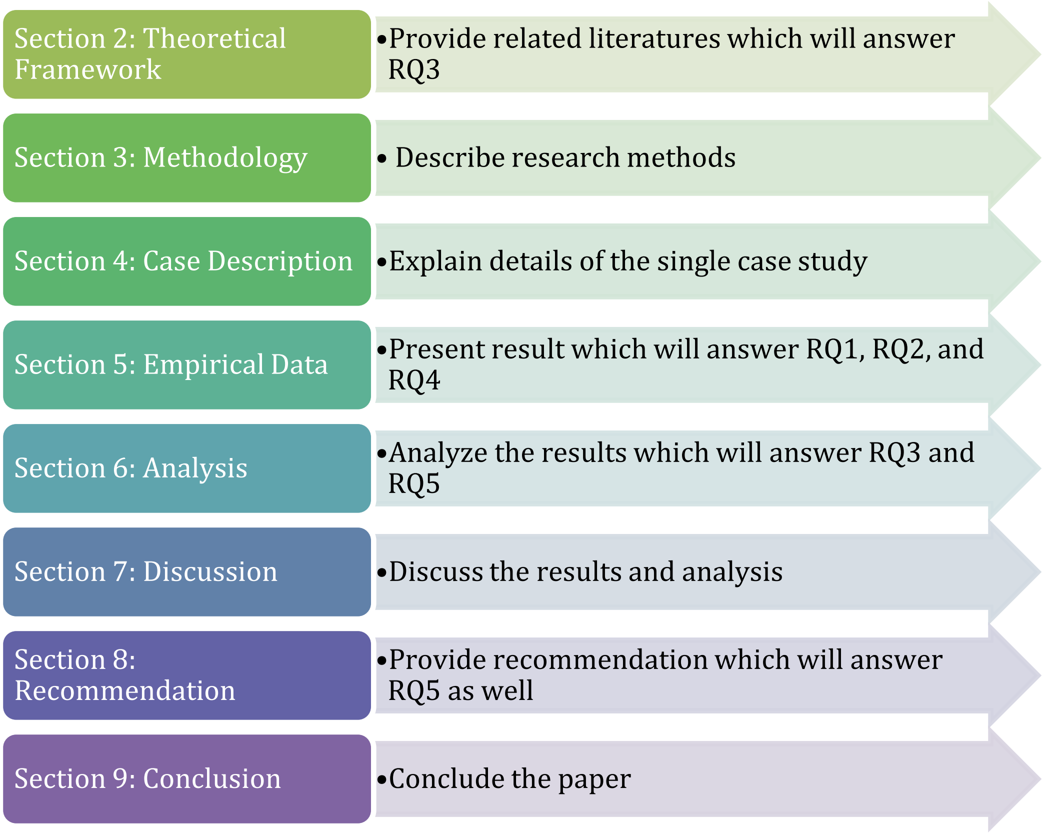
- RQ1: How are Agile approaches currently used in large software development organization?
- RQ2: What are the challenges in current agile approaches?
- RQ3: How can scaling Agile framework support current Agile approaches?
- RQ4: What are the challenges and impact by applying scaling Agile framework in software development organizations?
- RQ5: How could such a new framework be implemented in a current Agile organization?

1.3 Delimitations of the study

However, this research paper has some limitations due to a single case study based on one organization. The focus of this paper mainly studies the environment and current situation in the case study R&D department in a high technology company which we will call company “AiTea” throughout this paper. In addition, the topic of large-scale Agile is still lacking empirical studies to prove the success on how to implement Agile in larger organizations.

1.4 Structure of paper

This paper is structured into ten different sections which illustrate in Figure 1 below.



Section 2: Theoretical Framework	•Provide related literatures which will answer RQ3
Section 3: Methodology	• Describe research methods
Section 4: Case Description	•Explain details of the single case study
Section 5: Empirical Data	•Present result which will answer RQ1, RQ2, and RQ4
Section 6: Analysis	•Analyze the results which will answer RQ3 and RQ5
Section 7: Discussion	•Discuss the results and analysis
Section 8: Recommendation	•Provide recommendation which will answer RQ5 as well
Section 9: Conclusion	•Conclude the paper

Figure 1:Structure of paper

2 Theoretical Framework

In this section, we discuss related previous work and literature. First, we explain the Agile principle and Agile software development. Second, we explain the definition of large-scale Agile development. Third, we describe portfolio project management. Fourth, we explain scaling Agile software development frameworks. Lastly, we describe change management.

2.1 Project Management

Project management, according to the Project Management Institute (2017), is the application of knowledge, skills, tools, and techniques that the leader apply to the project in order to drive the project to meet the requirements and reach the outcome. While Wysocki (2014) added more business perspective and defined project management as a common-sense approach that gets client involvement to meet the needs and be able to deliver the expected business value. Project management is not something new that just happened during the last couple of decades but it has been used for hundreds of years. There is many evident and examples of outcome in the past e.g. Pyramid, Great Wall of China, Taj Mahal, etc. (PMI, 2017).

“One size does not fit all”, meaning that, there is no standard project management approach that works for every project (Wysocki, 2014). There are many project management approaches, the way to manage and plan the project, that was introduced after the 1950s. The traditional project management, or usually referred to as waterfall methodology, is the most well-known approach. This traditional method runs the project in sequential linear process (Wysocki, 2014). And then, there is the non-traditional project management which progresses in several iterations. One of the most famous non-traditional approaches is called Agile methodology. Agile methodology is commonly used in a more complex project where what is needed is not clearly defined (Wysocki, 2014). In this study, we will focus on the non-traditional project management, the Agile project management methodology.

2.2 Principle and Agile software development

Agile methodologies have become more popular and widespread during this last decade in software development organizations. Agile methods were evolved to overcome the weakness of traditional software development models (Hoda, Kruchten, and Marshall, 2013). Since organizations need to adapt themselves to move faster and be ready for high competition in software development market. Under these situations, organizational agility is a key success in the market. To achieve agility, organizations need Agile methods.

In 2001, Agile software development was first coded over “Agile Manifesto” by Martin Fowler and Jim Highsmith.

The Agile Manifesto (Fowler and Highsmith, 2001)

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Client collaboration** over contract negotiations
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more"

The Agile Manifesto has been the guideline foundation for all Agile project management (Wysocki, 2014). Furthermore, the Agile Manifesto defined Agile principles as empowered and persuaded software developers, depending on uncomplicated design and technical exquisiteness, create business value due to short iteration deliveries to clients (Dingsøyr, 2012). The core principles are a self-organizing team and to allow for a change in requirements at any phase of the development process (Wysocki, 2014). In addition, clients also play a big role in the development process by giving comments, feedback, or reflection that can lead to more satisfaction with the product outcome (Barbee, 2013).

Barbee (2013) identified three key features to Agile software development which are iterative, incremental and dedicated team. These key features are encouraged based on an Agile principle which is dominated by the Agile Manifesto (Fowler and Highsmith, 2001). First, **iterative** is a process that repeats periodically for short periods of time (e.g. every two or three weeks). This work period refers to iteration, in practice the work period can be one week, two weeks or even four weeks depend on each team or project. The length of the iteration can also depend on their business model or team's maturity of Agile. For example, teams new to Agile should probably start with longer iteration and then shorten down as their experience grow. Hence, there is no ideal iteration length. During these iterations, the project team works on accomplishing the results while the clients get a short-term impression of where the product is going instead of having to wait a long time for the final result. Second, **incremental** is to develop one piece or feature of work at a time. In the next iteration, the team will work on another piece of work. At the end of each iteration, a small part of the product is delivered to the client in order to show and get fast feedback from clients. Third, **dedicated team** (development team) means all team members work full time and focus on one project at a time. Figure 2 shows the concept of Agile key features.

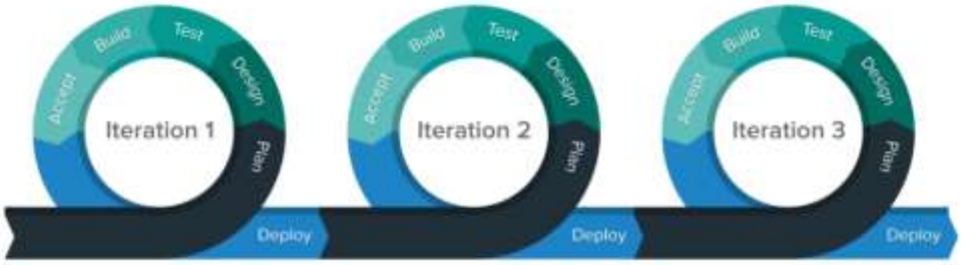


Figure 2: Concept of Agile key features (Gitscrum, 2018)

According to Dikert et al. (2016), the most popular Agile methods nowadays are Scrum, Kanban, and Extreme Programming (XP). **Scrum** is a method that highlights a project management perspective of Agile software development (Schwaber and Beedle, 2002). Scrum is a lightweight framework with iterative development and operated by using backlog (McKenna, 2016). While **Kanban** is a method which focuses on having the right work done at the right time and aims to improve the current working process by using Kanban board. In addition, Kanban tends to reduce waste in every step (Lei et al., 2017). For example, developers do not develop unnecessary features and do not implement more than they can test. In comparison, one big difference between Scrum and Kanban is the Scrum's backlog and the Kanban board. The backlog will promote time-boxed sprints and the way that work is pulled in batch while Kanban board, on the other hand, promote continuous delivery and work is pulled into pieces. The **XP** method is a set of processes that permissive efficient incremental development (Beck, 1999). Many of Agile software development implementation tend to mix these methods in some way (Fitzgerald et al., 2006).

When transforming to an Agile way of working, there is a significant change in role and responsibilities. In the traditional way of working, decisions are made from top-down or hierarchical. However, an Agile way of working tends to get rid of hierarchy and focuses more on uniting people to work together (Moreira, 2013). Since Scrum is widely used in many organizations, this paper aligns the Agile roles with Scrum roles. In Scrum refers to a group of people who are dedicated and work together to deliver client value that they committed within a sprint as **Scrum Team** (Moreira, 2013). A scrum team includes three roles which are Scrum master, product owner, and development team. The **Scrum Master** is the one that acts as a coach or servant leader to ensure that everything is understood and implemented correctly by the Scrum team. In addition, this role also plays a key role in order to enable an Agile mindset to the Scrum team (Moreira, 2013). The **Product Owner** (PO) acts as the voice of clients. The PO must understand and be able to prioritize client needs or requirements which are also called a product backlog. To create a meaningful user story (a short and simple description from customers perspective for a new feature) to product backlog for the development team to work with (Moreira, 2013). The **Development Team** is a cross-functional team of engineers which can include programmers, testers, software architectures, and technical writers. Such teams are considered as self-organized and empowered with cross-functional skills and collective decision making. Normally the team size is around five to nine members depending on each team (Moreira, 2013). Besides the members of a Scrum team, there is another important role known as an Agile coach. The **Agile Coach** is the key element when organizations decide to adopt Agile. He or she will be the one that helps organizations adopt Agile practices and mindset by provided training and initial knowledge to the teams. Normally, team members can easily go back to the old habit of the traditional way of working. An Agile coach needs to ensure that the team understand the change and set their mind to an Agile way of working (Moreira, 2013).

From above it seems that Agile methodologies have a lot of advantage due to fast delivery, self-organizing, allows for change, and works closely with clients (Wysocki, 2014 and Barbee, 2013). However, Agile methodologies still have some limitations. According to Ashish et al. (2016), these limitations are: lack of upfront planning, lack of sufficient documents, too many meeting and the requirement of training. Many authors also stated that Agile was developed for small teams and organizations and is not suited to large organizations, due to there is no clear plan and agile embracing risks which large organization trying to avoid at all cost (Dybå and Dingsøyr, 2008; Scheerer, Hildenbrand, and Kude 2014). However, Agile could still be a chance of huge improvement for the large organization. So, many specialists tried to introduce a way to overcome this challenge and it is called *Large-Scaled Agile*. The next section will discuss more on the large-scaled Agile development and its challenges.

2.3 Large Scaled Agile Development

The origin of Agile development was developed for small teams and organizations with proof of success stories, but large-scale Agile illustrated many challenges (Dybå and Dingsøyr, 2008; Scheerer, Hildenbrand, and Kude 2014). The challenges are related partly to the size of organizations, the larger the organization gets the slower it is for the organization to change (Livermore, 2008). Adopting Agile usually demands the entire organization to change because the concept of Agile development does not work for individual uses (Misra et al., 2010).

Many research studies identified large-scale Agile in term of size in persons or teams, project budget, and project scale (Berger and Beynon-Daviss 2009; Paasivaara et al., 2014). According to Dikert et al. (2016), the definition of large-scale Agile is software development organizations with 50 or more people or at least six teams. Not only developers are counted as a team, but also roles like Scrum master, product owner, and Agile coach also counted as one part of the team.

Agile was developed based on close communication and collaboration within the team, where it proved to work well with small organizations (Hanssen et al., 2011). When it comes to large distributed organizations, many challenges occur due to the difficulty of communication and collaboration with other teams, departments, and sites. In addition, lacking guidance on how Agile teams should interact or manage in large environmental contexts is also considered as a major challenge (Paasivaara et al., 2018). To solve these challenges organizations may need to find a suitable way or practice to bring distributed sites to come closer by improving communication and collaboration (Holmstrom et al. 2006).

