Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Patrik Carlson

Scaling Agile Software Development

A Multiple Case Study of the Scaled Agile Framework

Master's Thesis
Espoo, May 21, 2017

Supervisor: Professor Casper Lassenius, Aalto University
Advisor: Maria Paasivaara, D.Sc. (Tech.)

Aalto University
School of Science

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author:	Patrik Carlson		
Title:	Scaling Agile Software Development: A Multiple Case Study of the Scaled Agile Framework		
Date:	May 21, 2017	**Pages:**	vii + 116
Major:	Software and Service Engineering	**Code:**	SCI3043
Supervisor:	Professor Casper Lassenius		
Advisor:	Maria Paasivaara, D.Sc. (Tech.)		

Agile software development methods have been extensively studied in the context for which they were originally developed: small, collocated and cross-functional teams. The proven benefits of agile methods have caused larger organizations and programs to take an interest in them, although adopting such methods at large is far more challenging.

Several frameworks for adopting agile methods in a larger context exist but research surrounding them is scarce, consisting almost entirely of industry-produced experience reports. In this thesis we have studied the adoption of a popular scaling framework, the Scaled Agile Framework (SAFe), in two Finnish companies: Comptel, a telecom company; and NAPA, a company specialized in ship design and operation software. We sought to understand why these organizations chose to adopt SAFe, how they did it, what successes and challenges related to the adoptions the organizations encountered, and what effects the adoptions had.

We reviewed existing literature in order to find out why and how other organizations had adopted SAFe. Using the results of the review we constructed a set of questions which we used in 16 interviews at the case organizations. The interviews spanned all organizational layers of SAFe: we interviewed people from leadership, middle management, and teams.

The results of the interviews support many of the findings of the literature review, showing that planning the adoption in detail, training key personnel, and putting significant effort into the first planning session are key success factors in adopting SAFe. The interviews also showed that internal drivers of change are vital in establishing support for the adoption. Our results indicate that SAFe is a viable option for scaling agile software development, potentially improving visibility, collaboration and alignment.

Keywords:	agile software development, scaled agile, case study, scaled agile framework
Language:	English

Aalto-universitetet
Högskolan för teknikvetenskaper
Examensprogram för datateknik

SAMMANDRAG AV
DIPLOMARBETET

Utfört av:	Patrik Carlson		
Arbetets namn:	Uppskalning av agil systemutveckling: en flerfallstudie av ramverket Scaled Agile Framework		
Datum:	21 maj 2017	**Sidantal:**	vii + 116
Huvudämne:	Mjukvaru- och serviceproduktion	**Kod:**	SCI3043
Övervakare:	Professor Casper Lassenius		
Handledare:	TkD Maria Paasivaara		

Agila systemutvecklingsmetoder blir allt vanligare och har studerats extensivt i den kontext för vilken de ursprungligen utformats: små självorganiserande och samordnade team. De påvisade fördelarna med agila metoder leder allt fler och större organisationer till att försöka implementera dem—dock växer svårighetsgraden i takt med organisationsstorleken.

Ett flertal ramverk för uppskalning av agila metoder existerar, men akademisk forskning kring ämnet saknas nästan helt då majoriteten av litteraturen är fallstudier av varierande kvalité. I detta diplomarbete har vi studerat ibruktagningarna av det populära ramverket Scaled Agile Framework (SAFe) i två finska företag: Comptel, ett globalt telekommunikationsföretag; och NAPA, ett företag specialiserat på mjukvara för formgivning och anförande av fartyg. Vi ville förstå varför företagen valt att ibrukta SAFe, hur de gjort det, vilka problem och framgångsfaktorer de stötte på, samt vilka effekter ibruktaganden hade.

Vi undersökte den existerande litteraturen för att få preliminära svar på de ovannämnda frågorna, och skapade en serie intervjufrågor på basis av resultaten. Dessa brukade vi i 16 intevjuer vid fallföretagen, där vi intervjuade ledare, mellanchefer samt medlemmar av team i ett försök att täcka alla de lager av organisationen som även SAFe täcker.

Intervjuresultaten stöder många av litteraturstudiens resultat: noggrant planerande av ibruktagningen, utbildning av nyckelroller, samt en storsatsning på det första planeringstillfället är viktiga framgångsfaktorer för ibruktagandet. Intervjuerna visade också att interna pådrivare av förändringen är essentiella för att samla stöd för ibruktagandet. Våra resultat tyder på att SAFe är ett genomförbart alternativ för att uppskala agil systemutveckling, då det kan medföra ökad synlighet, ökat samarbete samt bättre anslutning mot ett gemensamt mål.

Nyckelord:	agil systemutveckling, agil uppskalning, fallstudie, scaled agile framework
Språk:	Engelska

Acknowledgements

I would like to thank Comptel and NAPA for allowing me to study their adoption and use of SAFe, and I naturally want to thank all the interviewees who agreed to take the time to participate in the creation of this thesis.

I want to thank professor Casper Lassenius for acting as my supervisor, and especially Maria Paasivaara for acting as my advisor—without your constant input, eager feedback and sound guidance this thesis would not exist.

Ett enormt tack går ut till mina nära och kära: tack för att ni stött mig i ur och skur, tack för att ni tålmodigt lyssnat till min tandagnisslan, och tack för att ni gett mig den extra knuff jag ibland behövt för att komma vidare. Ett stort tack går också ut till alla mina studiekamrater och vänner vid Aalto-universitetet, Teknologföreningen och utanför.

Helsinki, May 21, 2017

Patrik Carlson

Abbreviations and Acronyms

ART	Agile release train
CI	Continuous integration
CoP	Community of practices
DAD	Disciplined Agile Delivery
DevOps	Development and operations
IP	Innovation and planning
KPI	Key Performance Indicator
LeSS	Large-Scale Scrum
PI	Program increment
PM	Product manager
PO	Product owner
R&D	Research and Development
RTE	Release train engineer
SAFe	Scaled Agile Framework
SM	Scrum master
UX	User experience
XP	Extreme Programming

Contents

Chapter 1

Introduction

1.1 Background

Organizations of all sizes are becoming increasingly aware of the proven and potential benefits which agile software development methods offer compared to traditional plan-driven models. However, agile process frameworks such as Scrum are geared for small teams of less than ten people, while large organizations and projects can have several hundreds of employees organized into tens of teams working on a set of products or even a single highly complex product. Implementing agile software development methods in development organizations and projects of larger sizes remains difficult (Dingsøyr and Dybå, 2009).

The increased amount of teams and stakeholders in large projects increases the need for coordination on several levels: inter-team coordination, coordination and interfacing between different parts of the organization, and coordination between customer and the developmental organization becomes increasingly challenging as dependencies multiply. Several frameworks for scaling agile exists: Large-Scale Scrum (Larman and Vodde, 2005), Disciplined Agile Delivery (Ambler and Lines, 2012) and the Scaled Agile Framework (Leffingwell, 2011) all aim to address these challenges by providing a structured approach to employing agile methods at scale. While Large-Scale Scrum and Disciplined Agile Delivery are less prescriptive, the Scaled Agile Framework—or SAFe for short—extends beyond the team, encompassing all organizational layers relevant to software development and defining the roles and responsibilities of each layer. In VersionOne's State of Agile Report (VersionOne, 2016) SAFe is the most popular by far, motivating further research into the framework.

While there are some studies on large-scale agile, academically valid stud-

ies specifically concerning the Scaled Agile Framework are still surprisingly few even though the State of Agile Report (VersionOne, 2016) suggests that use of SAFe is increasing rapidly, jumping from 19 % in 2014 to 27 % of respondents in 2015. While the sample of VersionOne's study is limited to a geographically narrow subset of organizations, it indicates that the use of scaling frameworks and specifically SAFe is widespread, which in turn affirms the need for scientific studies on the subject. Dikert et al. (2016) identifies five topics requiring scientific study, one of which is studying scaling frameworks as these are reported to be in widespread use while lacking in studies. Several of the other identified topics support the topic of scaling frameworks as a research subject, such as the context-dependency of scaling practices as well as the larger enterprise context of an agile transformation of a development organization.

1.2 Research Problem and Questions

The aim of this thesis is to gain a holistic overview of the adoption of the Scaled Agile Framework by studying the case organizations' adoptions in their own context and answering the following research questions:

RQ1: Why have the case organizations chosen to adopt SAFe?

RQ2: How have the case organizations adopted SAFe?

RQ3: What common success factors for adopting SAFe can be identified?

RQ4: What common challenges for adopting SAFe can be identified?

RQ5: What effects has the adoption of SAFe had in the case organizations?

1.3 Structure

This thesis is divided into six chapters. The introduction presents the subject and research questions. The following chapter ventures deeper into agile software development: different agile methodologies and frameworks are introduced together with a more extensive look into the Scaled Agile Framework, and existing research pertaining to the Scaled Agile Framework is reviewed. Next, the research methodology is explained. The results and limitations of the study are presented in the fourth chapter, and the results discussed and compared with the reviewed literature in the fifth chapter. Finally, conclusions and suggestions for future research are made.

Chapter 2

Related Work

This chapter begins with a brief overview of agile software development as well as some of its more established methods and frameworks, and we also explain why companies would choose to attempt to scale them to larger teams and projects. The different methods for introducing agile methods to large organizations are briefly presented and compared. Finally, existing research related to the Scaled Agile Framework is presented and the findings of said research pertaining to each research question is summarized.

2.1 Agile Software Development

Agile software development is a set of practices, frameworks and ways of working related to developing software, best summarized by the Agile Manifesto (Beck et al., 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*
> *Customer collaboration over contract negotiation*
> *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 core of agile software development, or agile for short, is iterative and incremental development in which the most valuable items are constantly being implemented. Feedback is continuously gathered in order to refine

the development practices and better understand the system under development. Compared to traditional plan-driven methods of development—in which the software is planned in detail, developed, and tested before gathering feedback—agile provides teams and stakeholders the opportunity to better react to changes while not wasting work.

There are several frameworks and methodologies which attempt to provide guidelines and rules for developing software in an agile fashion; in this thesis we only shortly introduce those that have influenced or are employed in the Scaled Agile Framework.

2.1.1 Scrum

Scrum is a process framework originally created for software development projects by Jeff Sutherland and Ken Schwaber. Scrum is based around the Scrum Team, which should be small in size (3–9 developers is often stated as a rule), self-organizing and cross-functional. Two additional roles are also defined: the product owner is responsible for the product backlog, maintaining and prioritizing the items within, while the scrum master acts as a facilitator and agile coach, ensuring the efficient use of Scrum in the development team.

Development takes place in time-boxed cycles called sprints, with each sprint producing a potentially shippable version of the product which is demonstrated to stakeholders. The team continuously strives to remove impediments and improve its own ways of working, both through daily stand-up meetings (time-boxed to 15 minutes) and through retrospectives held at the end of each sprint, in which problems and improvement suggestions are discussed more in-depth and action points are added to the backlog. (Schwaber and Sutherland, 2013)

Scrum is non-prescriptive and light-weight in that it does not specify any other aspects than that of the team and its cycle of working.