Many scaling Agile software development frameworks have been introduced by specialists in the anticipation to solve those challenges when adopting Agile in large organizations. Examples are Scaled Agile Framework (SAFe) (Leffingwell, 2018), Large-scale Scrum (LeSS) (Larman and Vodde, 2018), and Disciplined Agile Delivery (DAD) (Ambler, 2012). According to the State of Agile Survey (VersionOne 2016), the most popular framework is SAFe. However, all of these frameworks still lack practices on how to implement, what kind of situations they are working well, and what are challenges and successes of their usage (Paasivaara et al., 2018). This is affirmed in the fact that current research is still lacking behind on the topic of large-scale Agile.

2.4 Portfolio Project Management

In this recent decade, Agile project management methods caused a revolution in how projects are organized and executed (Abrahamsson et al., 2009; Dybå and Dingsøyr, 2008). Agile methods originally introduced in software development projects are nowadays gaining more attention in the general project management field.

Agile software development proved to be fast, flexible with feedback, include iteration reviews, and a close relationship with clients. Without close collaboration between the development team and its clients, Agile methods seem to lose much of their efficacy (Hoda et al., 2010; Stettina and Heijstek, 2011). To implement these advantages of Agile methods in large organizations, one possible way is to apply Agile methods together with project portfolio management (PPM) (Christoph and Jeannette, 2015). PPM is basically the centralized management of the processes used by high-level management to manage time, resources, skills, and budgets that required to complete the project. Since PPM helps organizations maximize portfolio's financial value, link organizational strategy to the portfolio, and balancing resources with different projects within the portfolio (Cooper et al., 1999; Martinsuo and Lehtonen, 2007). The key point is that portfolios give an opportunity to make large organizations more agility outside of small projects (Christoph and Jeannette, 2015).

In traditional project management, the Project Management Institute defined a portfolio as a component collection of programs, projects, or operations management as a group to achieve strategic objectives (PMI, 2013, pp. 3). The relationship between portfolios, programs, and projects is that a portfolio is a collection of sub-portfolios, programs, and projects in order to help and ensure that the work outcome meets strategic business objectives. Programs are clustered within a portfolio which includes sub-programs and projects that are organized in a coordinated form in support of the portfolio. Projects can either be within or outside of a program which is still considered as part of a portfolio (PMI, 2013). Portfolio management assures that the connection between programs and projects are established and resources are allocated according to organizations priorities (PMI, 2013). In addition, PPM was designed to deal with the collaboration and coordination of multiple projects which have the same goals and fighting for the same resources (Cooper et al., 1997).

Portfolio management is well formed in traditional project management, but it still not often taken up in Agile project management. Leffingwell (2007, 2010) introduced a framework called Scaled Agile Framework (SAFe) which applied Agile methods with portfolio management. This helps larger organizations compete with small organizations in an Agile way of working (Christoph and Jeannette, 2015). However, like mention earlier in this paper, the framework itself still lacking empirical study and evaluation.

2.5 Scaling Agile Software Development Frameworks

According to section 2.2, many scaling Agile software development frameworks have been introduced by specialists in hope to solve the mentioned challenges when adopting Agile in large organizations. This section will describe three frameworks, which are Scaled Agile Framework (SAFe), Large-scale Scrum (LeSS), and Disciplined Agile Delivery (DAD) in more detail.

2.5.1 Scaled Agile Framework (SAFe)

From the previous topic, the Scaled Agile Framework (SAFe) has been introduced to support large organizations to compete with small organizations by merging Agile and project portfolio management. In order to overcome the comfort zone of Agile, portfolio management gives an opportunity to make organizations more agility outside individual projects (larger teams) by involving high-management level into agile process (Christoph and Jeannette, 2015).

According to the State of Agile Survey (VersionOne 2016), SAFe is the most popular framework among these three. Dean Leffingwell introduced the first release of SAFe in 2011 by describing how large organizations or how the program or portfolio level management handle their requirements or processes together with Agile teams (Richard and Stefan, 2015). SAFe can be view publicly by visiting scaled Agile framework website (Leffingwell, 2018). Nowadays, SAFe introduces up to four levels which are: Portfolio Level, Large Solution Level, Program Level, and Team Level. (See Figure 332)

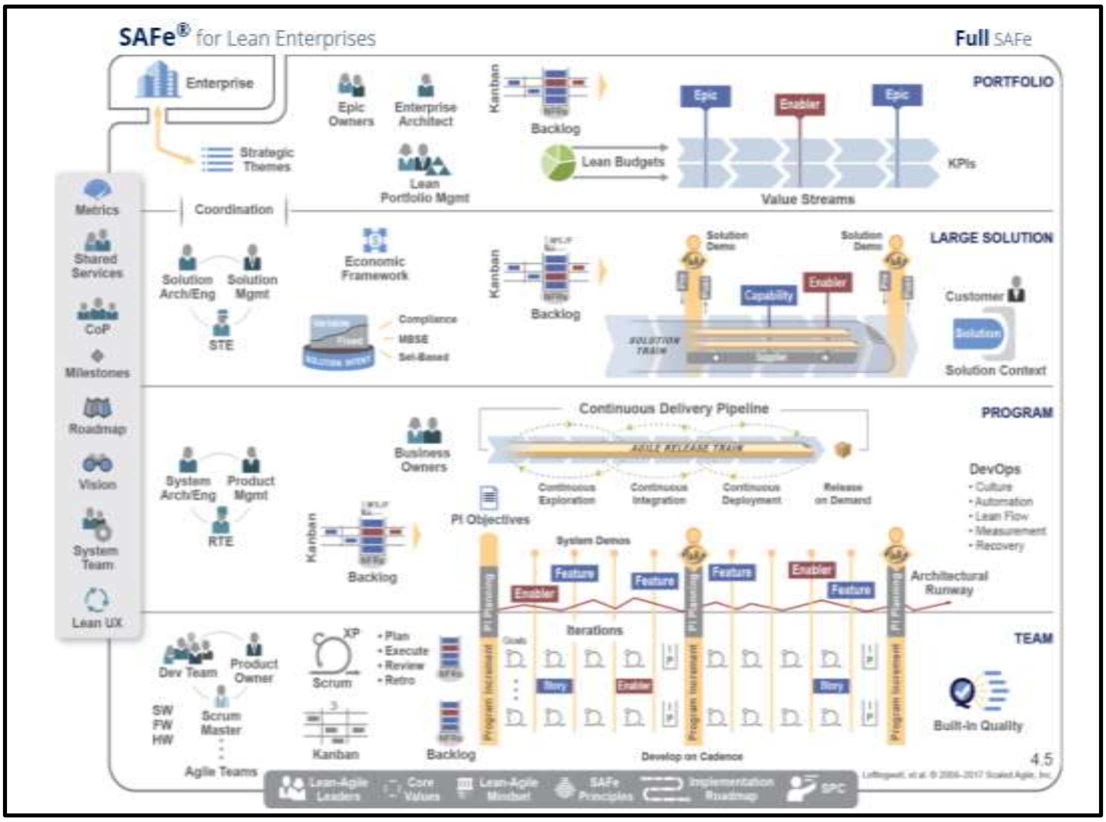


Figure 3: Scaled Agile Framework (SAFe) (Leffingwell, 2018)

Team level works mostly the same as traditional Agile methods by adopting Scrum, but Kanban is also possible (Maria, 2017). In addition to normal Scrum activities (sprint planning, sprint review, sprint retrospective), SAFe introduces the release planning meeting in order to update and keep every team on the same page every five iterations or sprints (Richard and Stefan, 2015). On the **Program level**, the framework introduces the Agile Release Train (ART) concept where 50-124 people work on the same value stream or release trains to connect the team level with the program level (Richard and Stefan, 2015). The ART supposes to help management, teams, and stakeholders to have a single vision. ARTs develop and maintain a Continuous Delivery Pipeline to keep release small features on a regular basis when completed. The **Large solution level** is optional for organizations that have multiple ARTs. To manage and deal with larger ARTs scale, SAFe introduced *Solution Trains* that help to find a workaround with a common solution for multiple ARTs (Leffingwell, 2018). The **Portfolio level** is the level on which planning on resources and budget are done, within what so call epics~~,~~ that identify large development initiative (Maria, 2017).

2.5.2 Large-scale Scrum (LeSS)

The Large-scale Scrum (LeSS) was introduced in 2005 by Craig Larman and Bas Vodde to help larger organizations to adopt Scrum for their development processes. LeSS can be seen as a regular Scrum that can adjust to being implemented at multiple levels in order to suit large-scale Agile development (Lerman and Vodde, 2018).

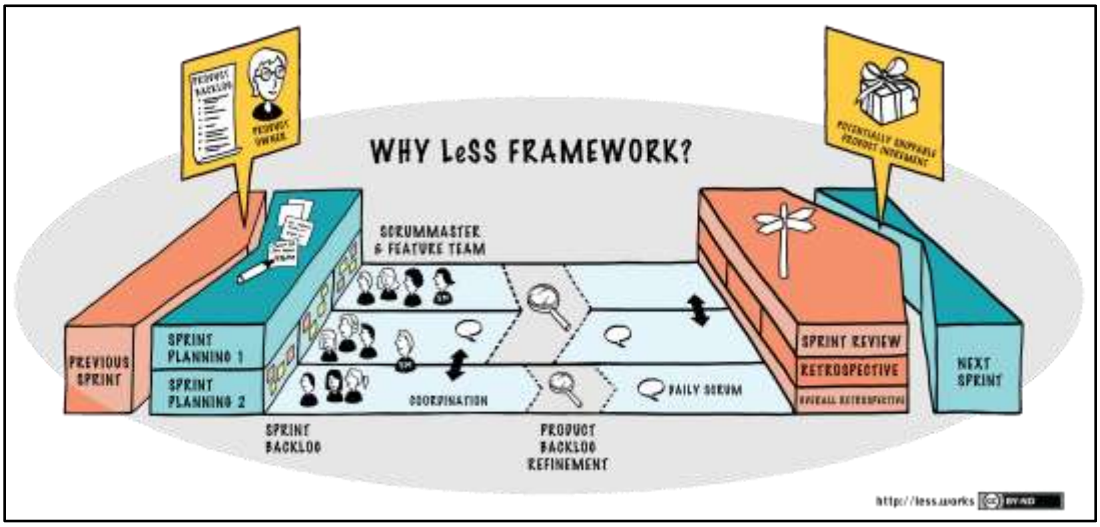


Figure 4: LeSS Framework (Larman and Vodde, 2018)

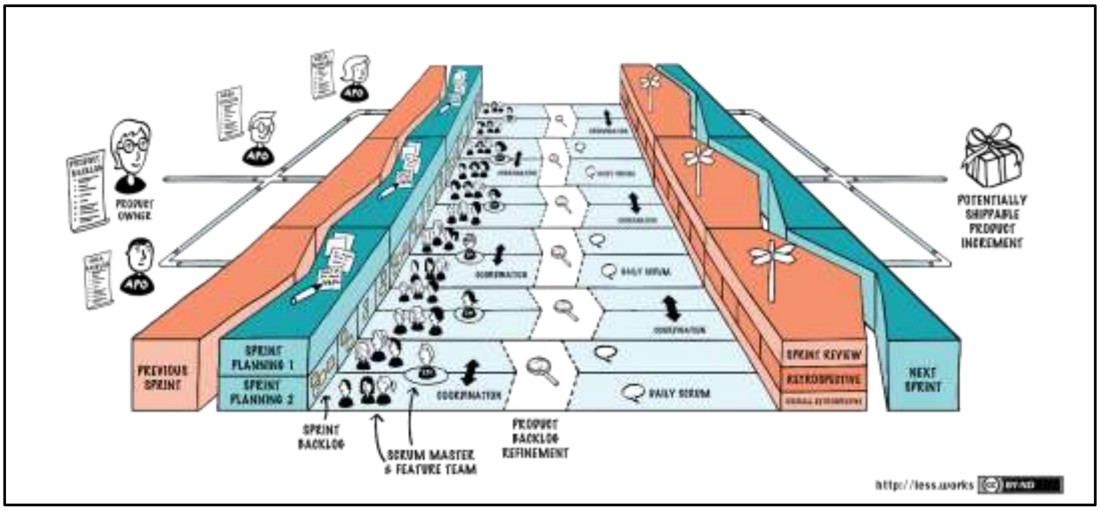


Figure 5: LeSS huge (Larman and Vodde, 2018)

LeSS has two frameworks which are LeSS and LeSS Huge (This paper will call LeSS as LeSS small in order to avoid confusion with Large-scale Scrum (LeSS)). The first one is **LeSS small** which is suitable for organizations that have two to eight teams that work under one PO in order to develop their product. (See Figure 4) Second, **LeSS huge** is suitable for organizations that have more than eight teams. Since only a single PO can no longer handle the entire product, this framework adds one more role, which is called Area POs (APOs) in order to enable scaling of product size. (See Figure 5) LeSS is considered as a great choice for the small organizations that are growing fast and looking for a framework to help scale up Scrum along with the organizational growth (Lerman and Vodde, 2018).

2.5.3 Disciplined Agile Delivery (DAD)

Ambler (2012) defined the Disciplined Agile Delivery (DAD) process framework as a people-first, learning-oriented hybrid Agile approach to IT solution delivery. DAD claimed to be scalable and goal driven. The framework was developed between 2009 to 2012 at IBM Rational. The aim of this framework is to fill in the gaps that Scrum ignores (Ambler, 2012).