2.1.2 Extreme Programming

Extreme Programming (XP) is a software development methodology for small or medium size teams (Beck, 1999b) developed by Kent Beck together with Ward Cunningham and Ron Jeffries. It is an agile methodology in which software is developed in short cycles and frequently released in order to better respond to changing requirements. In addition to the agile ideas of iterative and incremental development, XP contains a set of programming principles and practices which also aim to address the challenges of constantly changing requirements. The practices include pair programming, constant testing, continuous integration and constant refactoring (Beck, 1999a).

2.2 Scaling Agile

Implementing the existing agile process frameworks and methodologies as-is in larger organizations and projects is not straightforward as none of them address the realities of programs extending across more than one team. As such, several frameworks for scaling agile methods have been proposed and developed: Large-Scale Scrum (Larman and Vodde, 2005) and Disciplined Agile Delivery (Ambler and Lines, 2012) are presented in their own subsections, while the Scaled Agile Framework is presented in-depth in section 2.3.

2.2.1 Large-Scale Scrum

Large-Scale Scrum is the principles and practices of Scrum scaled up to include multiple Scrum Teams working on one product. The homepage of Large-Scale Scrum states that it is not a framework for scaling agile in which distinctions between the team and other organizational levels or types of work might exist, but rather a "barely sufficient methodology" for applying Scrum at a larger scale than a single team of 3–9 people.

Large-Scale Scrum exists in two versions: LeSS for 2–8 teams (of 3–9 people), and LeSS Huge for eight teams or more. Large-Scale Scrum is less prescriptive than DAD or SAFe in that it does not require any specific roles other than the ones mandated by "regular" Scrum.

2.2.2 Disciplined Agile Delivery

Disciplined Agile Delivery (DAD) is "an enterprise-aware hybrid software process framework" (Ambler and Lines, 2012) developed by Scott Ambler while working at IBM Rational as chief methodologist for IT. The framework extends Scrum with elements from XP, Lean, Agile Modeling and the Unified Process in order to address aspects which Scrum does not explicitly take into account when developing large enterprise-scale software, such as architecture and DevOps. While Scrum focuses on team-level development activities, DAD takes an end-to-end approach of the whole delivery lifecycle. DAD is non-prescriptive concerning the lifecycle of the development, aiming to encompass both cyclical and continous development methods. (Ambler and Lines, 2012)

2.3 Scaled Agile Framework

2.3.1 SAFe

The Scaled Agile Framework (SAFe for short) is a comprehensive set of practices, roles and ways of working which attempts to address the challenges posed by applying agile practices to large organizations. First described by Dean Leffingwell in his book *Agile Software Requirements* (2011) and originally released in 2011 (Scaled Agile, 2016a), the framework lays out ideas and principles for how to build an agile development organization above the team level. The framework is structured and prescriptive: different layers exist for development work, middle management, and leadership; with separate activities and roles for each layer, all of which aim to support the software development process.

The three levels of the framework as per SAFe version 3.0 are described below, together with the associated roles and responsibilities, after which the development process is explained. An overview of SAFe 3.0 can be seen in figure 2.1. While SAFe 4.0 has been released at the time of writing this thesis, SAFe 3.0 was the version in use at the case organizations when the interviews were conducted.

2.3.1.1 Portfolio Level

The highest level in the framework, the portfolio is meant to help align the development organization to the business objectives provided by the enterprise. Value for the customer or the organization is created through value streams, which consist of all effort needed to create the value, and agile release trains realize the value streams.

The strategic themes of the business are broken up into epics in cooperation with stakeholders and added to the portfolio backlog. Epic owners, often in charge of one or two epics, help refine, present and prioritize the epics. When an epic is approved for implementation the epic owner participates in all relevant planning sessions and provide support to the teams implementing said epic. The enterprise architect ensures that technical epics enabling new value creation are given sufficient priority and that the architectural runway of the organization is maintained, while also providing expertise to the analysis of business epics.

Program portfolio management ensures that the content which flows to ARTs and teams is relevant and valuable to the long-term goals of the organization, as they hold the final authority and responsibility over the program execution. By approving epics for implementation they set the direction for

the developmental organization.

2.3.1.2 Program Level

The program level of SAFe is where the coordination of team efforts happens: Agile Release Trains are composed of groups of teams and strive to deliver value for their own value streams. Epics approved for implementation are split into features and added to the program backlog, which is maintained and prioritized by product management.

The teams in the release train develop in the same cadence, which consists of a number of work iterations and one innovation and planning (IP) iteration. This longer time-box is called a program increment (PI), and is jointly planned and reviewed in a session called PI planning. This planning is done by the entire release train with stakeholders during the innovation and planning iteration, from here on referred to as the IP sprint. Beside the PI planning the IP sprint is used for holding Inspect and Adapt workshops, during which a system demo for the results of the whole PI is held; doing integration work between different teams; reducing technical debt; and training.

During the PI planning features are processed by teams and stakeholders, who split the features into stories and assign them to the teams. Dependencies and risks are identified by teams and coordinated jointly to ensure that interconnected stories and features are developed in a logical order.

The release train is driven by a release train engineer (RTE) which acts as scrum master for the entire train: the RTE coordinates scrum masters and product owners, removes obstacles, and facilitates common sessions.

2.3.1.3 Team Level

At team level, SAFe does not differ substantially from other agile software development methods. It is made up of cross-functional and self-organizing agile teams, consisting of 3–9 developers, a scrum master and a product owner. Called a ScrumXP team in SAFe terminology, the practices and responsibilities of the team are taken from Scrum and XP.

The product owner maintains the team backlog together with the team, and the scrum master acts as a servant-leader to the team, removing obstacles and facilitating the improvement of the work processes of the team.

2.4 Existing Research

The adoption rate of SAFe reported by surveys such as State of Agile (VersionOne, 2016) would indicate that SAFe is in widespread use and rapidly growing. The same cannot be said of the available body of research on the adoption and use of SAFe, as most of it is industry-produced case studies of varying quality. The lack of rigorous scientific studies about the adoption of SAFe and agile scaling methods in general form the main justification for this thesis. Wallis (2016) found only a handful of sources which were not industry case reports, and Dikert et al. (2016) also mentions agile scaling frameworks as a potential research topic due to the lack of studies. Only two sources focusing exclusively on SAFe which were not industry case reports or blog posts were found: Brenner and Wunder (2015), a very short conference paper; and Pitkänen (2015), a comprehensive single-case study on the implementation of the Scaled Agile Framework at Rally Software. Rally Software, a software company comprised of over 500 employees, adopted SAFe internally in July 2014. Rally also provides consultation for agile transformations and recommends the Scaled Agile Framework for agile software development at enterprise level (Pitkänen, 2015, p. 32).

2.4.1 Reasons for Adopting SAFe

The reasons for choosing to adopt the Scaled Agile Framework that companies state are varied and not always made explicit. In table 2.1 the main discernible reasons that companies report have been collected; however many of these cannot be said to be actual reasons for either implementing a framework for large-scaled agile or for this framework being SAFe instead of e.g. DAD or LeSS. In order to illustrate this all reasons are listed even though they might not be specifically relevant to SAFe. Some organizations mentioned several reasons and are therefore mentioned multiple times. Reasons mentioned in three or more sources are detailed in their own separate subsections, while all other are presented together in one subsection.

2.4.1.1 Needed a framework to scale agile

The most common reason for the adoption of SAFe in the found literature was that organizations needed to scale agile. They employed some form of agile, e.g. Scrum at team level, but lacked common vision (Pitkänen, 2015), portfolio thinking (Gusch and Herbai, 2015) and coordination (Brenner and Wunder, 2015). In one case, the organization had run into serious challenges

while attempting to scale agile and chose to adopt SAFe in order to mitigate these (Campbell-Pretty and Richards, 2013).

2.4.1.2 Proven, public and in widespread use

Four companies reported one reason for specifically selecting the Scaled Agile Framework was that SAFe was a proven, public framework in widespread use (Scaled Agile, 2014a, 2015b, 2016b; Weltsch-Coen, 2014). While the two main alternatives to SAFe for scaling agile, Large-Scale Scrum and Disciplined Agile Delivery, are also public, they are not reported to be in as widespread use (VersionOne, 2016). Being "proven" is not elaborated upon further in the cases that mention it (Scaled Agile, 2014a, 2016b); presumably the amount of case reports regarding the use of SAFe affects this perception. In Elekta's case, SAFe was chosen because it was the only scaling framework they knew about; the case study also mentions the urgency with which the selection had to be made (Gusch and Herbai, 2015).

2.4.1.3 Consultant recommendation

Six organizations chose SAFe after external consultancies recommended it (Campbell-Pretty, 2016; Ivar Jacobson International, 2014; Lam and Raman, 2014; Rutzen and Roy, 2014; Scaled Agile, 2016b; Vaje, 2014). In two cases, the recommendation was coupled with other reasons (Scaled Agile, 2016b; Vaje, 2014), while the motivations behind the consultant recommendation or the decision made based upon it remain unclear in the other cases.

2.4.1.4 Independent discovery

Independent discovery was less prominent than consultant recommendations. In three cases an employee discovered the Scaled Agile Framework and drove the adoption forward; in TomTom's case Dean Leffingwell's book *Agile Software Requirements* 2011 was specifically mentioned as the source of inspiration for choosing SAFe (Janisse, 2016; Rally Software, 2015; SEI Global Wealth Services). In another case the company incrementally discovered some underlying principles and practices of SAFe and then realized that SAFe would fit well into the organization (Pôle emploi, 2016).

2.4.1.5 Complexity and scale of products

The large programs and complex architecture in use was mentioned as a reason for adopting SAFe in four cases(Gat, 2006; Gusch and Herbai, 2015;

Scaled Agile, 2016b; SEI Global Wealth Services), and the amount of products coupled with the size of the organization in one (Pitkänen, 2015). In one case, the need was specifically to combine complex architecture in a short timeframe (Gat, 2006); in another, SAFe was chosen since Scrum does not address system level design and architectural runway while SAFe does (Gusch and Herbai, 2015).

2.4.1.6 No discernible reason

In three cases no reason was mentioned for the adoption of SAFe (Ball et al., 2015; North, 2013; Scaled Agile, 2015a).

2.4.1.7 Other reasons

In one case the reason for adopting SAFe was exceedingly clear: the client adopted SAFe, and the provider wanted to align itself with the client (Scaled Agile, 2014b). For HPE (Scaled Agile, 2016c), the reasons were seemingly due to a need for general agility: the company wanted to be able to respond more quickly to changing customer requirements and business environment shifts while at the same time cutting costs. Amdocs, in addition to the other reasons, also listed the level of detail SAFe contains as a beneficial factor in the selection; this enabled Amdocs to implement SAFe practices at all levels (Scaled Agile, 2014a).

2.4.2 Adoption Approaches

This subsection contains both the general approaches to the transformation process which organizations chose and the more specific practices which organizations employed. The approaches organizations chose for adopting SAFe can be split into two main categories: big bang, in which the whole organization or sub-organization took SAFe into use at the same time; and piloting, in which some part of the organization adopted SAFe first after which the rest of the organization followed.

The underlying organizational realities and specific practices employed by the organizations vary significantly, and as with the adoption reasons it is difficult to specifically identify practices regarding SAFe. Continuous integration (CI) and test automation are not specific to SAFe, but might be introduced into the organization as part of a SAFe adoption. Synchronizing iterations, on the other hand, is a practice recommended by SAFe.

As drawing the line between SAFe practices and general agile practices is not unambiguous it has not been done: all explicitly reported aspects have

Table 2.1: Reasons for Adopting SAFe

Reason	**Described by**
Needed a framework to scale agile	Brenner and Wunder (2015); Gusch and Herbai (2015); Holdorf (2011); Scaled Agile (2012, 2015b); SEI Global Wealth Services; Vaje (2014); Weltsch-Coen (2014)
Proven, public and in widespread use	Gusch and Herbai (2015); Scaled Agile (2014a, 2015b, 2016b); Weltsch-Coen (2014)
Consultant recommendation	Campbell-Pretty (2016); Ivar Jacobson International (2014); Lam and Raman (2014); Rutzen and Roy (2014); Scaled Agile (2016b); Vaje (2014)
Adopted by client	Scaled Agile (2014b)
Independent discovery	Janisse (2016); Pôle emploi (2016); Rally Software (2015); SEI Global Wealth Services
Complexity and scale of products	Gat (2006); Gusch and Herbai (2015); Pitkänen (2015); Scaled Agile (2016b); SEI Global Wealth Services
No discernible reason	Ball et al. (2015); North (2013); Scaled Agile (2015a)

been listed here. Approaches or practices which are reported in three or more sources are detailed in their own separate subsections, while the remaining ones are listed in a common subsection after them. All approaches are listed in table 2.2.

2.4.2.1 Big bang

Of the case reports studied, five companies (Campbell-Pretty, 2016; Cobb, 2012; Holdorf, 2011; Scaled Agile, 2014b; SEI Global Wealth Services; Weltsch-Coen, 2014) reported choosing to implement SAFe across the organization at once, two of which (Campbell-Pretty, 2016; Scaled Agile, 2014b) explicitly mentioned doing an "Agile Release Train QuickStart" while one (SEI Global

Wealth Services) did a QuickStart but did not explicitly use the term. In a QuickStart, 50–100 people are organized into an Agile Release Train during five days, first receiving a two-day training, then holding a two-day PI planning session and finally holding more in-depth training for the product owners and scrum masters. (Scaled Agile, 2016d)

2.4.2.2 Piloting

Piloting was more frequent than big bang approaches, specifically mentioned in eight of the case reports (Gat, 2006; Gusch and Herbai, 2015; Lam and Raman, 2014; Pôle emploi, 2016; Rutzen and Roy, 2014; Scaled Agile, 2015a,b, 2016c). Of these, three companies (Gat, 2006; Scaled Agile, 2015b, 2016c) began by piloting a team or several, while five companies piloted several ARTs—but not the whole organization—at once (Gusch and Herbai, 2015; Lam and Raman, 2014; Pôle emploi, 2016; Rutzen and Roy, 2014; Scaled Agile, 2015a).

2.4.2.3 Bottom-up transformation

In three cases the adoption of SAFe was reported to have been bottom-up, meaning that some employee(s) or team(s) took the initiative to adopt SAFe and drive change by building inner consensus first (Cobb, 2012; Gat, 2006; Gusch and Herbai, 2015; Janisse, 2016; North, 2013). Of these, three companies systematically built their implementation of SAFe starting from the Team Level up to the Program Level and the Portfolio Level (Gusch and Herbai, 2015; Janisse, 2016; North, 2013).

Elekta also reported that their change had top-down mandate but was a bottom-up transformation, and was the only case specifically mentioning that the initiative came from upper leadership or middle management (Gusch and Herbai, 2015). However, it might be feasible to assume that this was the case for many companies based on the amount of reports in which leadership and middle management received SAFe training first.

2.4.2.4 Management and leadership training prior to adoption

By far the most widely reported practice was to train leadership and middle management as well as key SAFe team roles such as scrum masters and product owners before rolling out SAFe to the rest of the organization, in line with directions by Scaled Agile. This was mentioned in nine cases (Holdorf, 2011; Ivar Jacobson International, 2014; Janisse, 2016; McMaster, 2014; North, 2013; Scaled Agile, 2014a, 2015b, 2016b; SEI Global Wealth Services), with

two of them also mentioning the importance of buy-in from leadership and how trainings helped cement it.

2.4.2.5 Formal training for everyone

Holding formal training for every single stakeholder was only mentioned directly in four cases (Campbell-Pretty, 2016; Rutzen and Roy, 2014; Scaled Agile, 2014a,b; SEI Global Wealth Services).

2.4.2.6 Use of external consultants and agile coaches

The use of external consultants was not only relegated to helping organizations select SAFe as the framework for scaling agile, it was also prominent in the actual adoption of SAFe via the use of agile coaches and trainers (Cobb, 2012; Gat, 2006; Gusch and Herbai, 2015; North, 2013; Rutzen and Roy, 2014; Scaled Agile, 2014a, 2016c; SEI Global Wealth Services).

2.4.2.7 Other approaches

If a value stream contains multiple ARTs Scaled Alliance recommends synchronizing all of them around a common cadence, with synchronized iterations and PIs for all ARTs (Scaled Agile, 2016e). While not mentioned if the ARTs are organized into value streams, in one case the long-term aim is to synchronize iterations and PI's across the whole organization, simultaneously employing normalized story points and consistent definitions (such as Definition of Done) (Ball et al., 2015).

In one organization a separate SAFe adoption team, made up of agile coaches, was formed with the aim of coordinating and supporting the adoption (Scaled Agile, 2014a).

While Dikert et al. (2016) identified one success factor to be the selection and customization of the approach, in two cases the approach was to follow SAFe guidelines as closely as possible with minimal customization (McMaster, 2014; Scaled Agile, 2014a). In the case of Seamless, major customization was made in order to adapt SAFe to a smaller company (Lundgren and Pająk, 2015). Seamless sought to implement the major themes of SAFe, employing both a program and portfolio level in addition to the team level, and maintaining a longer planning horizon. The role of RTE was not used, instead having a separate rollout team coordinating with a program manager, and scrum masters were repurposed into full-time agile coaches.

While continuous integration (CI), test automation and other DevOps practices are not inherent in SAFe, in two cases the adoption of SAFe also

spurred implementation of these agile practices (Gat, 2006; Scaled Agile, 2015a).

Table 2.2: Adoption Approaches

Approach	**Described by**
Big bang	Campbell-Pretty (2016); Cobb (2012); Holdorf (2011); Scaled Agile (2014b); SEI Global Wealth Services; Weltsch-Coen (2014)
Piloting	Gat (2006); Gusch and Herbai (2015); Lam and Raman (2014); Pôle emploi (2016); Rutzen and Roy (2014); Scaled Agile (2015a,b, 2016c)
Bottom-up transformation	Cobb (2012); Gat (2006); Gusch and Herbai (2015); Janisse (2016); North (2013); Scaled Agile (2016c)
Management and leadership training prior to adoption	Holdorf (2011); Ivar Jacobson International (2014); Janisse (2016); McMaster (2014); North (2013); Scaled Agile (2014a, 2015b, 2016b); SEI Global Wealth Services
Formal training for everyone before adoption:	Campbell-Pretty (2016); Rutzen and Roy (2014); Scaled Agile (2014a,b); SEI Global Wealth Services
Use of external consultants and agile coaches	Cobb (2012); Gat (2006); Gusch and Herbai (2015); North (2013); Rutzen and Roy (2014); Scaled Agile (2014a, 2016c); SEI Global Wealth Services
SAFe QuickStart	Campbell-Pretty (2016); Scaled Agile (2014b); SEI Global Wealth Services
Minimal customization	McMaster (2014); Scaled Agile (2014a)
Management and leadership buy-in first	Gusch and Herbai (2015); Holdorf (2011)
Train all at once, at same place, with same instructor	Campbell-Pretty and Richards (2013); Scaled Agile (2014b)

CONTINUED ON NEXT PAGE

Approach	Described by
Synchronized iterations and PI's across organization	Ball et al. (2015)
Top-down mandate	Gusch and Herbai (2015)
Normalized story points and consistent definitions	Ball et al. (2015)
Separate adoption team	Scaled Agile (2014a)
Aligned non-developmental parts of organization	Scaled Agile (2016b)
Implemented DevOps practices	Gat (2006); Scaled Agile (2015a)
Scaled down SAFe	Lundgren and Pająk (2015)

2.4.3 Success Factors

Several success factors related to SAFe adoptions were found when reviewing literature; all of them are listed in table 2.3. Some factors are specific to the successful adoption of SAFe, e.g. PI planning, but others might be related to agile transformations in general; discerning which is which is often not clear and, as such, all factors reported have been listed. For a success factor to be common it has to be present in three or more sources. The most often mentioned success factors are listed in their own separate subsections, after which the remaining factors are presented.

2.4.3.1 PI planning

An integral part of the Scaled Agile Framework is the concept of program increments: longer time periods which group iterations together and bring cadence to the work. The increment is usually preceded by a planning meeting or planning days, and in multiple cases the organizations reported that the planning itself was a major success factor in adopting SAFe (Brenner and Wunder, 2015; Campbell-Pretty, 2016; Ivar Jacobson International, 2014; Lam and Raman, 2014; Pitkänen, 2015; Scaled Agile, 2012, 2015b, 2016b).

In some cases the first PI planning held with the ART set the tone for the transformation that followed, and influenced both future PI plannings and ways of working while also establishing support for the change (Campbell-Pretty, 2016; Ivar Jacobson International, 2014). Other reported benefits

of the planning session was better alignment, especially concerning off-site teams (Scaled Agile, 2016b), and better identification and management of dependencies (Ivar Jacobson International, 2014; Scaled Agile, 2015b).

2.4.3.2 Collocation

Collocation is not specific to SAFe but was still mentioned as a success factor in some cases (Gusch and Herbai, 2015; Lam and Raman, 2014; Scaled Agile, 2016b; Vaje, 2014). Collocated teams was specifically name in one case (Vaje, 2014), while the others reported that face-to-face PI planning sessions were highly beneficial and productive (Gusch and Herbai, 2015; Lam and Raman, 2014; Scaled Agile, 2016b), in one case even "mandatory" when launching an ART (Scaled Agile, 2016b).

2.4.3.3 Incremental and iterative change

An agile approach to the adoption itself was mentioned as a success factor in several cases (Gusch and Herbai, 2015; Scaled Agile, 2015b, 2016c), meaning that organizations attempted to quickly adapt their SAFe adoption according to what was working rather than strictly adhering to guidelines and recommendations.