The DAD process framework expands the Scrum development lifecycle to address the full delivery life cycle, at the same time adopting strategies from Agile and lean methods. DAD process framework is hybrid, meaning that it modified Scrum by adding other methods such as Kanban, Extreme Programming (XP), and Agile Data (AD). The DAD is a root of Agile database practices (Ambler, 2012). DAD divides develop process into three phases which are Inception, Construction, and Transition. The **Inception** phase, sometimes mention as initiation phase or iteration zero, aims to analyze the business issues, establishing technical solutions, planning approach, setting team and environment. The propose of **Construction** phase is to develop a solution that gives business value to organizations to maintain the cost of deployment. The **Transition** phase is mainly focusing on training, preparing for actual deployment, and transitioning knowledge to another team who will take care of production system. (See Figure 6 and Figure 7)

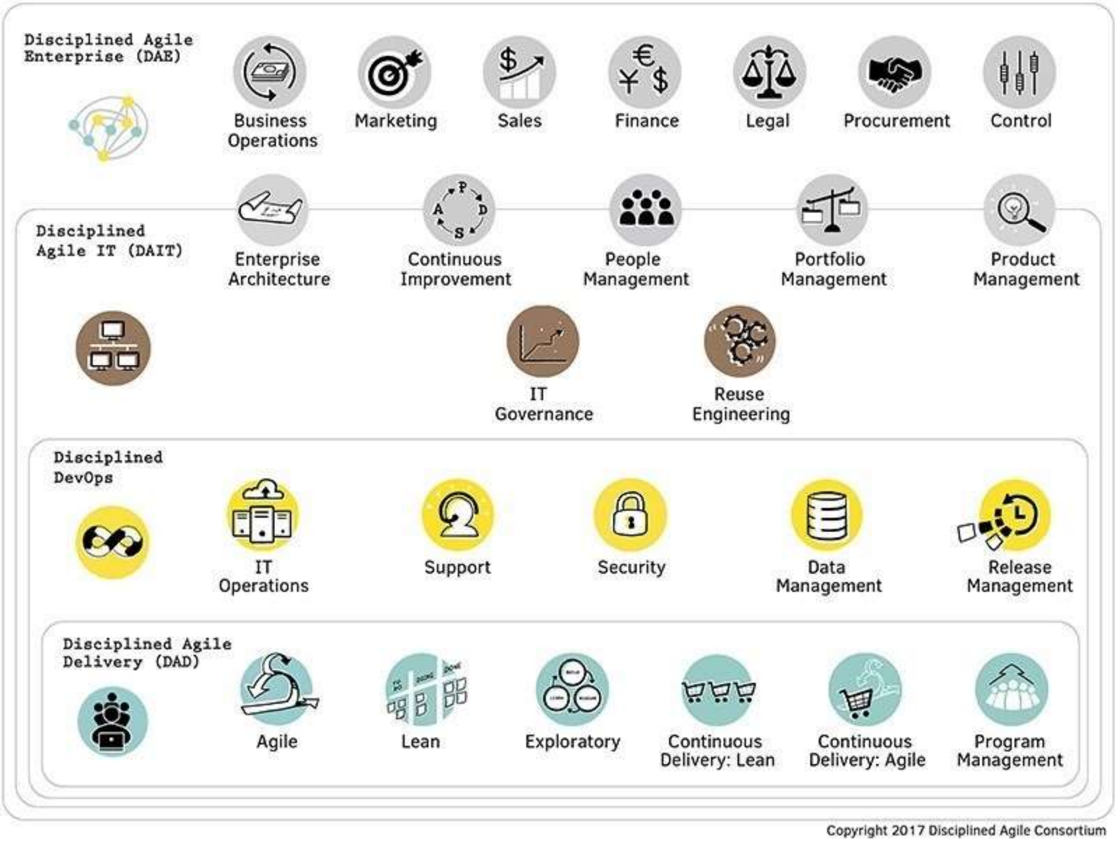


Figure 6: DAD (Disciplined Agile Consortium, 2017)

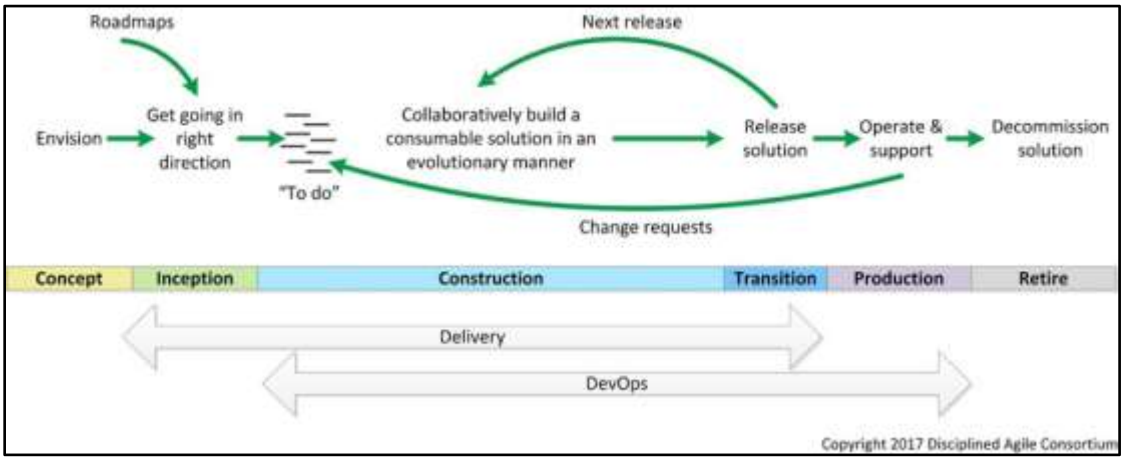


Figure 7: A high-level view of the delivery lifecycle (Disciplined Agile Consortium, 2017)

2.6 Change Management

Due to the fast growth of technology, organizations need to adapt and move faster in order to catch up with the latest technology to gain and keep their competitive advantage. To keep up with new technology, the way of working also need to be changed. Many software development organizations transformed by adopting new ways of working such as Agile (Paasivaara, 2018).

In order to do the transformation in organizations, change management is one of the key topics that should be considered. Change management has been identified as a key success factor in software development industry since this industry is rapidly growing and keep changing all the time (Apostolou et al., 2011). Change management is a structured approach to transitioning individuals, teams, and organizations from the current state to the desired future state (Sacheva, 2009, pp.109). Dias de Lima (2009) stated that change management consists of hard and soft sides. The **Hard side** knowns as procedures, methods, plans, strategies, and technologies that will support change implementation. The **Soft side** consists of behaviors and attitudes toward change which provide the hard changes to accomplish (Nograšek, 2011). Many authors have introduced models on how to manage change. To find the right model that suit organizations is not easy. Nograšek (2011) concluded that managing change is a challenging task. Beloof (2018) introduced eight steps for effective change management which are described in the table below.

Table 1: Eight Steps for Effective Change Management (Beloof, 2018)

No	Steps	Details
1	Identify what will change	- Define the desired outcome of the change - Identify the specific change and clarify goals - Define why a change is necessary and what are looking to improve can help create a solid foundation for a successful implementation
2	Present a solid business case to stakeholder	- Define stakeholders - Build business case
3	Plan for change	- Identify roadmap or plan for the change
4	Provide resources and data for evaluation	- Identify resource (Infrastructure, equipment, or software systems) - Identify data gathering and analysis to measure and monitor progress
5	Communicate	- Providing clear communication throughout the process
6	Monitor and Manage resistance	- Monitor resistance
7	Celebrate Success	- Recognize and celebrate to help people associate the changes with positive feelings
8	Review and improve the process	- Listen to the feedback and review in to adjust and improve the process

When it comes to change in organizations, most of them have been facing resistance to change. Resistance is an instinctive reaction to change for most people (Vrhovec, 2016). Resistance to change is first defined as an action against change in organizations (Lewin, 1947). It is also considered as a behavior to oppose an individual from the consequences of real or imagined change (Zander, 1950). Resistance to change has been proven by many researchers as one of the most severe problems which caused high failure rates in software development (Jiang et al., 2000; Kim, 2011; Meissonier, 2010).

The behavior which creates resistance to change can be divided into two major categories: active resistance and passive resistance. **Active resistance** is behavior that consists of searching for fault, mocking, tempting to fear, and manipulating. **Passive resistance** is behavior that consists of admitting verbally but not following what has been admitted, pretending ignorance, and restarting information (Bolognese, 2002). Lindinger and Göller 2004 study (cited in Fiedler, 2010) shows that categories of resistance based on several distinct reasons which are shown in *Table 2*.

Table 2: Categories and reasons for resistance to change (Fiedler, 2010, pp.371)

Categories of resistance	Reasons for resistance
Generic resistance	Culture of rejection, of refusal
Person-based resistance	Personality as a rejective person (“no-sayer”)
Provoked resistance	Too much pressure, overload due to change requirements and implications
Argumentative resistance	Weaknesses of the intended change concept

According to Ford et al. (2008), most of change management literature sees resistance to change as perceived from the aspects of people who search to evoke change, at the same time recipients create illogical barriers to prevent this change. Resistance is mainly related to employees (Fiedler, 2010). Resistance to change is not necessarily negative (Ford, 2008). It can also be used as a resource for the organization to learn from. Or even help the organization to recover from the malicious change, since not all change is good. PMI (2017) stated that resistance to change can be found in both internal and external environments, even possible to include members of the change organization itself.

Therefore, it is common that managers often do nothing to respond back to resistance when it actually happens, and even if they respond to it, the response is still ineffective (Rivard and Lapointe, 2012). To create an effective response back to resistance, managers have to increase a deeper insight into occurring resistance situations. However, effective response is tough to accomplish without stakeholder involvement due to the complexity of resistance itself (Hultman, 2003).

3 Methodology

This section will discuss methods and techniques that were used to conduct this thesis project, including our research approach and design, the techniques that were selected for the data collection and data analysis, followed by ethical considerations.

3.1 Research Design and Approach

Our thesis is designed and conducted as a single case study. A single case study is referred to as an empirical study that investigates an event in depth within real-life context (Willis, 2014). By using the single case study, it will allow us to study the topic in depth and focus on one particular issue. As a consequence, it will provide a very specific conclusion to the case study. According to Zainal (2007), this study with other studies can be used in a triangulation way which crates a valuable input and provides a more generic conclusion.

Our research approach is to first collect a considerable amount of data. Then we step back and to get a holistic view of the collection data. We then apply existing theory to the collected data and finally come to the conclusion which refers to an inductive research approach (Thomas, 2006). In order for us to answer all the research questions by using this single case study, an in-depth understanding and insight knowledge is required to analyze the situation and develop possible solutions. This indicates a qualitative research approach (Gill, 2008).

Since this subject is still rather new, information has been limited. So, using a single case study with inductive and qualitative research approach seems to be the only option for our study.

3.2 Data Collection

Before we could make an analysis and develop our answer, the questions and goal needed to be defined. Here the data collection was important. A large amount of data was collected mainly through interviews. Some part was also collected by observations. The data was gathered between January 2018 and May 2018.

3.2.1 Interview

The interview is one of the most used technique for data collection to gather information on a specific subject from interviewee's point of view (Gill, 2008). There are many types of interviews, for example, open-ended interview, semi-structured interview, validation interview, etc. (Edwards, 2013). The interview technique was selected as our main method of collecting data. This is because we already had a chance to be present in the organization, which allowed us to be able to easily talk to people within the organization. To answer the research questions, the detailed insight of the organization is required. We conducted three types of the interview which are the open-ended interview, semi-structured interview, and validation interview. In total, we conducted four open-ended interviews, 11 semi-structured interviews, and 12 validation interviews. We started to collect our data by using the open-ended interview method. According to Hoffman (2007), the open-ended interview is a type of interview that starts the session with an open question then the rest of the interview will follow the trend of what interviewee's answer. For this round of the interview, the open-ended interview, the goal is to start learning about the current situation including the organization structure, how they manage and distribute the work and their current way of working with Agile.

The **open-ended interviews** were conducted with people within the organization who presumed to know and understand the big picture of the organization. Our project manager and supervisor at the company provided a list of potential people within the organization who are knowledgeable and could provide us with useful information from. The open-ended interviews were done with invited people in a small meeting room where we had an open discussion on how they work. Each interview took around 30 minutes. The open-ended interview gave insight regarding the organizational structure, their processes in general, their workflow, their product, and included some input of their Agile transformation journey.

After we gained an overview of the department and its current situation, we started our **semi-structured interviews** in which we identify the challenges and problems. The semi-structured interview is one of the most popular ways of gathering qualitative research data which appear to be an interview type that most often used in education evaluation (Griffee, 2005). During the semi-structured interview, the interviewer will use a set of questions to cover the topic. But if issues arise from responses, it allows the interviewer to adapt the question, change order or add an unplanned extra question to clarify the information (Elliot, 2006). The result of the semi-structured interview is very relying on the interviewer's interviewing skills and also required a very well preparation, in order to not make the questions prescriptive or leading (Keller, 2018). The Semi-structured interview consists of casually talking, so it is usually difficult for the interviewer to identify whether or not the data is reliable since there is no way for the interviewer to know if the interviewee is lying. To overcome that, we make sure to conduct enough interviews, so that we could do the comparison and validate the data. We conducted 11 interviews in this round, it took around 60 minutes for each interview. We also asked the interviewees for permission to record the interview. The roles of the interviewees included people from the development team, Operation Product Owner, Area Product Owner, Release Leader, to the Manager Program Office. We had interviewees from all sub-departments with different organization level and different roles, in order to have a complete overview and varieties of perspectives of the department. We listed a set of interview questions, an interview guideline, which consists of 8 questions, shown in the appendix. The questions started with a very general and broad topic e.g., communication and collaboration, and then narrowed down to Agile and SAFe.