2.4.3.4 Visualization of dependencies and risks

The visualization of dependencies and risk, both during planning sessions and during iterations, was deemed a success factor in some cases (Ivar Jacobson International, 2014; Scaled Agile, 2015b; Weltsch-Coen, 2014). Making the dependencies explicit helped teams prepare for them before they manifested, and enabled teams to take action once they had done so (Scaled Agile, 2015b).

2.4.3.5 Normalized estimation and prioritization

Using a standardized measure for estimation across the organization helped predictability and reliability (Scaled Agile, 2015b). In two cases, the recommended practice of Weighted Shortest Job First (WSJF) was described as essential in scaling, both as an enabler (Campbell-Pretty and Richards, 2013) and as a way of quantifying the cost of delay (Scaled Agile, 2012).

2.4.3.6 Leadership buy-in and support

Gaining buy-in from leadership, especially middle management, was deemed important for the SAFe adoption in two cases (Campbell-Pretty, 2016; Scaled

Agile, 2014b). Leadership support throughout the adoption was specifically mentioned in one case (Gusch and Herbai, 2015).

2.4.3.7 Other success factors

Having an ample budget for the adoption and allocating sufficient funds and time it, as well as organizing thorough training for all stakeholders, helped the adoption in two cases (Gusch and Herbai, 2015; Scaled Agile, 2016c). One company used gamification in order to inspire teams to adopt Scrum practices: points were awarded for e.g. holding the relevant sessions, and a scoreboard was kept with (with no management follow-up of the points) (Scaled Agile, 2014a).

Establishing metrics for measuring the progress of the transformation itself was deemed useful in one case (Lam and Raman, 2014), while in another case the tools employed for tracking said metrics, as well as coordinating and collaborating across different sites (e.g. video conferencing during PI planning), was listed as a success factor (Scaled Agile, 2016b).

Piloting agile in smaller teams helped create support for the agile transformation as a whole in two cases, both by showing off successes to the rest of the organization and by creating agile evangelists which help drive the change (Gat, 2006; Scaled Agile, 2016c). While the use of consultants was almost as widespread as piloting as an adoption approach only two cases specifically listed their use as a success factor in adopting SAFe (Gusch and Herbai, 2015; Weltsch-Coen, 2014).

In two cases the use of a daily Scrum of Scrums in conjunction with the daily stand-up meeting was deemed useful (Richards, 2013; Weltsch-Coen, 2014); in one of the organizations they called it the daily cocktail party in which all teams held their stand-up meetings at the same time (Richards, 2013).

Having dedicated product owners (PO) was mentioned in one case as success factor in the adoption (Vaje, 2014). People in the PO role were given training and certification, and focused exclusively on their own teams. In the same case, having well-defined processes for the whole organization to use helped reliability.

While DevOps practices are not specific to SAFe they were explicitly mentioned in two cases: both mentioned test automation as important (Janisse, 2016; Scaled Agile, 2012) and in one case continuous integration and regression test suites were also employed.

Table 2.3: Adoption Success Factors

Success factor	**Described by**
PI planning	Brenner and Wunder (2015); Campbell-Pretty (2016); Ivar Jacobson International (2014); Lam and Raman (2014); Pitkänen (2015); Scaled Agile (2012, 2015b, 2016b)
Collocation	Gusch and Herbai (2015); Lam and Raman (2014); Scaled Agile (2016b); Vaje (2014)
Incremental and iterative change	Gusch and Herbai (2015); Scaled Agile (2015b, 2016c)
Visualization of dependencies and risks	Ivar Jacobson International (2014); Scaled Agile (2015b); Weltsch-Coen (2014)
Normalized estimation and prioritization	Campbell-Pretty and Richards (2013); Scaled Agile (2012, 2015b)
Leadership buy-in and support	Campbell-Pretty (2016); Gusch and Herbai (2015); Scaled Agile (2014b)
Use of consultants	Gusch and Herbai (2015); Weltsch-Coen (2014)
Daily Scrum of Scrums	Campbell-Pretty and Richards (2013); Weltsch-Coen (2014)
Ample budget and training	Gusch and Herbai (2015); Scaled Agile (2016c)
Piloting	Gat (2006); Scaled Agile (2016c)
Collaboration and conversation	McMaster (2014); Vaje (2014)
Gamification	Scaled Agile (2014a)
Collaboration and tracking tools	Scaled Agile (2016b)
Upper leadership support	Scaled Agile (2014b)
Product owners	Vaje (2014)
Well-defined processes	Vaje (2014)

CONTINUED ON NEXT PAGE

Success factor	Described by
Negotiate scope with customer	Pôle emploi (2016)
Metrics for measuring transformation	Lam and Raman (2014)
DevOps practices	Janisse (2016); Scaled Agile (2012)

2.4.4 Challenges

Fewer challenges than successes were reported, and fewer still were reported in several cases. As with sucess factors, the challenges specific to the adoption of SAFe are seldom explicitly separated from those encountered during any agile transformation or organizational change, so in this section all challenges encountered have been listed. The most commonly reported challenges, with three or more mentions, are described in their own sections while the rest are described in a general section. All challenges as well as their sources are listed in 2.4.

2.4.4.1 Staffing

Finding people to fill the roles set out by SAFe was found to be difficult in five cases (Campbell-Pretty and Richards, 2013; Gat, 2006; Holdorf, 2011; McMaster, 2014; SEI Global Wealth Services). The role of scrum master was mentioned in two cases (Holdorf, 2011; SEI Global Wealth Services), the challenge being finding people willing to commit to the role especially long-term. The difficulty of the product owner role, and the challenge of finding the correct people to fill it, was mentioned as an adoption challenge in four cases (Campbell-Pretty and Richards, 2013; Holdorf, 2011; McMaster, 2014; SEI Global Wealth Services).

2.4.4.2 Change resistance

Change resistance towards the adoption of SAFe and the agile transformation was specifically mentioned as a challenge in three cases (Janisse, 2016; McMaster, 2014; Pitkänen, 2015), although it is reasonable to believe that any major organizational change causes at least some form of resistance towards it. In one case, the internal resistance was amplified by the lack of middle management support of SAFe, as this was seen as tacit disapproval of the adoption of the framework (Janisse, 2016). There also existed differing

opinions about the necessity of change and whether SAFe as a whole was compatible with the organization.

2.4.4.3 Other challenges

Distributed teams presented a challenge both to the adoption itself and to the coordination of teams (Gat, 2006; Gusch and Herbai, 2015). Additional challenges in that case stemmed from some team members working for several teams, and from requirements spanning across several teams making it difficult to tell when a feature was considered done.

Release management was deemed a challenge in one case (Gusch and Herbai, 2015), stemming from the business environment requiring the use of stage gates for going to production.

The lack of agility in the surrounding organization was stated as a challenge in one case (McMaster, 2014): as funding and reporting were not adapted to agile software development the adoption of SAFe practices and relevant metrics was difficult since e.g. the deliverables were assumed to be known one year in advance. In the same case, test automation was also deemed a challenge.

Organizing the PI planning was discussed in several cases but challenges mentioned explicitly in only one case, in which getting stakeholders to participate the whole two days was the most challenging part (Weltsch-Coen, 2014).

Weighted Shortest Job First (WSJF) requires some sort of normalized metric for evaluating stories and features; the normalization of story size was reported as a challenge in one case (Richards, 2013). As each team had been using their own definition of the size of a story point, normalizing them across the entire agile release train was not trivial. In one case, writing good stories was named a challenge due to the complex architecture of the company's products (Lam and Raman, 2014). The need for deep analysis and architecture design of features was mitigated by writing architectural Spikes, which in turn were planned in before the actual implementation sprints.

Cross-functionality was reported as challenging especially on the program level. The architecture and UX design was challenging to do in an "agile" fashion, as the related stakeholders were used to designing both to a minute degree well in advance (Janisse, 2016). Creating an architectural runway with only the needed decisions in place and adapting both architecture and UX as needed was not trivial.

Lastly, even though the company reported that teams successfully self-organized, the added need for guidance and coordination as teams were supposed to coordinate within an agile release train was deemed a challenge in

one case, especially as more teams were added to the train (Scaled Agile, 2015b).

Table 2.4: Adoption Challenges

Challenge	**Described by**
Staffing	Gat (2006); Holdorf (2011); McMaster (2014); SEI Global Wealth Services
Change resistance	Janisse (2016); McMaster (2014); Pitkänen (2015)
Distribution	Gat (2006); Gusch and Herbai (2015)
Coordination	Gat (2006)
Release management	Gusch and Herbai (2015)
PI planning	Weltsch-Coen (2014)
More teams need more guidance, even when self-organizing	Scaled Agile (2015b)
Test automation	McMaster (2014)
Surrounding organization	McMaster (2014)
Writing good stories due to complex architecture	Lam and Raman (2014)
Architecture and UX	Janisse (2016)
Normalizing story size	Richards (2013)
Cross-functionality	Richards (2013)

2.4.5 Benefits

As with successes versus challenges, the benefits reported to be stemming from the adoption of SAFe far outweigh the amount of drawbacks reported. In the same manner as the other factors, it is difficult to assess whether some particular benefit was achieved because of SAFe, or because of adoption of agile or the organizational change in general. As such all reported benefits have been included in this section. A summary of the benefits and their

sources can be seen in table 2.5.

The benefits are roughly grouped into three main categories: those pertaining to software, those pertaining to the organization and its people, and those related to the customer and the finances of the company. The most commonly named benefits with three or more mentions are described in their own sections, while the rest are gathered in a general section.

Software

2.4.5.1 Fewer defects

In nine organizations the adoption of SAFe reportedly led to fewer defects (Ball et al., 2015; Campbell-Pretty, 2016; Campbell-Pretty and Richards, 2013; Lam and Raman, 2014; McMaster, 2014; North, 2013; Scaled Agile, 2015a, 2016c; Vaje, 2014), and in one case it was also reported to have decreased the amount of rejected defects (Scaled Agile, 2015a). In most cases, however, it remains unclear whether "fewer defects" means less defects discovered in testing, less bugs detected in production, fewer defects reported due to changes in testing processes etc.

2.4.5.2 Improved quality

Improved quality was a benefit reported in four cases (Gat, 2006; Gusch and Herbai, 2015; Scaled Agile, 2016b; SEI Global Wealth Services), and as with fewer defects it is difficult to determine what it actually means and how it was measured, or if it was the same in each case. Since it was reported, however, it is listed here.

2.4.5.3 More frequent releases

Reported in four cases, the improved frequency of releases was seen as a benefit of SAFe adoption (Campbell-Pretty and Richards, 2013; Gat, 2006; Holdorf, 2011; Scaled Agile, 2014a). Shortened feedback loops were seen as beneficial to software quality.

2.4.5.4 Other software-related benefits

In two cases the effort required to produce builds was reportedly reduced by the adoption of SAFe (Ball et al., 2015; Gat, 2006), and in one of the cases easier software configuration management was also reported (Ball et al., 2015). Better automated test coverage was reported by two companies (Campbell-Pretty and Richards, 2013; Vaje, 2014). Decreased system stabilization time

was reported in one case, whether this was due to automated tests or not is not evident (Scaled Agile, 2014a).

Organization

2.4.5.5 Increased transparency

Increased organizational transparency was the most widely reported benefit, being listed in ten of analyzed cases (Cobb, 2012; Gusch and Herbai, 2015; Ivar Jacobson International, 2014; Janisse, 2016; Lam and Raman, 2014; McMaster, 2014; Rutzen and Roy, 2014; Scaled Agile, 2014b; Vaje, 2014; Weltsch-Coen, 2014). It was often coupled with other benefits, such as improved employee satisfaction through empowerment, improved expectation management of customers, increased trust and even improved quality, as making everyone's work visible led to everyone setting higher standards for themselves.

2.4.5.6 Improved communication and trust

Improved communication and more trust was reported as a benefit of SAFe adoption in four organizations (Cobb, 2012; McMaster, 2014; Pitkänen, 2015; Scaled Agile, 2014b). The communication was described as more open and frequent, owed in part to more trust across the organization, including between different sub-organizations—particularly IT and business (McMaster, 2014)—and between managers and team members.

2.4.5.7 Increased employee engagement and satisfaction

One of the most frequently reported benefits, an increase in employee satisfaction or engagement was named in seven cases (Campbell-Pretty, 2016; Campbell-Pretty and Richards, 2013; Cobb, 2012; Gat, 2006; Holdorf, 2011; Rutzen and Roy, 2014; Scaled Agile, 2015a). Engagement and satisfaction was often grouped together with increased communication, trust and ownership.

2.4.5.8 Increased predictability

Seven organizations reported that increases in reliability or predictability (Campbell-Pretty and Richards, 2013; Janisse, 2016; Lam and Raman, 2014; Scaled Agile, 2016c; SEI Global Wealth Services; Vaje, 2014; Weltsch-Coen, 2014) were evident after adopting SAFe. Although not strictly the same, both

predictabilty and reliability was helped by introducing cadence and shortening feedback loops, e.g. through more frequent releases (Janisse, 2016).

2.4.5.9 Increased productivity

Increased productivity was listed as a benefit in four cases (Ball et al., 2015; Gat, 2006; Pitkänen, 2015; Rutzen and Roy, 2014). Metrics employed vary or are not explicitly reported so generalizability remains low, but the cases reported an increase in e.g. feature delivery (Pitkänen, 2015) or developer efficiency (Ball et al., 2015).

2.4.5.10 Better customer and employee alignment

In three cases better alignment in the organization was reported (Cobb, 2012; Gat, 2006; Gusch and Herbai, 2015). In one case, the customer was reported as better aligned (Gat, 2006) while the other two described a change in which teams and organizational sub-units collaborated better towards common goals rather than competing. In one case this improved alignment of employees was attributed to the PI planning (Gusch and Herbai, 2015).

2.4.5.11 Increased ownership and self-coordination

In three cases an increased sense of ownership among teams and employees was reported together with improved self-coordination (Cobb, 2012; Ivar Jacobson International, 2014; Janisse, 2016). In one case, this was attributed to teams becoming empowered to solve their own problems instead of having to rely fully on middle management (Cobb, 2012).

2.4.5.12 More sustainable pace and cadence

In three companies the adoption of SAFe was deemed to have brought with it an organization-wide cadence as set by program increments for the agile release train(s), which in turn helped set a more sustainable pace for both the organization and individual teams (Ivar Jacobson International, 2014; Janisse, 2016; SEI Global Wealth Services).

2.4.5.13 Reduced costs and team sizes

While it is questionable whether this is a benefit for the development organization, three organizations reported that adoption of SAFe helped them reduce costs up to 50 %, among others by cutting team sizes (Campbell-Pretty and Richards, 2013; Scaled Agile, 2012, 2016b). In the case of Swisscom

(Scaled Agile, 2012), it was reduction in test team sizes due to test automation, and for AstraZeneca (Scaled Agile, 2016b) it was not specifically mentioned.

2.4.5.14 Other organization-related benefits

In two cases, SAFe adoption helped bring about a change in mindset in the upper echelons of the company (Cobb, 2012; Scaled Agile, 2016c). The connection between company (or product) vision and teams' daily work was strengthened, and the gap between business and IT bridged, which in turn convinced leadership that agile as a company-wide strategy was worth pursuing. Also, management focus shifted towards a longer period of time instead of just focusing on the upcoming sprints.

Better dependency management and resolution was reported explicitly in one case (Scaled Agile, 2015b). The common understanding of products and interdependencies was improved through e.g. PI planning, and also helped with customer expectation management as the chain of dependencies became clearer.

Customer & Economy

2.4.5.15 Increased customer satisfaction

In five cases, customer satisfaction increased as a result of SAFe adoption (Campbell-Pretty and Richards, 2013; North, 2013; Rutzen and Roy, 2014; Scaled Agile, 2012; SEI Global Wealth Services). While metrics vary or are unreported between companies, in one case the Net Promoter Score (NPS) increased by over 15 % compared to pre-SAFe times.

2.4.5.16 Shorter time to market

Shorter time to market or time to value was reported by six companies (Cobb, 2012; Holdorf, 2011; Janisse, 2016; Rutzen and Roy, 2014; Scaled Agile, 2012, 2016b). This is related to many of the other benefits, such as more frequent releases allowing for shorter time to market, while a better understanding of stakeholder priorities and value creation also allow for more valuable releases.

2.4.5.17 Other benefits related to customer & economy

Improved demand and expectation management was mentioned in two cases (Ball et al., 2015; Scaled Agile, 2015b). Improved demand management and traceability was reported by Accenture, while better management of customer

expectations by LEGO. Better return on investment (ROI) was mentioned in one case (Gat, 2006), but no specific metrics were provided.

Table 2.5: Adoption Benefits

Benefit	Described by
Software	
Fewer defects	Ball et al. (2015); Campbell-Pretty (2016); Campbell-Pretty and Richards (2013); Lam and Raman (2014); McMaster (2014); North (2013); Scaled Agile (2015a, 2016c); Vaje (2014)
Improved quality	Gat (2006); Gusch and Herbai (2015); Scaled Agile (2016b); SEI Global Wealth Services
More frequent releases	Campbell-Pretty and Richards (2013); Gat (2006); Holdorf (2011); Scaled Agile (2014a)
Build effort reduced	Ball et al. (2015); Gat (2006)
Better automated test coverage	Campbell-Pretty and Richards (2013); Vaje (2014)
Organization	
Increased transparency	Cobb (2012); Gusch and Herbai (2015); Ivar Jacobson International (2014); Janisse (2016); Lam and Raman (2014); McMaster (2014); Rutzen and Roy (2014); Scaled Agile (2014b); Vaje (2014); Weltsch-Coen (2014)
Improved communication and trust	Cobb (2012); McMaster (2014); Pitkänen (2015); Scaled Agile (2014b)
Increased employee engagement and satisfaction	Campbell-Pretty (2016); Campbell-Pretty and Richards (2013); Cobb (2012); Gat (2006); Holdorf (2011); Rutzen and Roy (2014); Scaled Agile (2015a)

CONTINUED ON NEXT PAGE

Benefit	Described by
Increased reliability and predictability	Campbell-Pretty and Richards (2013); Janisse (2016); Lam and Raman (2014); Scaled Agile (2016c); SEI Global Wealth Services; Vaje (2014); Weltsch-Coen (2014)
Increased productivity	Ball et al. (2015); Gat (2006); Pitkänen (2015); Rutzen and Roy (2014)
Better customer and employee alignment	Cobb (2012); Gat (2006); Gusch and Herbai (2015)
Increased team ownership and self-coordination	Cobb (2012); Ivar Jacobson International (2014); Janisse (2016)
More sustainable pace and cadence	Ivar Jacobson International (2014); Janisse (2016); SEI Global Wealth Services
Reduced costs and team sizes	Campbell-Pretty and Richards (2013); Scaled Agile (2012, 2016b)
Agile mindset in leadership	Cobb (2012); Scaled Agile (2016c)
Better dependency management and resolution	Scaled Agile (2015b)
Customer & Economy	
Increased customer satisfaction	Campbell-Pretty and Richards (2013); North (2013); Rutzen and Roy (2014); Scaled Agile (2012); SEI Global Wealth Services
Shorter time to market	Cobb (2012); Holdorf (2011); Janisse (2016); Rutzen and Roy (2014); Scaled Agile (2012, 2016b)
Improved demand and expectation management	Ball et al. (2015); Scaled Agile (2015b)
Better ROI	Gat (2006)

2.4.6 Drawbacks

The amount of reported drawbacks from adopting SAFe in the case studies is relatively low when compared to the benefits, and no single drawback stands

out as common across several cases. All reported drawbacks are listed in table 2.6.

While many cases reported increased employee satisfaction after adopting SAFe, in one case the opposite happened: employee engagement was slightly lowered as a consequence of adopting SAFe (Pitkänen, 2015). As SAFe introduced more discipline, the autonomy of teams was lowered and engagement suffered as a result.

In one case, the work load was reported to be increased, mostly for scrum masters and product owners (McMaster, 2014). In the same case, constantly changing product owners (possibly due to the increased work load) were sited to cause difficulties for continuity.

TomTom reported three separate drawbacks (Janisse, 2016). Firstly, the increase in visibility across the organization shows weak teams which can be problematic to address. Secondly, empowering teams too much can lead to chaos, as they constantly challenge architects and other experts in their fields. Thirdly, this empowering and democratization of decision-making can also contribute to the chaos when it leads to unnecessarily frequent or unbased second-guessing and pivoting.