After the semi-structured interview, we moved to the **validation interview**. we conducted two types of validation interviews which are information-validation and respondent validation interviews. The term validation interview is defined by Buchbinder (2010) as a session of communication or dialogue between interviewer and interviewee aiming to confirm, verify or correct the researcher's finding. We conducted a total of four information validation interviews and eight respondent validation interviews. We conducted the **information-validation interview** to the people outside of the studied organization. The purpose is to (1) acquire knowledge from those experienced internal people who have an expertise or have experience within a specific area related to what we studied, and also to (2) validate our information and understanding of the earlier interviews. (Example of those discussion topics are; topic about DevOps, discussion about experience and feedback with people from another department who is in the middle of adopting SAFe). Unlike the first part of the interviews, the informal validation interviews were conducted by using different sets of questions depending on the interviewee's expertise and the topic we would like to examine. In addition, we also had a chance to present the concept of SAFe to another group of people outside of the studied department who have interest in Agile methodology. We had a discussion and then gathered feedback and comments about our project. Lastly, we conducted the **respondent validation interview** after we have completed our data analysis. The purpose is to get feedback and validate our result. The interviewees were selected from the list of people from the semi-structured interview. We selected people from different part of the departments and different levels of the organization in order to have a broader perspective on the comments and feedback.

3.2.2 Observation

To gain an understanding of the organizational context, just reading from available resources or listening to some people may not be enough. To get a deeper understanding, we tried to put ourselves closer to the case and tried to get ourselves involved in the circumstance of the working process.

Observation is a research method where the researcher tries to understand a certain topic by studying people in their native environment (Baker, 2006). We conducted our observation at the very beginning of our learning period together with the open-discussion interviews. The aim is to see current situations in practice and to confirm and support the data we collected from the other interviews. We conduct the observations by tag along with a cross-functional team at the company, that was suggested by a line-manager. The team consists of six members. We spent three weeks continuously on observing this team. We tried to attend all the meeting that this team is involved, to see how they work, including the daily scrum meetings and demo meeting at the end of the sprint. We conducted the observations as non-participant observations. According to Cooper (2004), non-participant observations is a research technique that the researcher watch and observe group activities without participating or interfering. Allow the researcher to collect information from neutrality perspective. During the observations, we took notes on what happened in the meeting, what topic was brought up, what was discussed, how this team interacted and behaved. At the same time, we tried to identify challenges faced by people on the team level, which mostly were technical problems. Then, at the end of each meeting, we brought up questions and discussions.

3.2.3 Documents

A couple of documents and diagram from company's internal shared information were also used to help us to understand the organizational structure of this research as well as to identify the current way of working with development at the department.

3.3 Data Analysis

Our data analysis was conducted qualitatively by the following steps. First, we decoded all the interview records we had. Then we tried to manually list all the challenges and problems that were mentioned during all the interviews. Since we obtained a lot of challenges, we have to narrow down the scope of the challenges and problems to what related to their process of working or agile. After that, we prioritized all those challenges and problems by using the amount of the interviewees that lifted up that challenges and/or problems. Then we got a list of eight most important challenges and problems to the case study company.

3.4 Ethics

While we conducted our research on this thesis, we also considered morals and ethics especially with the data privacy to the company and how to deal with all the participants ethically. In term of information and data privacy, we had and followed the data privacy agreement between the researcher and the organization. It is stated that every information regarding the company is confidential, so that we anonymized every information regarding the organization.

We will also keep the confidentiality of all participants (anonymity). Every interviewee that participated in our research of their own free will and allowed to decline the interview invitation. This ensures that all the participants will participate in the study voluntarily. In addition, regarding the process of booking people for an interview, we booked people for an interview only on their available schedule during their working day. And only after we sent an email letting them know the purpose of the interview and asking whether or not they are willing to support this study. On every semi-structured interview, we also asked for the permission to record the interview which they also can always deny if they feel not comfortable with recording the interview.

4 Case Description

Our single case study is conducted from a current situation during the time that we study at a Research and Development (R&D) department. This department is one of the biggest departments at AiTea. The product of this department is products and services of wireless mobile telecommunication technology. The products and services also include hardware, software and also other related support services. This department consists of four sub-departments which we will call department A, B, C and D as *Figure* 8 below.

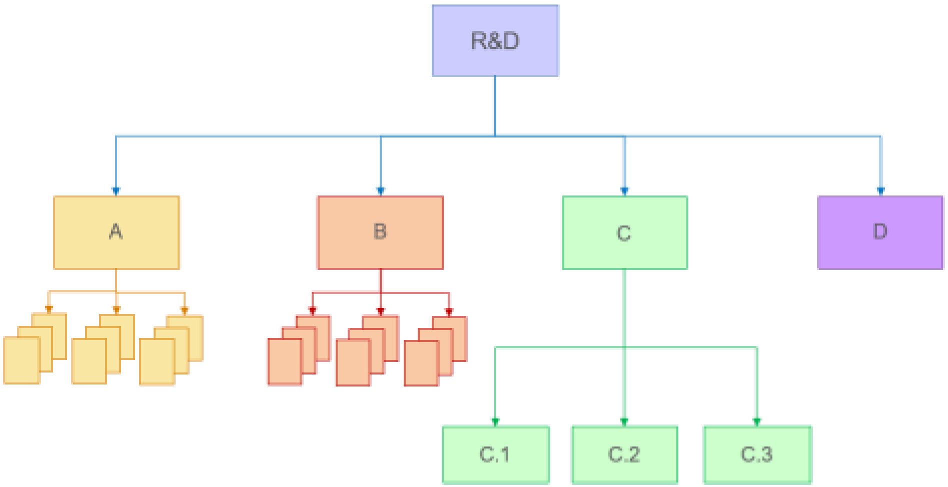


Figure 8: The case study R&D Department Structure

Department A and B are research and development department of two different networking products, which could also be used together depending on the situation and client. While department C is a department to support those two departments regarding releasing the product, maintenance, and also client services. Lastly, department D is a department that provides solutions as a service to clients. The clients of this product are mostly the big telecommunication companies or telephone service providers on a global base (eg. Sweden, America, Thailand) who provide mobile telephone services and fixed telephone services.

The aim of this paper is to start to fill the gap of large-scale Agile research and literature by study the feasibility of applying scaling Agile framework on this single case study. AiTea already went into the Agile transformation during 2012 transforming from what they called their "old way of working". In other words, moving from the waterfall organization model towards the Agile methodology. Their old way of working released the product twice a year. The transformation is many year processes and is still ongoing. However, the implementation of scaling has been done in different ways in different parts of the organization without much support from research and literature. The scale of teams within the department is large approximately 70 to 80 teams. In this case, it is considered as large-scale Agile. So, there is a need to better understand which scaling method works best for the department. AiTea is interested to have a feasibility study on SAFe. Since it claims to support large product development which can be scaled up to thousands of people. SAFe buildup from three main knowledge concepts which are Agile Software Development, Lean Product Development, and System Thinking. SAFe also considered as scalable and flexible by allowing organizations to implement in the way that suits their needs. So, SAFe would be a great consideration for this case study R&D department to study whether SAFe would be fit and help them solve the existing problem or not.

5 Empirical Data

In this section, we answer three research questions which are; RQ1: How are Agile approaches currently used in large software development organization? RQ2: What are the challenges in current agile approaches? RQ4: What are the challenges and impact by applying scaling Agile framework in software development organizations?

According to the data collections that we collected from open discussion, observation, and interview in this case study; 1) The current way of working is a mixture of many Agile methods and frameworks, 2) Seven challenges have been identified with the Agile way of working, 3) Three challenges and impacts have been identified by applying scaling Agile framework. We will discuss these three topics in more details in the following sub-sections.

5.1 Mixture of many Agile methods and Frameworks

Beside from interview selected people across the case study R&D department, we also conduct several observations with development teams and attend some demo-session for new features mainly in department A and B. From observation and interview sessions, the evidence reveals that the case study R&D department is quite mature in an Agile way of working which illustrates in below quote.

> *"The Agile transformation proved to be a successful as the way of working in our department. I think we are working pretty well and quite mature with Agile in development teams. Most people in our department understand the Agile way of working quite well."*
>
> *– Middle Management*

> *"We applied several Agile methods like Scrum, Kanban, XP in our department. I would say that we are a mixture of everything. I think we also borrow some concepts from Less and SAFe to implement here as well. Since we tried to pick up all concepts that suite and help us move faster and be better in the market."*
>
> *– Manager*

As illustrates in above quote, the case study R&D department is a mixture of several Agile methods. Development teams are cross-functional, and each team has a scrum master who acts as a leader to ensure everything is in accordance. The case study department has a hierarchical product owner system as stated by the LeSS method. They divided product owner in three different level which shows in *Figure 9* below. Product owner level 1 is the one that works closely with development teams in operational level. Product owner level 2 works on a holistic picture and mainly focuses on the solution level. They act as a middleman between product owner level 1 and 3. Product owner level 3 is the one that works on the strategic level of organization.

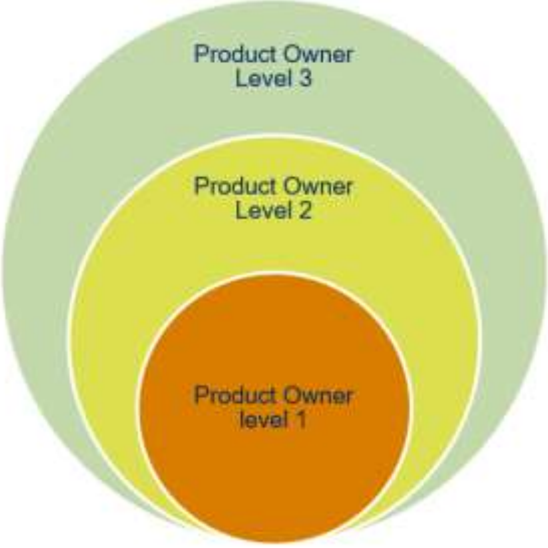


Figure 9: Hierarchical Product Owner

Furthermore, the department C works mainly on Continuous Integration (CI). The development teams have around three to four weeks to implement new features until the code freezes, a period of time that program or product is restricted for change. After the code freeze period, the integration and verification will start testing again to ensure that everything correct. Normally the process takes around one month before new features are released to clients.

To sum up, the evidence reveals that the case study R&D department is mature in the Agile way of working and they are a mixture of several Agile methods and framework which are Scrum, Kanban, XP, LeSS, and SAFe. However, the case study R&D department still has many challenges that they are facing with the current Agile way of working. We will describe all the challenges that exist in more detail in the next sub-section.

5.2 Challenges with the current Agile way of working

During our data collection phase, the interviewees across the R&D department have identified seven main major challenges which related to the Agile way of working (See *Figure 10*). However, all interviewees satisfy with the current Agile way of working and prove that their Agile transformation journey is very successful as mentioned in the previous sub-section.

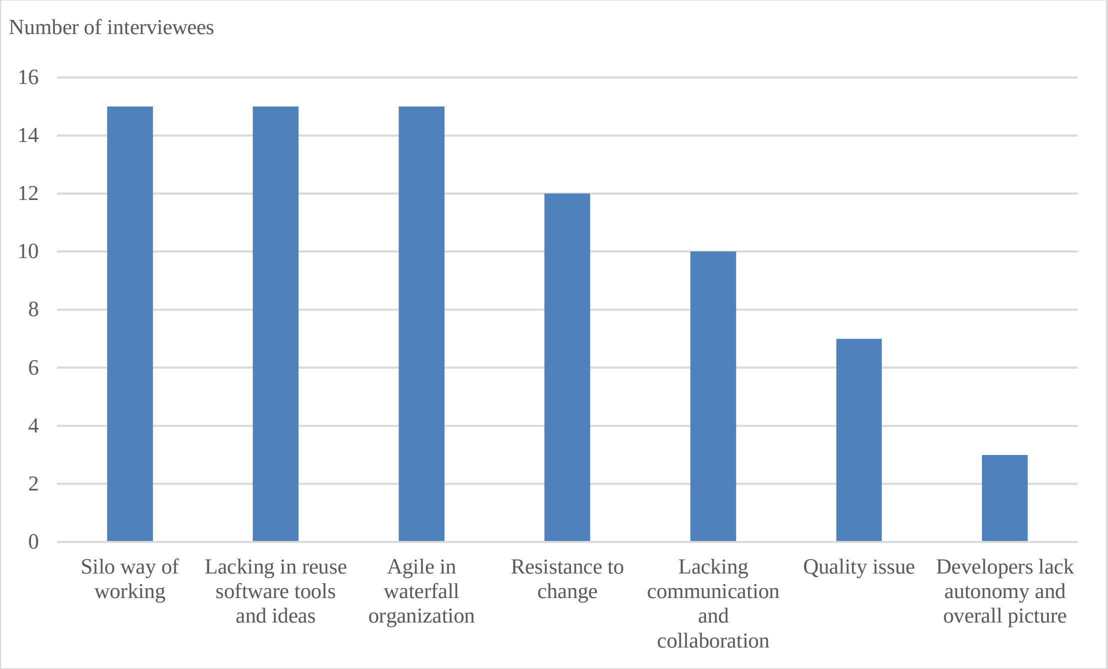


Figure 10: Challenges with the current Agile way of working

5.2.1 Silo way of working and related challenges

This sub-section describes silo way of working and challenges that related to silo which are “Lacking communication and collaboration” and “Lacking in reuse software tools and ideas”.

5.2.1.1 Silo way of working

Silo is a word that used to describe the fact that departments within organizations are isolated from each other. In this case study, all of the interviewees revised that their department has been grown in a silo. Each sub-department does not communicate and collaborate with each other as much as it supposes to be. However, several people in the higher position mentioned that silo does not mean bad all the time, in some sense silo can also be beneficial for the whole organization as well. One big reason behind is that AiTea is an engineering company where employees need to specialize and develop expert knowledge in one specific area. Normally employees tend to focus only on one product that they have been assigned to develop because most of the products are quite complex and need special skill and knowledge to develop.

> *“There is no need for team members in department A and B to communicate and collaborate with each other. Except, they want to share and exchange knowledge since these two products are quite different from each other.”*
>
> *– Senior Management*

However, the silo is considered as a problem when it comes to communication and collaboration between development and operation sub-department within R&D department.

> *"Silo is severe in strategic level with sub-department C when it comes to mid-term and long-term planning. I find it harder to collaborate on a strategic level with them. Sometimes I also have no idea of what my neighbor is doing."*
>
> *– Middle Management*

The evidence shows that silo is considered as one of the main challenges from all interviewees. Most of the time challenge related to collaboration between development sub-department (Department A and B) and release and support sub-department (Department C), as illustrated from the above quotes in this case study.

5.2.1.2 Lacking communication and collaboration

Lacking communication and collaboration are consequence problem that caused by silo way of working. In this case study, lacking communication and collaboration does not exist within sub-department itself. The challenge exists at the department level. The evidence from interviewees shows that department A and B do not have good communication and collaboration with each other. The silo way of working is one of the major reason behinds of this challenge. Since each sub-department has a limited connection between each other, most of the time they have no idea what the other is doing.

> *Communication in general is hard. Our organization is not doing a good job of communicating. If we could communicate better, we will do thing a lot better.*
>
> *– Middle Management*

> *We are in department B have no idea what people in department A are doing. Sometime we found out later that some features that we were developing are quite similar to some features that developed by department A. It would be better if we can have closer communication and collaboration with them sometime.*
>
> *– Team Member*

The evidence from some of the interviewees revises that lacking communication and collaboration related to the culture of the organization that working in a siloed way. It illustrated in the quotes above that close collaboration and better communication can be a benefit for the organization in some situation.

5.2.1.3 Lacking in reuse of software tools and ideas

Most of the time people start to build everything from scratch without looking at ideas or tools that already exist in the organization. This challenging can be seen as a continued chapter of silo way of working. Since each department is isolated from each other and always forget that other departments might have software tools or ideas that they can learn and adapt from other.

> *“It would be better if we can reuse some tools or ideas from other departments. So we can save money and time.”*
>
> *– Product Owner*

> *“It is hard for us to change and try to implement the same platform and tools across the whole organization. When we had a brainstorming session, everyone agreed on the concept that it can help the organization reduce waste of redundant tools and each department can save time by reuse some ideas that already exist. But each department has their own legacy systems that need special tools and knowledge to handle. We have been growing in the silo that each department did not plan and have a chance to talk with each other on this topic.“*
>
> *– Manager*

Evidence reveals that reuse software tools and ideas are a challenging topic for a large organization where each department is in a silo and has limited communication and collaboration with each other. Most of the time people are too busy and do not have time to look at their neighbor. The organization had recognized this challenge and tried to take some action. One team in AiTea is trying to merge in order to have the same software tools and platform across the whole organization as illustrated in the quote above. However, the problem is still hard to solve since each department needs to have special software tools to serve their own legacies. In addition, it is hard to change people mindset as well.

5.2.2 Agile teams in a waterfall organization

All of the interviewees commented that the higher level of the organization still works in a waterfall way of working. In portfolio level, the concept of Agile does not currently exist. Planning for strategies, budgets, and resources are still in the old fashion. On the other hand, the development teams, release management, and testing are living with Agile nowadays. Different ways of working cause conflict within the organization.

> *“We have teams that work in Agile, but the higher level of organization is not that Agile. We are still work in the old fashion (waterfall) up there.”*
>
> *– Developer*

> *“In teams level, we drive by Agile. We have cross-functional teams, scrum master, product owner. But we are lacking Agile in our portfolio level. On that level, we are still live in the waterfall.”*
>
> *– Product Owner*

> *“When it comes to the way of working, our department has a crash in between. It would be better if we have the same way of working with people up there.”*
>
> *– Product Owner*

Even though the organization has been transformed into Agile many years ago, but the whole organization is not Agile yet. From the investigation at the team level, they are mature with the Agile way of working. They implement the concept of cross-functional teams, automated testing system, and continuous integration (CI). The evidence shows that different ways of working cause major problems between employees in a different level of organization as illustrated from above quotes.

However, it does not mean that people in the upper level of the organization against the Agile way of working. There are many reasons behind that make it hard for them to change. One major reason is that their clients are still using waterfall as the way of working. It is hard for them to force their clients to change as illustrated by the quote below:

> *"We also want to change to Agile as well but the problem is not only us. Our clients are still working with waterfall and it does not appropriate to force them to change and follow the way that we are working. To change we need to prove to our clients that Agile really work and make benefit for them."*
>
> *– Senior Management*

The challenges related to portfolio level of organization is not being Agile yet. Evidence reveals that it is harder to change to the Agile way of working in portfolio level because there are a lot of dependencies that need to be considered.

5.2.3 Resistance to change

Since the case study organization is considered as a high technology company. So, there is a need to adapt fast and catch up with new trend and technology in order to be competitive in the market. But when it comes to the words like "New approach", "New way of working", "New tool", and "New framework", not all of the employees within the organization feel comfortable to accept those words. Some employees did not want to accept and change the way of working at all. It happens that sometimes people tend to resist to change their old habit to the new one. They feel like they lost their comfort zone and they feel insecure.

> *"Some people found it hard to let go of the old way of working. Most of the time they feel uncomfortable and lose their control."*
>
> *– Product Owner*

> *"Some feel enjoys and like the new way of working, on the other hand, some feel so hard to abandon the old habit and accepting the new way."*
>
> *– Team Member*

The evidence related to change resistance was strong on a personal level. People feel that they have uncertainty and not sure how to act and prepare for the new way of working.

5.2.4 Test quality issue

In both development sub-department (Department A and B), the development teams have about three to four weeks for developing new features before the code freeze in mainline (The place to integrate code before releasing to clients). After that verification teams will start testing the code again, which take around one month before the new features release to clients. Most of the feedback comes from interviewees in department B that they have more test quality issue when compare to department A. These are many reasons behind the test quality issue in department B which illustrated in the quotes below;

> *"These two departments used different approaches when it comes to quality assurance. The development in department A has to test new features until they are 100% sure that these new features are free of bugs before they submit them to the mainline. In order to reduce the chance to find fault in the mainline as much as possible. On the other hand, department B submit new features to the mainline more often because they applied and using more Continuous Integration (CI) concepts. When they find faults in the mainline, they will pull it back and fix it stead. This is why they found more fault in the mainline when compare to department A."*
>
> *– Product Owner*

> *"The products of department B are more complex and has more pressure from clients when compare to department A."*
>
> *– Middle Management*

> *"Nowadays lead time for both us and clients is reduced. So our clients tend to be more frustrated and concern more with the fault that they found."*
>
> *– Middle Management*

The evidence related to quality came from some interviewees, most of them also revised that the quality issues are better than before at the time of interviews. The challenge is also related to CI, pressure from clients, and complexity of the product.

5.2.5 Developers lack autonomy and overall picture

Product Owner (PO) is the one that prioritizes and assign new features to each development team that under his/her responsible. The evidence from some interviewees shows that development teams do not have a chance to pick up features or speak up that the features do not suitable for their team expertizes.

> *"We have no choice to pick features that really suite for our teams. We have to accept and develop features that PO assigned to us only."*
>
> *– Developer*

Development teams are also lacking the bigger picture and overall of the whole features and products. Most of the time they develop one part of the feature and move on to the other part without connection and order from the previous part, as illustrated in the quote below.

> *"It would be better if we have a chance to know the overall of what we are developing. Most of the time we develop one part of the feature and then jump to develop the other part of the feature. So we never have a chance to know how our code appears and uses as an end product for clients."*
>
> *– Developer*

The evidence reveals that development teams are unhappy with the assignment and lacking an overall picture of the whole feature.

5.3 Challenges and impact by applying scaling Agile framework

In this case study, we focus mainly on Scaled Agile Framework (SAFe). Since it revealed to be the most popular framework among other (LeSS and DAD). As mentioned earlier that the case study organization is considered as a high technology company. So, there is a need to adapt fast and catch up with new trend and technology in order to be competitive in the market. The interviewees in this case study revealed that their organization always keeps changing and adapting to be better and be a leader in the market.

> *"We never stay still to the old way of working that keeps us from success. We keep moving and try to pick up the best thing to implement here in order to become better and be a leader in the market."*
>
> *– Developer*

> *"We are a mixture of many methods and frameworks that we think they are good and suited us. We never stop and limit ourselves to one thing, we keep going and try new approaches and methods that can make us stronger, faster, and better all the time."*
>
> *– Manager*

The evidence from above quotes illustrated that the culture of the organization in this case study does not limit on one particular method or approach. The organization always moves and continuously trying to improve itself. It proves that the concept of SAFe does not fit with case study organization's culture. To apply SAFe in this case study is quite challenging. Since SAFe is more like a box that they cannot jump out which contrast with the culture of the organization. It provides a specific way to implement scaling Agile by introduced specific roles in each level with the specific concepts that organization has to follow. In addition, SAFe is more suitable for small organizations that require steps and framework to scale up their size. But when it comes to this case study department that they already huge at the beginning with 70-80 teams, concepts in team and program levels in SAFe are not useful for them anymore. Lastly, if the organization decides to implement SAFe, one major challenge that they have to face is resistant to change.

To apply SAFe in software development organization. These are three main challenges that revealed in this case study: 1) SAFe is like a box that company cannot jump out which contrast with their culture, 2) SAFe more suitable for small companies that want to scale up, 3) If organization decide to apply SAFe, they will have to face with resistant to change the way of working within organization.

6 Analysis

In this section, we answer our last research question, RQ5: How could such a new framework be implemented in the current Agile organization?

We made analysis mainly focuses on Scaled Agile Framework (SAFe). As mentioned before that SAFe is the most popular framework among other (LeSS and DAD). After we identified the current way of working and all the challenges in the previous section, we analyzed data and compared them with SAFe. In order to find out and answer that SAFe will help the organization to solve challenges that related to the Agile way of working or not.

The evidence shows that the case study R&D department mature in the Agile way of working at the team level. Some concepts in program level already exist in the current way of working. The concept of the large solution also partially exists. However, the case study is lacking most of the Agile concepts in the more higher level when compared with SAFe which illustrated in below *Figure 11*.

Remark: The green circle means that concept exists in the current way of working
The yellow circle means that concept partially exists in the current way of working
The red circle means that concept does not exists in the current way of working

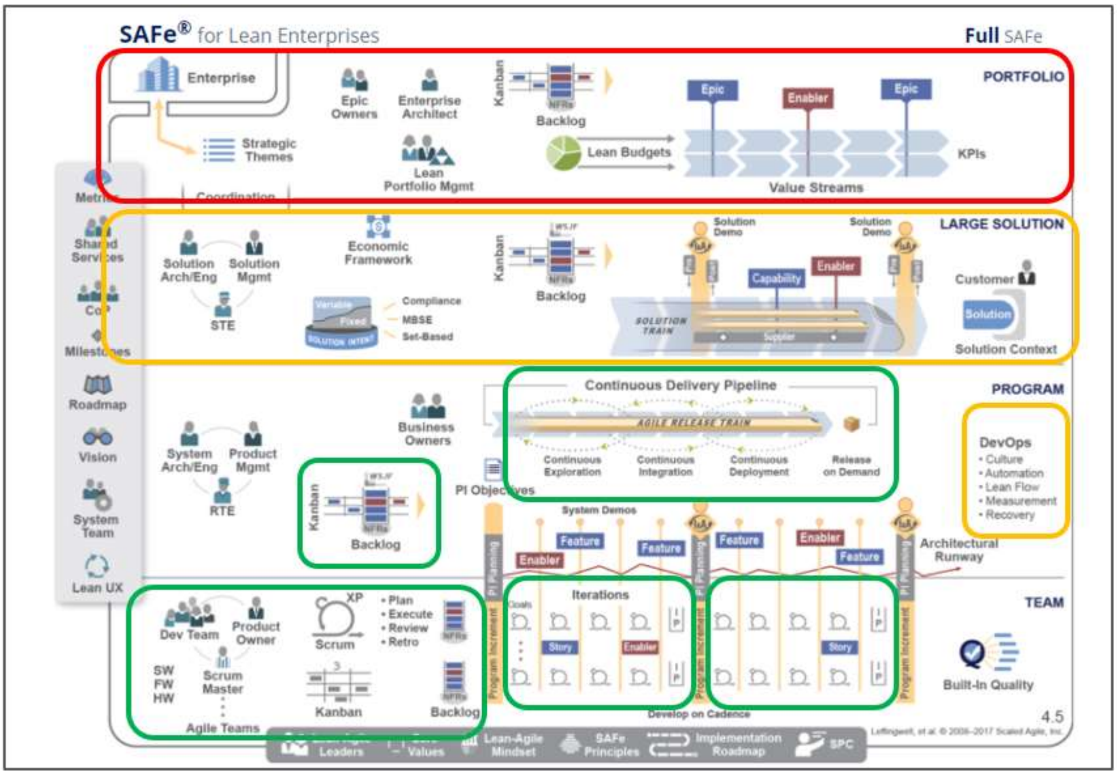


Figure 11: SAFe VS Current way of working

Furthermore, we tried to identify the rest of concepts within SAFe. By analyzing each concept in order to see that which one can help the case study R&D department solve those challenges in the Empirical data section.