Table 2.6: Adoption Drawbacks

Drawback	**Described by**
Lowered employee engagement	Pitkänen (2015)
More work than expected	McMaster (2014)
PO ownership continuity difficult	McMaster (2014)
Second-guessing and pivoting	Janisse (2016)
Shows weak teams	Janisse (2016)
Teams unnecessarily challenge experts and architects	Janisse (2016)

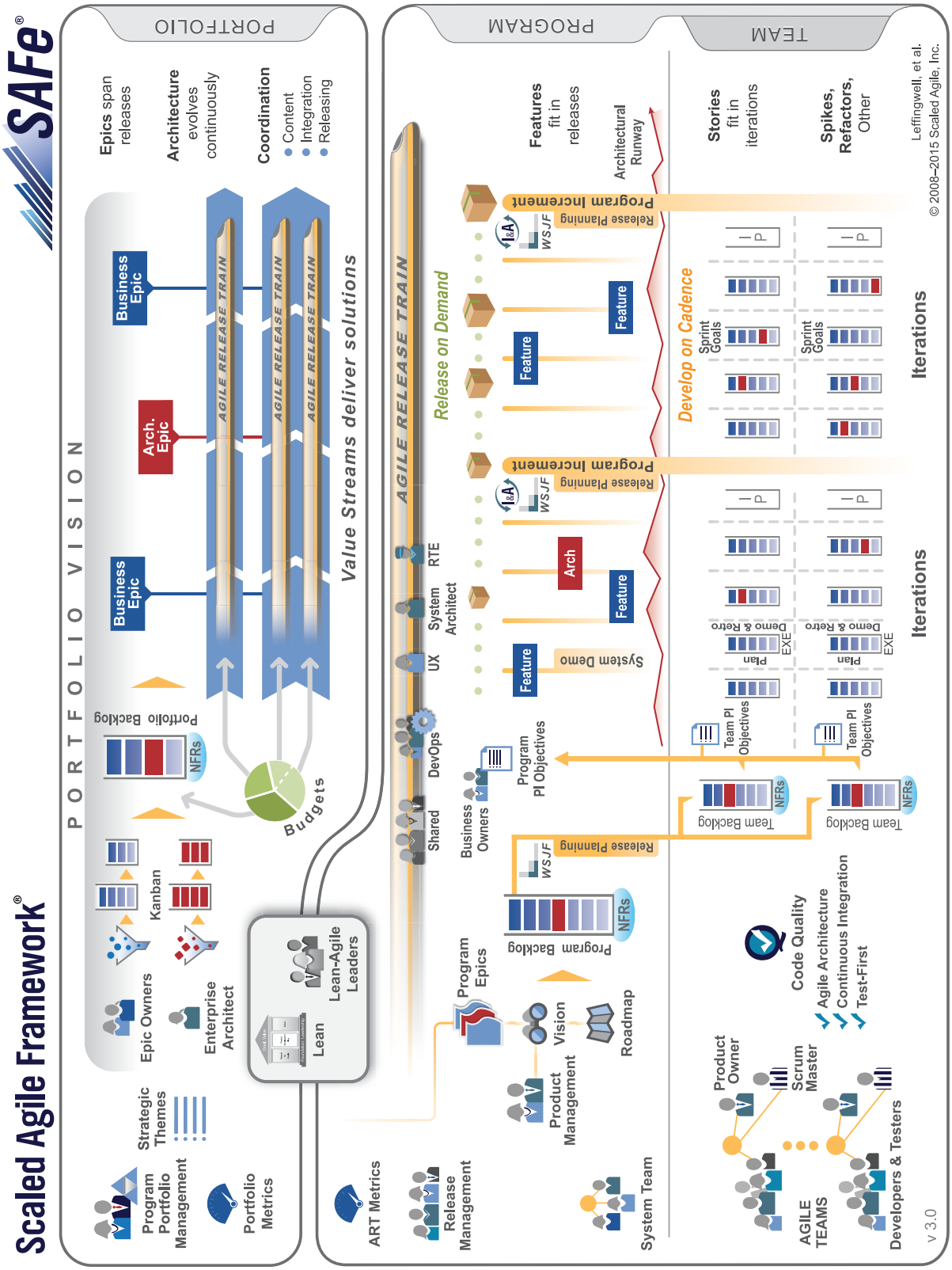

Figure 2.1: Scaled Agile Framework 3.0 — Reproduced with permission from Scaled Agile, Inc.

Chapter 3

Research Design

This chapter describes the methods and principles used for gathering and analyzing the research data. In this thesis we employ the case study method (Yin, 2009), as we mainly sought out to study a current phenomena in its own context over which we have no control, and seeking to gather in-depth descriptive knowledge about each case.

3.1 Data Collection

In order to collect the necessary data about people's experiences regarding the adoption of SAFe, a round of semi-structured interviews were conducted at the target organizations in autumn 2016. We began by contacting key personnel: leadership in the case of Comptel, and an ex-employee in the case of NAPA. After getting their approval to participate in the study we set interview dates and also asked them to suggest persons for future interviews, which we then contacted. An effort was made to interview a variety of different roles on different levels, as SAFe extends across the entire hierarchy of the organization instead of e.g. concerning just team members.

The 16 interviews, 11 of which were held at Comptel and 5 at NAPA, were conducted with one person at a time and held face-to-face for onsite employees in Helsinki and via Skype for the two offsite Comptel employees which were interviewed. Through the interviews of offsite personnel we sought to gain an understanding of how the adoption was perceived outside of Helsinki, as Comptel had a significant offshore presence. All interviews followed the same open-ended semi-structured approach employing an interview guide (Patton, 2001), in which a set list of questions was used to structure and guide the interviews with informal conversational elements allowing for deeper exploration of topics. This list can be found in appendix B. We be-

gan by introducing ourselves and the purpose of the interviews, and asked the interviewee to describe their role and history at the case organization. Next, we inquired about the structure of the surrounding organization of the interviewee—team or project size, collocation etc.—and whether the interviewee had changed roles, projects or teams during the SAFe adoption. We then asked questions related to the first research questions, trying to establish why the interviewee thought the case organization had chosen SAFe. Following that, we asked questions about how the adoption had been performed and perceived. Questions about the effects of the adoption were asked next, with which we attempted to find potential benefits and drawbacks of SAFe. We then moved on to success factors and challenges relating to the adoption, and closed by asking the interviewee for their personal opinion regarding SAFe and the adoption of it, as well as potential advice for future adoptions.

In all interviews there were two interviewers: the main interviewer and one interviewer—being the senior researcher—supporting the main interviewer with appropriate questions while also taking detailed notes. All interviews, except for one for which the audio recording had failed, were transcribed by a professional transcribing service and manually corrected and supplemented in case of uncertainties, both through re-listening to the recordings and by referring to the written notes of each interview. The detailed notes taken by the senior researcher enabled us to also analyze the interview for which the recording had failed. The case organizations of each interviewee, the SAFe layer at which each interviewee operates, the length of employment at the case organization (if known), and the date and length of each interview are summarized in table 3.1.

3.2 Data Analysis

The verbatim transcriptions of the interviews were read through and coded using the data analysis software ATLAS.ti. Passages of text from the transcriptions were assigned codes based on themes stemming from the research questions: why the organizations had chosen SAFe (RQ1), how they had done it (RQ2), what success factors (RQ3) and challenges (RQ4) the interviewees had perceived and what effects—both positive and negative—SAFe had brought with it. Text containing background information related to specific organizations were coded with their own codes, and general descriptions of e.g. agile methods employed not directly relevant to SAFe were also coded, in case queries would need to be constructed on more specific areas. Information from the existing literature was used to create a list of preliminary codes first employed when assigning codes to passages of text. However, the

Table 3.1: Interviewees

Organization	**Level**	**Employment Length**	**Interview Date**	**Length**
NAPA	Program	-	17.11.2016	111 minutes
NAPA	Program	5 years	23.11.2016	113 minutes
NAPA	Portfolio	17 years	23.11.2016	64 minutes
NAPA	Team	9 years	2.12.2016	95 minutes
NAPA	Program	13 years	16.12.2016	76 minutes
Comptel Unit 2	Portfolio	-	31.10.2016	111 minutes
Comptel Unit 2	Portfolio	22 years	31.10.2016	98 minutes
Comptel Unit 2 - off-site	Program	9 years	15.12.2016	Recording failed
Comptel Unit 2	Program	10 years	2.11.2016	128 minutes
Comptel Unit 2	Team	5 years	9.12.2016	70 minutes
Comptel Unit 1	Portfolio	10 years	25.11.2016	79 minutes
Comptel Unit 1	Program	2 years	15.11.2016	88 minutes
Comptel Unit 1 - off-site	Program	13 years	14.12.2016	119 minutes
Comptel Unit 1	Team	6 months	25.11.2016	63 minutes
Comptel Unit 1	Team	10 years	30.11.2016	92 minutes
Comptel Unit 2 & Unit 1	Team	17 years	23.11.2016	65 minutes

coding was open-ended and we employed the constant comparison method so that new passages of text were compared with the existing codes, and if no code was thought to fit the passage well then a new code was created. Alternatively an existing code was replaced if it became clear that some other way of describing the phenomenon was more accurate.

After all interviews had been analyzed the list of codes was read through and the existing codes grouped into code families. Through this grouping the list of codes became more manageable, with some codes being merged or removed as duplicates and some codes being assigned to multiple families, if some concept or phenomenon was highly relevant to all of those code families. A total of 1104 passages of text were coded, and 674 of these were grouped into 65 families—these are presented in appendix A.

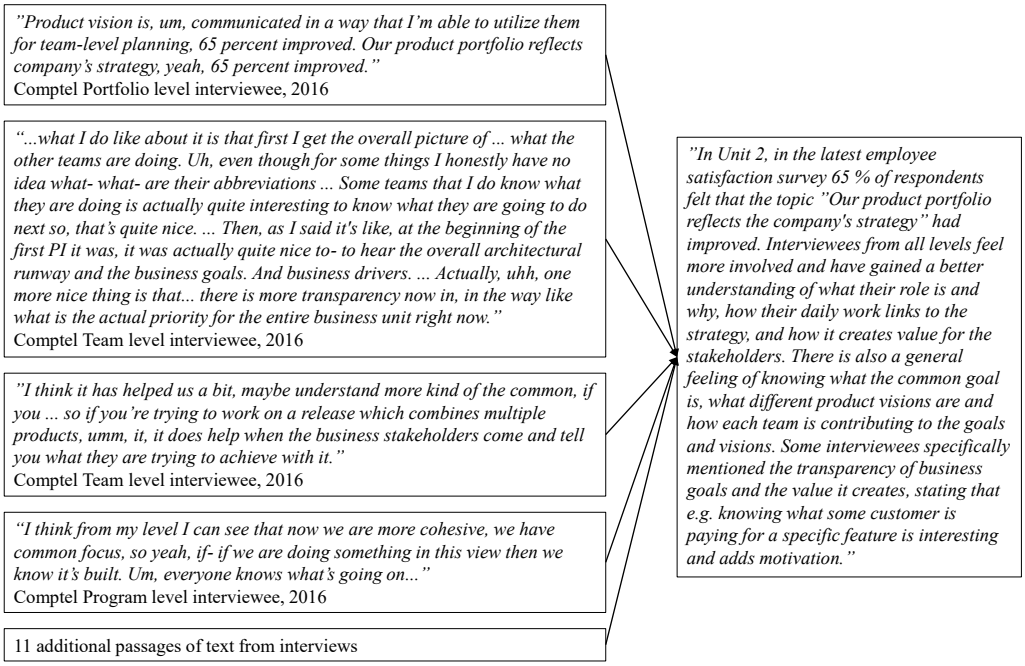

Figure 3.1: Example of the analysis process of the code family "Better alignment across the organization", the results of which can be seen on page 81.

Chapter 4

Results

We begin this chapter by introducing the case organizations in section 4.1. Next, in section 4.2 the backgrounds and reasons for each organization's adoption of SAFe is described. In the following section (4.3) we present the ways in which the organizations have adopted—and adapted—SAFe. In the next two sections, we describe success factors (section 4.4) and challenges (section 4.5) in the adoption process. Lastly, in section 4.6 we present the benefits and drawbacks that the SAFe adoptions have brought.