Table 3: Challenges VS SAFe

Challenges	Solve by SAFe concepts	Solve by other concepts
5.2.1.1 Silo way of working	X	
5.2.1.2 Lacking communication and collaboration		X
5.2.1.3 Lacking in reuse software tools and ideas	X	
5.2.2 Agile teams in a waterfall organization	X	
5.2.3 Resistance to change		X
5.2.4 Quality issue	X	
5.2.5 Developers lack autonomy and overall picture	X	

After analyzed all concepts within SAFe, we found out that only five challenges can be solved by SAFe which are: 1) Silo way of working, 2) Lacking in reuse software tools and ideas, 3) Agile teams in a waterfall organization, 4) Quality issue and 5) Developers lack autonomy and overall picture. The other two challenges we cannot find any concepts that help to solve those problems which illustrated in *Table 332* above.

7 Discussion

In this section, we discuss our main finding of the case study and compare them with the theoretical frameworks that we presented earlier in this paper. To see whether that data that we collected are similar or different from the theoretical frameworks

7.1 Current Agile situation and challenges

The data that we collected from the interviewees revises that AiTea is mature in the Agile way of working and they use a mixture of several Agile methods and framework which are Scrum, Kanban, XP, LeSS, and SAFe (Schwaber and Beedle, 2002; McKenna, 2016; Lei et al., 2017; Beck, 1999). Which our finding revealed to align with Fitzgerald (2006) that many Agile software development organizations tend to mix these methods to fit their firm in some way. The reason behind this is possibly due to the fact that each Agile method or framework provides a different advantage over some particular area. Plus, the Agile concept itself always supports and gives a big opportunity for improvement. One good example is the retrospective session, a session of self-inspection to discuss and reflect on what happened during the iteration: what went well, and what could be improved. And larger firms tend to have bigger space for improvement, so they could pick some concept from other methods or framework that seems to work or fit with their way of working and then apply those with their Agile approach. However, we also found that the case AiTea still faced many challenges that relate to the Agile way of working for a large company, which we discuss further in this section.

7.1.1 Challenges related to silo way of working

According to the challenge with silo way of working, the silos are the main cause of two other challenges which are 1) Lacking communication and collaboration and 2) Lacking in the reuse software tools and ideas. Furthermore, these two challenges also related to the size of the organization as well. Supported from the literature

First, the communication and collaboration in general is a hard topic to deal with. But it becomes even harder when it comes to large organizations like AiTea. In addition, Agile itself is actually executed based on close communication and collaboration (Hanssen et al., 2011). The R&D department reveals that communication and collaboration problems are hard to be solved due to the size and silo culture in their organization. This also aligned with our theoretical framework that communication and collaboration will become a major challenge within a large distributed organization using Agile (Paasivaara et al., 2018). The silo culture within the organization even makes communication and collaboration between sub-departments less important, since they perceive that there is no need to communicate or collaborate with each other. However, the data that we collected from interviewees reveals that it would be better if they can communicate and collaborate with each other more.

In addition, the lack of communication and collaboration between different people from different organizational levels (in this case, PO and developer) also causes the developers lack in autonomy and the challenge of having the overall picture too. To solve this problem the PO could use this feedback and start by initiating more collaboration and involved development teams in planning session or communicate more information with them. This way developers will have a chance to speak up and feel that they are involved in the process.

Second, it is important for large high technology companies to consider to reduce both time and cost to produce products by reusing software tools and ideas within their organization (Spoelstra, 2011). The case study reveals that it is hard for them when the the size of their organization is so large. Especially when each department is formed as a silo and has limited communication and collaboration with each other. It is even harder when each department has developed as a silo for so long in order to gain expertise in each field and product. Each department also needs to have special software tools and techniques to serve their own legacies system.

In conclusion, to solve these two problems, the R&D department as a whole has to find the right balance of working and breaking through the silos. The consequence could be both positive and negative since it still important to keep expertise in each field. Because most of the products are quite complex and need special skills and knowledge to develop. However, they also need to find a way to break down the silos in order to solve these two problems. Our suggestion is to start to have a monthly or bi-monthly meeting where people from each department (Department A, B, and C) can share knowledge and share what they are doing in their department. This meeting can also improve the communication and collaboration between each department since they have a chance to know each other and get a better picture of what the other departments are doing or facing nowadays.

7.1.2 Agile teams in a waterfall organization

When it comes to the Agile way of working in a large organization. It is revealed from the case study that the whole organization is not Agile yet. They are lacking the agile way of working in the upper level of organization. However, the evidence reveals that they are mature in the agile way of working at the team level and have some part implemented in the middle level of organization.

The R&D department shows that it is harder to change to the Agile way of working in the upper levels of the organization because there are a lot of dependencies that need to be considered. Also, there is a lack of empirical studies and real practical cases on how to apply Agile at the upper organizational level. Since it is very new and not many organizations consider Agile for the whole organization yet. The topic that we found out here also aligned with our theoretical framework that there is a need for more empirical study in this field (Christoph and Jeannette, 2015).

However, to solve this challenge, we would recommend the organization to consider the way of working at the upper organizational level. The way of working in the upper part of organization needs to be aligned with the rest of the organization to reduce the crash in between at least start thinking as a long-term plan.

7.1.3 Resistance to change

It reveals from the case study that the R&D department is facing resistance to change when it comes to the words like “New approach”, “New way of working”, “New tool”, and “New framework”. The evidence from the case study R&D department aligns with our theoretical framework that when it comes to change in organizations, most of the employees have always been facing resistance to change (Vrhovec, 2016).

To solve this problem the organization needs to take action on how to communicate change throughout the whole organization. Relate back to the theoretical framework in order to reduce the resistance to change, it is also important to involve all stakeholders in the process as well. To make sure that everyone gets the same message and understands it correctly. At the same time, the organization needs to have a strong strategic plan of change management in order to have an efficient way to handle the resistant and communicate with the whole organization, according to Hultman (2003).

7.1.4 Quality issue

According to the R&D department, the evidence shows that they are currently facing a quality issue during their product development. However, it does not mean that the quality is bad. Most of the interviewees reveal that the issue regarding product development quality has improved recently.

AiTea already adopted the Continuous Integration (CI) to their department. According to the Wysocki (2014) and Barbee (2013) regarding the Agile Manifesto, the concept of CI is expected to help the organization getting better with the product quality due to the fast feedback from users or clients. The department B, which is facing the quality issue, is also doing great with their CI by applying the “fail fast, fix fast” concept for their product development process. The concept means that they will submit the new feature to the mainline as soon as it finishes, if there is a bug then let it fails so they can fix it as soon as possible. As a result, their quality seems to actually get better over time.

Then, why are they still facing the quality issues? The reason is, in this way, the number of faults was significantly increased. Plus, with their very complex system, it is almost unavoidable that a lot faults might happen. And with a client that did not truly understand the Agile and CI concept, the high number of faults during the development raises their concern, that it might affect the final product quality.

As previously mentioned, the challenge is related to CI, pressure from clients, and complexity of the product. There is no easy way when dealing with an issue involving clients. To get into a better situation, forming more support to the client helping them to gain a better understanding of Agile and CI concept possibly help to reduce the pressure and might even gain better cooperating from the client.

7.2 Possibility to apply SAFe

According to our analysis, we found out that not all of the challenges can be solved by SAFe. Just five out of seven challenges that we found could be solved with the application of the concepts within SAFe: 1) Silo way of working, 2) Lacking in reuse software tools and ideas, 3) Agile teams in a waterfall organization, 4) Quality issue and 5) Developers lack autonomy and overall picture. Most of the concepts are at the higher level of SAFe which is large solution level and portfolio level.

Furthermore, the results also align when we compared the current Agile way of working with SAFe. We found out that the R&D department is mature in the Agile way of working at the team level. Some concepts in the program level already exist in the current way of working. The concept of the large solution also partially exists. However, the upper level of the organization is currently not working with the portfolio level in SAFe. So, according to our finding in our case study and our empirical study, the large-scaled Agile organization is still lacking the Agile way of working in project portfolio management level.

To sum up, the whole SAFe concept is not useful to be implemented in this R&D department. We found out that SAFe is more like a box that company cannot jump out and are limiting organization in their way of working because they are not flexible as has been seen from the empirical finding that most of the Agile software development organization tend to mix many methods and tend to be more flexible (Fitzgerald et al., 2006), which is already our case study organization culture.

As mentioned earlier for applying SAFe, there are three main challenges that revealed from our empirical findings which are 1) SAFe is like a box from which the company cannot jump out which is in contrast to their dynamic culture because SAFe is not flexible, 2) SAFe seems more suitable for small companies that want to scale up rather than companies that already have a large scale at the beginning like AiTea, 3) If organizations decide to apply SAFe, they will have to face resistance to change the way of working within organization.

So, the best way to deal with the earlier mentioned problems in relation to SAFe, is to select only some of the concepts that benefit the case study, which is mostly on the higher level of SAFe, especially at the portfolio level. Like we mentioned earlier that Agile does not exist in the current way of working in portfolio level when compare to the other level of organization. According to the literature, one possible way to scale up Agile into a bigger organization is to implement Agile together with project portfolio management (PPM). Since PPM give an opportunity to make large organizations more agility outside of small projects (Christoph and Jeannette, 2015). This could even help them maximizing their financial value and balancing resources with different projects within the organization (Cooper et al., 1999; Martinsuo and Lehtonen, 2007).

According to Misra et al. (2010), adopting Agile usually demands the entire organization to change because the concept of Agile development does not work for individual uses. This is a support from theoretical framework that there is a need to discuss and apply Agile throughout the whole organization in order to reduce those conflict and challenges.

8 Recommendation

This section will provide the recommendation that related to project management perspective according to what we have learned from our case study. There are two main recommendations which are: 1) Apply Agile portfolio management and 2) Implement change management process.

8.1 Apply Agile portfolio management

According to our analysis and discussion, we found out that implementing the whole SAFe might not be the right solution, but rather implement only some concepts that benefit for the organization is the best way to implement SAFe. Furthermore, evidence revealed that the whole organization is not Agile yet. The major problem of this case study is lacking an Agile way of working in a higher level of organization which is portfolio level when compare to SAFe. The problem caused conflict within the organization due to different ways of working. To solve this problem, we recommend the organization to consider and apply Agile portfolio management in order to align the way of working throughout the whole organization. The agile concept should be implemented in planning for strategies, budgets, and resources. However, the research in Agile portfolio management is meager. There is a need for more empirical study and real practical case on how to apply Agile at the portfolio level.

8.2 Implement change management process

To apply Agile in portfolio management, it is clear from the case study that when it comes to the new way of working the organization will face issues with resistance to change. To minimize the impact that will occur from the change, we recommend the organization to implement change management processes. Below we describe recommended steps of change management processes (Beloof, 2018).

First, identify what will be improved after the change. It is important to clarify the outcome of the change, otherwise there is no point to make any change at all. Second, present a solid business case to stakeholder. Third establish the plan or roadmap for the change by involving every stakeholder in the plan. Forth, identify resource, data gathering, and analysis to measure and monitor progress. Fifth, communicate the change throughout the whole organization in order to keep every stakeholder on the same page and know exactly what will happen. Sixth, monitor and manage resistance. As mentioned in theoretical framework resistance is a natural part of change because people tend to fear the unknown and feel that they will lose control. Seventh, celebrate success to help people associate the changes with the positive feelings. Eighth, review and improve the processes. Change management is a hard topic to handle with. So it is important to listen to the feedback and review in order to adjust and improve the process to make it successful at the end (Beloof, 2018).

9 Conclusion

In this paper, we described how scaling Agile can help improve the current Agile way of working in large-scale organizations. The focus of this paper mainly studies the current situation in the case study R&D department in the high technology company in Sweden. We presented the current Agile way of working, the challenges faced, the comparison between the current Agile way of working and the scaling Agile framework (SAFe), the main findings could be applicable to other large scale software organizations, and the recommendations could be useful for the case study.

There is a lack of research studies on how to conduct a successful Agile transformation in large organizations. Also, scaling the Agile project method is not easy, and many challenges have arisen due to collaboration and communication between Agile teams, as well as the challenge of projects distributed over different time zones and culture.

Many Agile specialists introduced scaling Agile frameworks to manage Agile in a large organization such as SAFe, LeSS, and DAD. All of them have great success stories presented on their web pages. However, these scaling frameworks are still lacking empirical scientific studies on how to be implemented, what kind of challenges are present, and how to overcome those challenges. In this paper, we chose to focus mainly on SAFe because it is the most popular scaling framework among these three frameworks. Our study tried to fill the gap of large-scale Agile research and literature. By studying the possibility of how scaled Agile works with the scaling Agile framework.

First, we answered RQ1: How are Agile approaches currently used in large software development organization? We found out that the current Agile way of working in the case study R&D department is a mixture of many Agile methods and frameworks and they are quite mature at the team level but lacking Agile at a higher level of organization which is portfolio level.

Second, we answered RQ2: What are the challenges in current agile approaches? We identified seven main major challenges which related to the current Agile way of working which are: 1) Silo way of working, 2) Lacking communication and collaboration, 3) Lacking in reuse software tools and ideas, 4) Agile teams in a waterfall organization, 5) Facing resistance to change, 6) Facing quality issue, and 7) Developer lack autonomy and overall picture.

Third, we answered RQ3: How can scaling Agile framework support current Agile approaches? We compared the current Agile way of working with SAFe and found out that only some concepts are benefits. Most of them are at higher levels in SAFe which are large solution and portfolio level.

Forth, we answered RQ4: What are the challenges and impact by applying scaling Agile framework in software development organizations? There are three main challenges that were revealed in the case study: 1) SAFe is rather inflexible which is in contrast with the dynamic and swift changing work culture of the case study, 2) SAFe is more suitable for small companies that want to scale up, 3) If organizations decide to apply SAFe, they will have to face with resistance to change the way of working within the organization.

Fifth, we answered RQ5: How could such a new framework be implemented in the current Agile organization? We found out that the whole SAFe is not useful to be implemented in this R&D department. They are mature in the Agile way of working at the team level. Some concepts on the program level already exist in the current way of working. The concept of the large solution also partially exists. However, the case study is lacking most of the Agile concepts on the portfolio level. So, the best way to implement is to select only some concepts that benefit which mostly on the higher level of SAFe.

For future research, we recommend conducting additional case studies on Agile portfolio management, as research in this area is meager. It also revealed from the case study R&D department that they are lacking the Agile way of working at the portfolio level. There is a need for empirical study and real practical case on how to apply Agile at the portfolio level. Portfolio management is well formed in traditional project management, but it still not often taken up in Agile project management. In addition, we suggest having more scientific studies on the use of scaling Agile framework like SAFe, LeSS, or DAD. Since our study cannot fill all gap and almost no scientific studies on how to be implemented, what kind of challenges are present, and how to overcome those challenges in different environments exists.

10 References

ABRAHAMSSON, P., CONBOY, K., WANG, X. (2009) Lots done, more to do: the current state of agile systems development research.

AGRAWAL, A., ATIQ, M.A., and MAURYA, L.S. (2016) A Current Study on the Limitations of Agile Methods in Industry Using Secure Google Forms. *Procedia Computer Science*, 78, pp.291-297.

AMBLER, SW. (2012) Disciplined Agile Delivery: a practitioner's guide to agile software delivery in the enterprise. *IBM Press*.

APOSTOLOU, D. et al. (2011) A collaborative decision framework for managing change in e-Government services. *Government Information Quarterly*, 28, pp.101-116.

BAKER, L.M. (2006). Observation: A Complex Research Method. *Library Trends*, vol. 55, no. 1, pp. 171-189.

BARBEE, D. (2013) Agile Practices for Waterfall Projects: Shifting Processes for Competitive Advantage. *United States of America: J.Ross Publishing, Inc.*

BECK, K. (1999) Embracing change with extreme programming. *Computer*, 32(10), pp.70-77.

BELOOF, K. (2018) 8 Steps for Effective Change Management. Available http://www.smartsheets.com/ [Accessed 12/06/2018].

BERGER, H. and BEYNON-DAVIES, P. (2009) The utility of rapid application development in large-scale. complex projects. *Inf Syst. J*, 19(6), pp.549-570.

BOLOGNESE, A. (2002). Employee Resistance to Organizational Change. *Winthrop University, Rock Hill.*

BUCHBINDER, E. (2010). Beyond Checking: Experience of the Validation Interview. *Qualitative Social Work*, 10(1), pp.106-122.

CHRISTOPH, J.S., and JEANNETTE, H. (2015) Agile portfolio management: An empirical perspective on the practice in use. *Int. J. Proj. Manag*. 33, pp.140-152.

COOPER, J. et al. (2004). Using participant or non-participant observation to explain information behaviour. *Information Research*, 9(4).

COOPER, R.G., EDGETT, S.J., and KLEINSCHMIDT, E.J. (1997) Portfolio management in new product development: lessons from the leaders I. *Research Technology Management*, 40(5), pp.16-28.

COOPER, R.G., EDGETT, S.J., and KLEINSCHMIDT, E.J. (1999) New Product portfolio management: practices and performance. *J. Prod. Innov. Manag*, 16, pp.333-351.

CROMITY, J. and STRICKER, U.D. (2011) Silo Persistence: It’s not the Technology, it’s the Culture!. *New Review of Information Networking*, 16(2), pp.167-184.

DIAS DE LIMA, J. (2009) Managing Change: Winning Hearts and Minds. *Harvard Business Publishing.*

DIKERT, K., PAASIVAARA, M., and LASSENIUS, C. (2016) Challenges and success factors in large-scale agile transformations: A systematic literature review. Systems Software.

DINGSØYR, T. et al. (2012) A decade of agile methodologies: Towards explaining agile software development. *System Software*, 85, pp. 1213-1221.

DISCIPLINED AGILE CONSORTIUM (2017) The Disciplined Agile (DA) Framework. Available from http://www.disciplinedagiledelivery.com/ [Accessed 20/05/2018]

DYBÅ, T. and DINGSØYR, T. (2008) Empirical studies of agile software development: a systematic review. *Inf Softw Technol*, 50(9-10), pp.833–859.

EDWARDS, R. and HOLLAND, J. (2013). What do the key terms used about qualitative interviews mean?. In What is Qualitative Interviewing? (The 'What is?' Research Methods Series, pp. 1–10). *London: Bloomsbury Academic*. Available from http://dx.doi.org/10.5040/9781472545244.ch-001 [Accessed 06/05/2018]

ELLIOT, M. et al. (2016) Semi-structured Interview. A Dictionary of Social Research Methods: *Oxford University Press*. Available from http://www.oxfordreference.com.proxy.lib.chalmers.se/view/10.1093/acref/9780191816826.001.0001/acref-9780191816826-e-0373. [Accessed 06/05/2018].

FIEDLER, S. (2010) Managing resistance in an organizational transformation: A case study from a mobile operator company. *International Journal of Project Management*, 28, pp.370-383.

FITZGERALD, B., HARTNETT, G., CONBOY, K. (2006) Customising agile methods to software practices at intel shannon. *Eur. J. Information System*, 15(2), pp.200-213.

FORD, J., FORD, L., D’ Amelio, A. (2008) Resistance to change: The rest of story. *Academy of Management Review*, 33(2), pp.362-377.

FOWLER, M. and HIGHSMITH, J. (2001) The agile manifesto, Available from http://www.agilemanifesto.org/ [Accessed 05/03/2018].

GILL, P. et al. (2008). Methods of data collection in qualitative research: interviews and focus groups. *BDJ*, 204, pp.291-295.

GITSCRUM (2018) Agile Glossary Iteration, Available from https://site.gitscrum.com [Accessed 19/05/18].

GRIFFEE, D.T. (2005). Research Tips: Interview Data Collection, *Journal of Developmental Education*,28, no. 3, pp. 36-37.

HANSSEN, G., SMITE, D., MOE, N. (2011) Signs of agile trends in global software engineering research: a tertiary study. In: Global Software Engineering Workshop (ICGSEW) (2011) Sixth IEEE International Conference on, pp.17-23, Available from https://doi.org/10.1109/ICGSE-W.2011.12 [Accessed 14/03/2018].

HODA, R., KRUCHTEN, P., NOBLE, J., MARSHALL, S. (2010) Agility in context. Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications. *ACM, NY, USA*, pp.74-88.

HODA, R., KRUCHTEN, P., NOBLE, J., MARSHALL, S. (2013) Self-Organizing Roles on Agile Software Development Teams. *Software IEEE*, 39(2), pp.422-444.

HOFFMAN, E.A. (2007). Open-Ended Interviews, Power, and Emotional Labor. *Journal of Contemporary Ethnography*, 36(3), pp.318-346.

HOLMSTROM, H., FITZGERALD, B., AGERFALK, PJ., CONCHUIR, EO. (2006) Agile practices reduce distance in global software development. *Inf Syst Manag*, 23(3), pp.7-18, Available from https://doi.org/10.1201/1078.10580530/46108.23.3.20060601/93703.2 [Accessed 14/03/2018].

HULTMAN, K. (2003) Resistance to Change Managing Encyclopedia of Information System

JIANG, JJ., Muhanna, WA., and KLEIN, G. (2000) User resistance and strategies for promoting acceptance across system types. *Information & Management*, 1, pp.25-36.

KELLER, S., CONRADIN, K. (2018). Semi-Structured Interviews. Available from https://www.sswm.info/planning-and-programming/decision-making/gathering-ideas/semi-structured-interviews. [Accessed 06/05/2018]

KIM, H.W, (2011) The of Switching Costs on User Resistance to Enterprise Systems Implementation. *IEEE Transactions on Engineering Management*, 58(3), pp.471-482.

LEI, H. et al. (2017) A statistical analysis of the effects of Scrum and Kanban on software development projects. Robotics and Computer-Integrated Manufacturing, 43, pp.59-67.

MEISSONIER, R. and HOUZE, E. (2010) Toward an 'IT Conflict Resistance Theory': Action research during IT pre-implementation. *European Journal of Information Systems*, 19(5), pp.540-561.

NOGRAŠEK, J. (2011) Change Management as a Critical Success Factor in e-Government Implementation. Business Systems Research, 2(2), pp.1-56.

PAASIVAARA, M., DURASIEWICZ, S., LASSENIUS, C. (2008) Using scrum in a globally distributed project: a case study. Softw. *Process Improv. Pract.* 13(6), pp.527-544.

PAASIVAARA, M. et al. (2013) Integrating Global Sites into the Lean and Agile Transformation at Ericsson. *Software IEEE*. Available from https://ieeexplore.ieee.org/document/6613078/ [Accessed 22/02/18].

PAASIVAARA, M. (2017) Adopting SAFe to Scale Agile in a Globally Distributed Organization. *Software IEEE*, 34, pp. 98-103.

PAASIVAARA, M. et al. (2018) Large-scale agile transformation at Ericsson: a case study. *Empir Softw Eng*, pp.1-47, Available from https://doi.org/10.1007/s10664-017-9555-8 [Accessed 22/02/18].

PETERSE, K. and WOHLIN, C. (2010) The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empir Softw Eng*, 15 (6), pp. 654–693.

PMI. (2013) Standard for Portfolio Management. 3rd Ed. *Project Management Institute, Inc*. (PMI). Online version available from https://app.knovel.com/hotlink/pdf/id:kt00BRNSY7/standard-portfolio-management/what-is-portfolio-management [Accessed 23/04/18].

PMI. (2017) A Guide to the Project Management Body of Knowledge, 6th Edition. *Project Management Institute, USA.*

LARMAN, C. and VODDE, B. (2018) Less Framework, Available from http://less.works/ [Accessed 28/02/18]

LEFFINGWELL, D. (2007) Scaling Software Agility: Best Practices for Large Enterprises. *Addison-Wesley Professional.*

LEFFINGWELL, D. (2010) Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and Enterprise. *Addison-Wesley Professional.*

LEFFINGWELL, D. (2018) Scaled agile framework, Available from http://scaledagileframework.com/ [Accessed 27/02/18].

LEWIN, K. (1947) Frontiers in Group Dynamics: Concept, Method and Reality in Social Science; Social Equilibria and Social Change. *Human Relations*, 1(1), pp. 5-41.

LIVERMORE, JA. (2008) Factors that significantly impact the implementation of an agile software development methodology. *J Softw*, 3(4), pp. 31–36.

MARIA, P. (2017) Adopting SAFe to Scale Agile in a Globally Distributed Organization. *IEEE*, 12, pp.36-40.

MARTINSUO, M., LEHTONEN, P. (2007) Role of single-project management in achieving portfolio management efficiency. *Int. J. Proj. Manag*. 25, pp. 56-65.

MCKENNA, D. (2016). The Art of Scrum: How Scrum Masters Bind Dev-Teams and Unleash Agility. *Apress, Berkeley, CA.*

MISRA, SC., KUMAR, V., and KUMAR, U. (2010) Identifying some critical changes required in adopting agile practices in traditional software development projects. *Int. J. Qual. Reliab. Manag*, 27(4), pp. 451-474.