4.1 Case Organization Backgrounds

4.1.1 Comptel

Comptel is a global telecommunications company founded in 1986 in Finland with around 740 employees and about 30 offices worldwide. Comptel began processing data in conjunction with the first GSM call in 1991 and has since grown to process around 20 % of all mobile usage data in the world, serving over 300 customers—mostly communications service providers—with over 1,2 billion end users. The solutions provided by Comptel form the link between the operators' physical telecommunications networks and the software layer built on top of them, allowing the operators to both bill users correctly and develop more advanced services and earnings logic.

Data processing and service activation are the cornerstones of Comptel's business, and in 2015 these two parts were separated into two business units, the names of which have been anonymized in this thesis. Around 21 % of the budget is used for research and development (Comptel Oyj, 2015), and the company has expanded into data analytics and SaaS-based solutions. While Comptel maintains offices in more than 30 locations, R&D is located mainly

in Helsinki, Finland, and Kuala Lumpur, Malaysia with some small teams around Europe.

Before 2009 all development in Comptel was done in waterfall-style projects. In 2009–2010 Comptel decided to adopt Scrum throughout their R&D organization, employing external consultants to aid in the transformation by training onsite teams in Helsinki. Internal Scrum trainers then trained other offsite teams globally. While this transformation was challenging and impacted the daily work of team members, releases remained infrequent due to the business environment. Customers need significant time and effort to take a release into use, and no releases are tailored to a specific customer, so before adopting the Scaled Agile Framework a major release happened around every 9–12 months. Even though teams employed Scrum in their daily work the higher-level planning of features and releases still followed a waterfall process.

During the process of adopting Scrum each team was tasked with choosing a part-time scrum master; some teams chose to rotate the scrum master on a regular basis while others selected one team member more permanently. Each team also received a product owner, with some PO's handling multiple teams. In Comptel, the product owner role is separate into two specific types of product owners: the technical product owner—at Comptel called "product owner"—which mostly interfaces with the team, and the regular product owner—at Comptel called "product manager"—which mostly interfaces with customers. The creation of the technical PO (from here on referred to as product owner) role was done due to the work load of product managers, as some teams suffered from a lack of contact with these. There is significant distribution within teams, both across borders and sometimes within Finland.

Unit 1

The first business unit, called Unit 1 in this thesis, is comprised just over 100 persons organized into three teams in Helsinki, Finland (with a few team members residing mostly in other parts of Finland); six teams in Kuala Lumpur, Malaysia; one team in Oslo, Norway; four employees in Reading, UK; and at times a handful of employees in Sofia, Bulgaria. Team sizes vary, with some teams larger than 10 people and some people allocated to more than one team.

Unit 2

The second business unit at Comptel, called Unit 2 in this thesis, consists of about 100 employees, of which 10 in product management, with four teams in Helsinki, Finland; one team split between Helsinki and Saint Petersburg, Russia; four teams in Kuala Lumpur, Malaysia; and three teams in Sofia, Bulgaria. Most teams consists of 5–10 employees.

4.1.2 NAPA

NAPA is a Finnish software firm providing solutions for designing and operating ships. The company has over 170 people across the world, with development offices in Finland and Romania as well as subsidiaries and representation offices across the globe. NAPA's customer base resides mostly in Asia, with shipyards across Germany and in Turku, Finland, being the largest European customer locations.

NAPA produces and maintains software for use during ship design, e.g. 3D modeling, simulation of critical events, and structural stability calculation; and for optimizing ship operation, e.g. ecological efficiency, decision-making support and optimal cargo loading. The technology branch, in charge of developing the software, consists of around 70 people, of which 60 are developers organized into 12 teams. The teams are distributed, with around 15 employees in Romania and the rest in Helsinki. There was previously a development office in India with around 20 employees at the most that was shut down in October 2015.

Before 2012 the development process of NAPA was undefined at company level, with projects using a jumble of agile and waterfall, team members being allocated in up to five teams, and with no real process control in place. The code base was old and no test automation was implemented, and all development happened as projects with fixed scope releases once per year. The fixed scope was a significant challenge for NAPA, leading to late releases and poor quality as targets were missed, bugs discovered and last-minute prioritization and content changes done. According to one interviewee, bug fixes for a release could continue for years afterwards, with seemingly random software crashes being common.

At the end of 2011 and beginning of 2012 NAPA began a transformation in which all teams were moved to a strictly Scrum-based development process, with each team receiving a scrum master and product owner (some PO's handling up to three teams) and relevant training and certification being provided to said scrum masters and product owners by external consultants. Leadership had given a mandate for the change but it was up to middle

management and process owners to define how to change and then drive the change itself. Communities of Practices (COP) were set up to support the newly appointed scrum masters and product owners, and workshops were held to e.g. create common Definitions of Done. The transformation was driven by the middle management of the technology unit, seeing significant issues in the current development organization and also viewing the move towards agile as necessary based on the general software industry trends.

In 2013 the owner of the software development and release processes, and also quality lead at the time, discovered SAFe and began employing principles from the framework in order to scale agile across the development organization. The first joint release planning was held, and the development organization moved from a roughly yearly release cadency to one of three months. Even though the idea was to move away from fixed scope releases, with features being either accepted or rejected at the time of release, getting it to work proved challenging and releases were stretched.

The principles of the Agile Release Train (ART) were adopted during 2013, and in 2014 the first trainings were held for the drivers of change at that time. Also, from 2014 onwards new scrum masters and product owners were being given SAFe trainings and certification. At no point was SAFe declared to be in use, but the technology unit considers 2014 as the first "official" year of using SAFe.

4.2 Rationale for SAFe Adoption

In this section, we present the results related to the first research question of this thesis:

RQ1: Why have the case organizations chosen to adopt SAFe?

The reasons and rationale given for the adoption of SAFe varied significantly between interviews, and as with the literature reviewed in chapter 2 discerning the actual reasons is not trivial. Common for both companies (and both business units within Comptel) is that Scrum had been previously adopted at team level, so most reasons revolve around the lack of similar agility in other parts of the organization or a lack of purposeful, supportive structure in organizational layers above the team. The reasons have been categorized into five topics—with some overlap existing—and are listed in table 4.2. Due to the overlap it is difficult to discern the most prominent reason: better dependency management and improved visibility is connected to better reactivity, all of which depend on better long-term decision-making. Contributing to

this difficulty is that no single theme emerged as the most important during the interviews. Each reason is described in its own subsection.

4.2.1 Wanted capability for more frequent releases

Comptel

Previous to SAFe, every product had individual teams working on them with releases around once every 9–12 months. There was a need to move from single products to more integrated solutions built out of parts of several products, but every time more than one team was needed to work on the same product(s) it was a special process with severe challenges for synchronization and management of the work. The fact that global software development was proving challenging was not helping, and PO's and PM's would fight over resources and not necessarily allow team members to help other projects even though the priority might have been higher. One interviewed leader summarized it as:

> *"It was kinda "good luck and godspeed" the whole thing. And pretty often our releases were heavily delayed. And then sometimes, what we were aiming for was of course not the same as what we got in the end, which is quite OK and normal."*

As each different site and team had varying practices, such as sprint lengths and demo times, progress tracking across teams and locations was difficult for managers. A need for focus and for a common goal was perceived at least by managers, and the practice of releasing seldom meant it was difficult to know which teams did what at which times. As such, timely integration and testing was proving difficult, which contributed to the often late releases. One perceived benefit of SAFe was a unified scale for estimation across teams, which would help increase predictability and realism, while making it easier to both track and balance the work load of teams.

Another perceived benefit was the increase in reactivity to changes, especially changes to customer priorities. Content was planned for nine months ahead and changing it was very challenging or sometimes impossible. Moving product management to a more agile process while reducing or removing the amount of "big upfront planning" was one goal of the adoption, and one interviewed manager stated:

> *"...the moment we are able to react quickly to the market, and we are able to kind of respond and release on demand I think that would be kind of the holy grail we are seeking for."*

NAPA

All software development in NAPA was previously done in projects, to which developers were resourced in varying degrees. Releases were done once a year, with fixed scope and fixed release dates, but the release dates never held and were sometimes up to six months late. After the releases bug fixes could take half a year, and if some feature was dropped it could take the customer up to 16 months to receive it due to the inflexibility of long-term plans and scope.

The large batch size was causing problems not only in quality but also in developer satisfaction, as the stress level towards the release date increased exponentially. Some employees at NAPA saw that many of the products interfaced with each other, and while Scrum was proving to be successful at team level it did not address larger issues such as integration across products or common architecture development.

In order to address this chaos and lack of flexibility NAPA turned to SAFe, with the main themes pursued with the adoption being quality and internal cadence. Moving towards continuous delivery with an even release cadence was seen as a natural step from project-based development, especially since most development work was done on existing products. The release cadence would help create a more sustainable pace with less peaks of stress, which in turn would help quality.

The improvement in quality together with the more frequent releases (of more flexible scope) brought on by SAFe would, it was thought, help deliver better value to customers at a more rapid pace. At the same time, organizing all teams into a single Agile Release Train was seen as essential to achieve the coordination and transparency across all sites needed to achieve the goal.

4.2.2 Lack of a portfolio level

Comptel

Before SAFe every feature went through the product managers in a strictly waterfall-based process which lacked systematicness. The company had a strategy but product managers were single-handedly deciding how to best meet the strategic objectives, as only they knew about current industry trends and relevant themes—which they employed as they saw fit in their roadmaps. Teams had to rely on product managers to spend time explaining these roadmaps and themes to team members, which did not always happen. Communication between product managers and POs was informal and sometimes infrequent, and the only input a PO gave during the creation of a

roadmap was how much of the items desired by the product manager would fit. As such, the roadmaps produced by the product managers were large and relatively fixed in scope, meaning that reacting to changes—in the market or in customer requirements—was not easy and almost always meant some other item had to be dropped.

In essence, all knowledge and decision-making related to the long-term goals of the organization was tied to single product managers, with each one handling the process differently. The "silos" also made resource sharing difficult or even impossible. Business unit level prioritization was non-existent, and customers were often disappointed in late releases or releases not containing every promised feature due to the inflexibility of one-year plans mentioned above. Product-tied roadmaps were prioritized above integrated releases employing many parts of different products. The portfolio layer of SAFe was seen as a potential solution to these challenges, with a clear process for high-level ideation and prioritization of requirements which focused on the strategic objectives of the organization.

The regular synchronization and planning sessions which SAFe would have brought were seen as a potential remedy to the lack of milestones and work management in the long release plans originally descended from the relatively rigid business environment. Making sure everyone understands what the common goals and priorities are in order to enable collaboration towards these common goals was seen as another potential benefit of SAFe. Through this shared understanding and systematic process Comptel wanted to be able to produce more multifaceted releases with more integration between different products, in turn creating more value for the customers. This was in line with the recent trends in the telecom industry, with customer requirements moving from product point solutions towards system level solutions.

NAPA

While the team level of NAPA was agile, managing ideas and the flow of them from idea to implementation was challenging. Based on this, the portfolio level of SAFe was seen as something which would both fit well into the organization and which would help create a systematic approach for managing ideas and decisions related to them. As there are more ideas than what is feasible to implement at any one time, NAPA needed the portfolio level to help focus on what was relevant to the strategic objectives of the organization. The portfolio level as defined by SAFe was seen as the most concrete aspect which SAFe would bring, as relatively few changes would have to be done to release or team processes.