MOREIRA, M.E. (2013) Adapting to Agile Roles and Responsibilities. In: Being Agile. *Apress, Berkeley, CA.*

RICHARD, B. and STEFAN, W. (2015) Scaled Agile Framework: Presentation and Real World Example. *Software IEEE*. pp. 1-2.

RIVARD, S. and LAPOINTE, L. (2012) Information Technology Implementers Responses to User Resistance: Nature and Effects. *MIS Quarterly*, 36(3), pp.897-920.

SACHEVA, S. (2009) Change Management for e-Governance. I-Ways Journal of E-Government Policy and Regulation 32, *IOS Press*, pp. 109-117.

SCALED AGILE (2018) FullSAFe. [Online image] Available from: http://www.scaledagileframework.com/.

SCHEERER, A., HILDENBRAND, T., and KUDE, T. (2014) Coordination in Large-Scaled Agile Software Development: A Multiteam Systems Perspective. *Hawaii International Conference on System Science*, 47, pp.4780-4788.

SCHWABER, K. and BEEDLE, M. (2002) Agile Software Development with Scrum. Series in Agile Software Development. *Prentice Hall.*

STETTINA, CJ. and HEIJSTEK, W. (2011) Five agile factors: helping self-management to self-reflect. Proceedings of European Software Process Improvement Conference (*EuroSPI 2011), Roskilde, Denmark.*

STETTINA, CJ. and HÖRZ, J. (2015) Agile portfolio management: An empirical perspective on the practice in use. *International Journal of Project Management*, 33, pp. 140-152.

THOMAS, D. (2006). General Inductive Approach for Analyzing Qualitative Evaluation Data. *American Journal of Evaluation*, 27(2), pp.237-246.

VERSIONONE Inc. (2016) 10th annual "state of agile development" survey. [Online] VersionOne. Available from https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf
[Accessed 26/02/18].

VRHOVEC, S.L.R. (2016) Responding to Stakeholders' Resistance to Change in Software Projects: A Literature Review. *MIPRO*, pp.1157-1161.

WILLIS, B. (2014). Advantages and Limitations of Single Case Study Analysis. Available from https://www.e-ir.info/2014/07/05/the-advantages-and-limitations-of-single-case-study-analysis/ [Accessed 07/05/2018]

WYSOCKI, R. (2014) Effective Project Management: Traditional, Agile, Extreme. 6th ed. *Indianapolis, Indiana: John Wiley & Sons, Inc*.

ZANDER, A.F. (1950) Resistance to change: its analysis and prevention. *Advanced Management*, 4(5), pp.9-1.

ZAINAL, Z. (2007). Case Study as a Research Method. Jurnal Kemanusiaan, 9. Available from http://psyking.net/htmlobj-3837/case_study_as_a_research_method.pdf [Accessed 07/05/2018]

11 Appendix

11.1 Interview Questions

1. Interviewee background (e.g. role, tasks, history in the organization)
2. Overview of communication and collaboration within the department (e.g. team communication and collaboration, interaction with other people, success and challenge stories)
3. Overview of the current Agile way of working in the department and organization (e.g. transformation journey, the current way of working, success and challenge stories, opinion about the current way of working)
4. Testing and continuous integration (e.g. testing practice, CI practice, release practice, success and challenge stories)
5. Scaled Agile Framework (e.g. knowledge about SAFe, opinion about SAFe)
6. Challenge and solution for large scaling Agile (e.g. biggest challenges, any solution to solve)
7. Plan for the future (e.g. plan for next step, option for what should be done in the future)
8. Final comment (e.g. anything you would like to comment or add)

11.2 Recommendation Specific to Case Study

In this section, we continue to answer our last research question, RQ5: How could such a new framework be implemented in the current Agile organization? After carefully analyze SAFe, we found out that there are five concepts that might be useful and help the case study department solve the remaining challenges.

11.2.1 Program Increment (PI) Planning

Leffingwell (2018) defined PI planning as a face-to-face event that aligns all the teams to a shared mission and vision (See Figure 12). The main concept of PI planning is to have a face-to-face communication across all team members and stakeholders in order to build the social network, align development to business goals, identify dependencies, and match demand to capacity.

Figure 12: PI planning a face-to-face event

We found out that concept of PI planning might help the case study department to solve challenge 5.2.5 Developers lack autonomy and overall picture. Since the concept itself is to have a face-to-face meeting that requires all stakeholders to participate which including the development team. Involving the developer teams in planning session gives them a chance to speak up on which feature is suitable for their teams. At the same time, the developer teams will get to know more information and overall concept of the features as well.

However, to include everyone in one single room is quite challenging and proved to be chaos. To solve this problem, we recommend that each team nominate one representative to join the session with other stakeholders. The representative role can be rotated within a team, so every team member can have a chance to join planning session.

11.2.2 DevOps

DevOps is a combination of two words which are Development and Operation. Without a DevOps method, there is always a sign of stress between people who develop new features and people who support and maintain steadiness of production environment. If these two group of people are not aligned and have tension between each other, delivery inefficiency can happen. The goal of DevOps is to break down silo by improving collaboration between development department and operation department (Leffingwell, 2018).

We found out that DevOps concept might help the case study department solve challenge 5.2.1 Silo way of working. This concept can breakdown silo between development sub-department (Department A and B) and release and support sub-department (Department C). However, to implement DevOps in this case study is quite challenging because their operations are at customer’s side. They did not own the whole operation process and most of the time their customers will not let them involve on the operation part. In addition, DevOps concept has close collaboration and feedback that might help to challenge 5.2.4 Quality issue as well.

It turns out that to implement DevOps is also depended on their client since most of the operation part is at clients side. So we recommend DevOps concept as a long-term solution for the case study department. The concept will definitely be useful in the future when their clients willing to work closely with them.

11.2.3 Solution Intent

Solution Intent is a crucial knowledge depository to collect, manage, and transmit “What is being built” and “How it will be built” (Leffingwell, 2018). The concept is to provide a single source of knowledge across the whole organization.

The value of using this part of solution intent is to build knowledge storage system, try to reuse what the organization already have, and avoiding redundant work in the upcoming future. The solution intent could relate to the challenge that we found in our case study that the organization itself is not very good at reuse tools and ideas between sub-department or even within the same department. As mentioned earlier in this paper, the evidence shows that this is also a consequence of the silo culture within the organization.

We found that the concept of the solution intent could be the answer to this challenge. Even though the problem itself is very complex, not just there is no platform to share but also people mindset and technical concern. However, by applying the solution intent concept and then make the shared system very easy to access, and very user-friendly, together with trying to change the mindset of every member of the organization could still result in a big difference. This process could take time and should be considered as a long-term plan for the organization.

11.2.4 Large Solution Level

Large solution level is the third layer of the SAFe. This large solution is only needed for the company that facing very large and complex product which beyond the scope of using only single ARTs. The large solution level consists of roles, artifacts, and processes which will provide support for the finding of a solution for those large and complex solutions (Leffingwell, 2018).

We found out that the organization itself has already partially initiated some idea related to this which is the existing of the department D. However, this department D did not solve any internal problem but rather provide more availability and ease of access to their client.

This concept of the large solution level solution could be one of the ways to provide generic perspective for all employees within the organization and to break down the silos between department A and B. They could start applying this concept by combining two product lines into one solution train. Yet they currently have no demand for merging these two departments but our suggestion is to consider this concept for the future solution if they would like to break the silos and merge these two departments together.

11.2.5 Portfolio Level

Portfolio Level is the fourth layer and the highest level of the SAFe. The portfolio level provides Agile portfolio operation and Lean governance and dealing with people and resources needed. And this is where the strategy and funding are handled within the organization (Leffingwell, 2018).

Since this case study organization is facing a crash between operation level and strategy level as one of the biggest challenges. The concept of SAFe portfolio level might help the case study organization to overcome the agile team in waterfall organization challenge. Due to it provides lean budget management plus the high-level management in the Agile way, which will result in reducing the conflict we mentioned earlier.

However, according to our research, since this topic of Agile in portfolio level of a large organization is still very new so there is no record plus no published practical case study or any evidence yet to prove to be a success as implementing this level of the organization to agile. So, we can only recommend considering this concept as a long-term solution for the case study department. The concept will definitely be useful in the future but it will take times to get there.

In addition, we provide six steps for our case study to initiate Agile in portfolio level which are; 1) Decide whether to be Agile throughout the whole organization or not, 2) Leaders need to adopt Lean-Agile concept and be the leader to lead the change throughout the whole organization, 3) Establish strategic theme of the whole organization, 4) Implement portfolio backlog and Kanban system, 5) Implement Lean-Budget practice, 6) Foster Lean-Agile approach to supplier and customer. (See Figure 13 below).

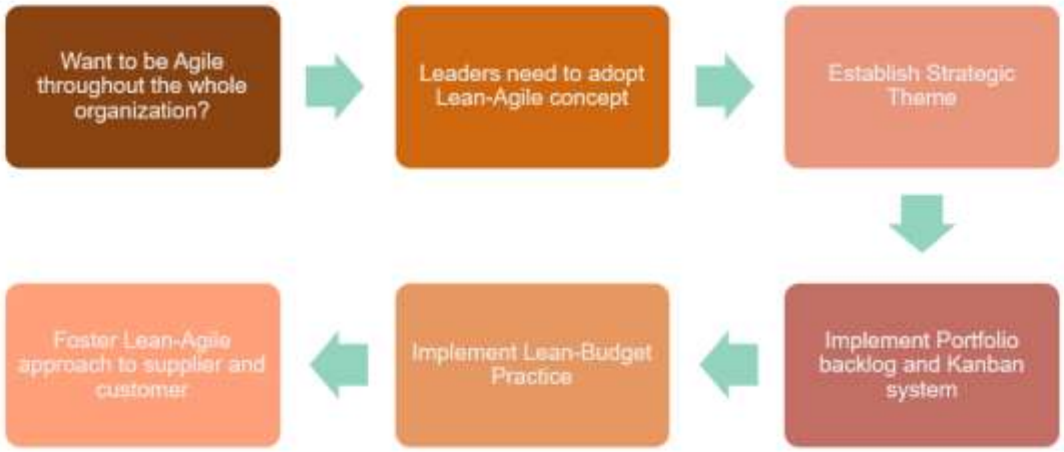

Want to be Agile throughout the whole organization?
Leaders need to adopt Lean-Agile concept
Establish Strategic Theme
Implement Portfolio backlog and Kanban system
Implement Lean-Budget Practice
Foster Lean-Agile approach to supplier and customer

Figure 13: Step to apply Agile in Portfolio level

11.3 Roadmap Map Specific to Case study

After we analyzed data and recommend SAFe concepts for the case study. In this section, we provide four roadmaps for our case study in order to solve all those challenges that we presented earlier in this paper. The first roadmap is to involve development teams into the overall process by apply PI planning concept as we recommended earlier. The second roadmap is to find the right balance of working and breaking the silo. The third roadmap is to set strategy to reuse what the organization already have, and avoiding redundant work in the upcoming future. The forth roadmap is to apply Lean-Agile in portfolio level of organization. (See Figure 14)

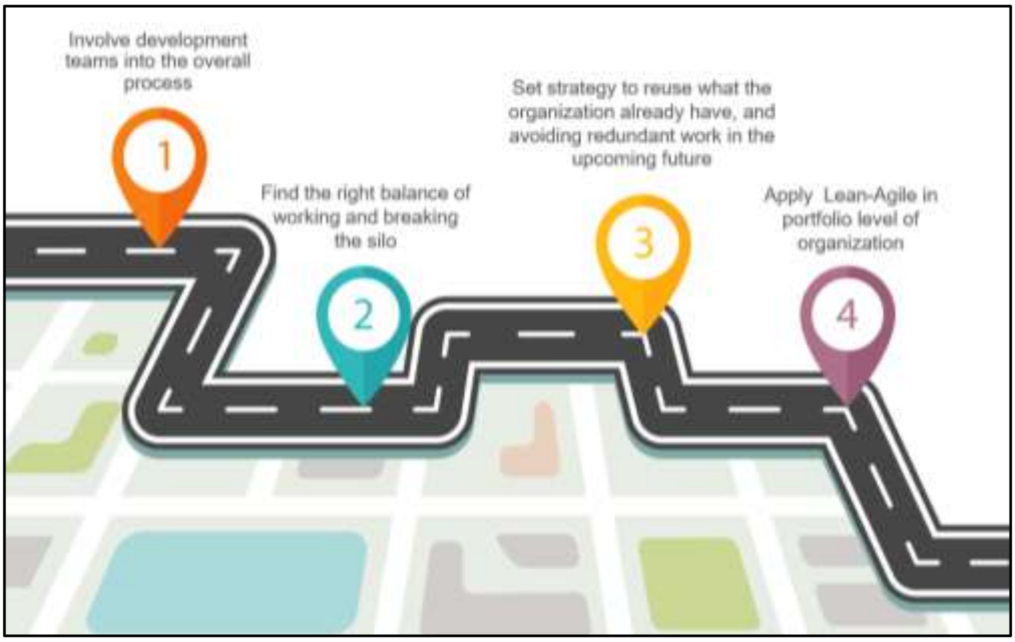

Involve development teams into the overall process
1
Find the right balance of working and breaking the silo
2
Set strategy to reuse what the organization already have, and avoiding redundant work in the upcoming future
3
Apply Lean-Agile in portfolio level of organization
4

Figure 14: Roadmap