4.2.3 Lack of dependency management

Dependency management overlaps significantly with the portfolio level, but was mentioned to such an extent in Comptel interviews that it was separated into its own reason. A dependency exists when the team cannot develop the complete system entirely by themselves, e.g. requirements, expertise, technical solutions, funding etc. are needed from a stakeholder (or several) external to the team. Dependencies can be one-, bi- or multidirectional, depending on the complexity of the solution and the amount of teams and persons working on it.

Comptel

The Scrum adoption was challenging but ultimately rather successful on a team level. Dependency management between teams remained challenging, which also affected the teams' understanding of the "bigger picture". Product managers were not oblivious to these challenges, and came to see SAFe as a solution for the lack of collaboration in dependency management.

As releases were planned far ahead with no real dependency management, some dependencies were not discovered until very close to the actual release. At that point, the team which could address the dependency might already have received their roadmap until the next release, leading to—in the worst case scenarios—features being delayed for well over a year. As one interviewee puts it:

> *"... and then they suddenly realize that, hey, we've completely forgotten this part, this needs to be added here. Same problem again. Our grandiose plans fall apart completely. Which is totally natural, of course."*

SAFe was deemed to bring several points of improvement: the PI planning visualizes dependencies, and the portfolio level allows for earlier resolution and better management of dependencies. The program level would help in synchronizing and aligning teams, especially between teams from different sites, which was something Comptel was struggling with at the time.

4.2.4 Lack of a supportive structure above team level

Comptel

During the major transformation in 2009–2010 all teams in R&D had moved to Scrum, developing their specific product in sprints but according to the waterfall-type fixed-scope release plans. The managers had a commitment to

the customers to get everything in the plan released on time, but even though teams had been doing Scrum for well over five years there was significant variation between them, especially in the way teams estimated. As some teams implemented stories that were part of the same product this difference in both estimation and velocity provided an additional challenge when trying to synchronize work. This lack of cohesion was a challenge that SAFe was seen to address, and managers wanted to enable teams to work as a unit through SAFe.

NAPA

A general lack of defined processes for software development was proving challenging, as every developer was an expert in their own right, delivering whatever they could in the scope of time defined for the release. For the major release once per year the progress was attempted to be bundled into a release for the customer, which was then improved with bug fixes and the like in the following minor release. The need for something more systematic than "cowboy coding" was somewhat urgent, as people could be assigned to five different teams within the same project based on the current needs and their own expertise.

After the adoption of Scrum as a team level practice (but before the adoption of SAFe principles) many of these problems were addressed, but for instance the role of product owner remained challenging, and varied from person to person. Some spent most of their time interfacing with the customer, effectively leaving the team without input, or interfaced mostly with the team, leading to a lack of customer focus. The program level of SAFe was thought to mitigate this challenge, by adding a layer of support to teams and to the scrum masters and product owners.

Scrum was thought to be too narrow, addressing only the actual programming and development of products, and not how they interface with each other or with a common architecture, how they should be integrated etc. SAFe was seen as a natural extension of Scrum, helping to address the aspects which Scrum did not without interfering too much with the already established daily work practices of the teams.

4.2.5 Lack of transparency

Comptel

A common complaint on team level before the adoption of SAFe was that developers did not have any real understanding of the bigger picture. Scrum

was working relatively well in each team, but team members had difficulty seeing how their daily work linked to, and affected other parts of, the organization. The issue was compounded by the waterfall-style roadmap planning of product managers, which teams rarely saw other than when receiving the product plan for the next 3–4 months.

There was a clear need for some sort of structure in which team members would be encouraged to share and synchronize with each other as well as across team and national borders. This need ties into many of the other reasons for adopting SAFe, as the desired increase in transparency was also thought to improve dependency management and help create a better understanding of how everyone's work helped create value.

SAFe was adopted not only to improve transparency inside of and between development teams, but also to improve visibility across different organizational units. Having better visibility between technology and business units would allow for a better shared understanding of customer needs and how to best address them while also helping foster an understanding of how different organizational units operated.

NAPA

NAPA was facing challenges with the cooperation and information sharing within the many different sites and teams in its global organization. Planning was done behind closed doors by a small group of higher-ups, teams or even developers working on the same products were poorly synchronized, and there was no real shared understanding of e.g. a product's vision across organizational layers or borders.

SAFe was seen to address these challenges via the release planning, in which everyone is allowed to give their input and ask questions, and via the general increase in visibility in upper organizational layers. The increase in visibility was also thought to help motivate people, the idea being that having to show your plans or priorities or code publicly—no matter what your position—would motivate everyone to bring their best.

4.2.6 SAFe seen as easy to adopt

One topic frequently brought up during the interviews was that SAFe seemed to fit well into the organization. While this in itself is not a reason for selection of SAFe, it seems to indicate that Scaled Agile (the organization behind the Scaled Agile Framework) has managed to create a framework that organizations see as easily adoptable. There emerged two themes supporting this: information about SAFe is plentiful and publicly available, and SAFe

is seen as relatively easy to "fit" into the existing organization without too much disruption.

In the case of Comptel, a supervisor of an interviewed manager had taken a course and brought the framework to the manager's attention. The amount of publicly available information, including case studies with positive outcome, helped cement the opinion that SAFe was a proven method for scaling agile software development.

At NAPA, Dean Leffingwell's book *Agile Software Requirements* (2011) was used as a source of inspiration and ideas for the transformation, which then turned into a more complete adoption of SAFe as both the team level and program level was deemed to be in use but the portfolio level sorely needed. Since existing parts of the organization fit into the framework, the decision to adopt SAFe was not seen as difficult.

4.3 Approaches

In this section, we present the results related to the second research question of this thesis:

RQ2: How have the case organizations adopted SAFe?

For NAPA, SAFe was first internally discovered and then gradually adopted over several years, with no real threshold existing where the organization could be said to have begun adopting SAFe. In Comptel's both business unit, on the other hand, adoption of SAFe was performed rather swiftly and comprehensively. The table 4.3 summarizes the different aspects of the approaches organizations used.

4.3.1 Main approach

Comptel

In both business units' R&D organizations the adoption was done all at once, in a big bang approach instead of piloting it in smaller teams. The adoption was planned and then a joint first PI planning was held in which the release train was launched: in Unit 1 in June 2015 and in Unit 2 in January 2016. In both organizations the initial drive to adopt SAFe came from product management and leadership. The adoption only concerned the R&D sub-organizations, with other parts of the organizations remaining outside of SAFe.

Table 4.2: Rationale for SAFe adoption

Reason	**Description**
Wanted capability for more frequent releases	Long release cycles led to releases of poor quality and poor customer satisfaction at all case organizations as the waiting time was very long.
Lack of a portfolio level	High level prioritization and management of work at Comptel was rigid and person-dependent, leading to a poor ability to react to changes.
Lack of dependency management	At Comptel, handling dependencies between teams was challenging and made worse by the long development cycles, leading to last minute crunches, with increased stress and decreased quality as the end result.
Lack of supportive structure above team level	The lack of cohesive work practices in e.g. estimation at Comptel, and the lack of any program level coordination and planning with multiple teams at NAPA, led to poor ability to have multiple teams work on the same items among the case organizations.
Lack of transparency	Teams were not aware of the reasons behind prioritization decisions, managers did not involve teams in planning and communication between and within teams was limited at both NAPA and Comptel.
SAFe seen as easy to adopt	SAFe was deemed a good fit to the organization and there was a lot of publicly available information and case reports, which—although not reasons—helped all case organizations select SAFe specifically.

In Unit 1, a consultant was employed to help make the decision about which scaling framework to choose. The adoption was ramped up in under two months, and one interviewee mentioned that while the team and program levels were adopted directly, building a working portfolio level took time, and more focus was put to synchronizing teams and aligning the organization in the beginning. In Unit 2, on the other hand, focus was more on portfolio management—training relevant personnel, changing portfolio management processes and structures—than on teams.

NAPA

In NAPA, the adoption was more gradual and bottom-up than at Comptel, happening over several years with different principles and ideas of SAFe being adopted at different times, starting with the release (PI) planning in 2014. The driver of the adoption came from the program level, having been in the team level previously. The team and program levels were built first, and a functioning portfolio level was described as somewhat new by one interviewee. The adoption stemmed from product development and later involved product management, but HR and legal etc. are still separate sub-organizations with their own processes.

4.3.2 Introduction and training

Comptel

During the beginning of the adoption at Unit 1—before the first PI planning—external consultants were hired to go over the basic principles with product management, leadership and the chief architect. Team members had an internal two-day introduction held by the RTE and an old product owner but no formal training. After around three months of beginning the adoption the product managers and product owners were trained in two-day external trainings and certified. Six months into the adoption, consultants were hired to do an audit of the adoption: how well were practices working, how had the adoption evolved, how was planning and management currently being done, etc.

In Unit 2, everyone was introduced to SAFe in some way before the adoption began. Product owners and product managers, including those from sites other than Helsinki, received intensive training from external consultants, which organized a set of workshops and trainings, one of which being "Leading SAFe", during four days in the autumn of 2015. During the workshops the groundwork was laid for the adoption, among other things the

first epics and features created and dependencies listed. In addition, product owners took a two-day SAFe PO training and certification shortly before the first PI planning. Teams were introduced to the adoption by in-house management in shorter sessions of around three hours per team with a follow-up Q&A session a few days after that, but with no formal training or certification. One interviewee stated that since the adoption did not affect teams very much, it was easier and faster to discuss only the relevant and Comptel-specific aspects with them. The RTE held "refresher" sessions with teams after the first few PIs during the autumn of 2016, and was later certified as a SAFe program consultant, which enables training teams and new employees.

NAPA

The main drivers of change, at that time the quality lead and software development process owner, first took the SAFe Agilist training (held by consultants). Consultants were hired whenever needed during the first two years. After having adopted some SAFe principles and seeing that that was where the organization was headed, from 2014 onwards all scrum masters were trained and certified as SAFe practitioners by external consultants. Team members did not receive any training.

4.3.3 Communication about SAFe

Comptel

One offsite interviewee stated that SAFe was not specifically mentioned in the beginning at Unit 1, instead there was talk about how they were scaling their Scrum. Another onsite interviewee mentioned that communication to teams was somewhat haphazard, with some emails sent and managers holding short sessions with teams. A team level interviewee supported this, stating that the potential benefits of the adoption, such as improved dependency management and better visibility, were explained but the rationale of the adoption of SAFe was not. There are now designated change agents with SAFe certification in all locations, with plans to have all newcomers received certification directly and to have an "evangelist" group organizing these and other agile-related events.

In Unit 2, interviewees did not report the same aspects so it might be feasible to assume that communication about SAFe was more uniform.

NAPA

At no point in the adoption of SAFe principles was there any clear communication to the whole organization about it. As the adoption was gradual, and aspects and ideas of SAFe gradually taken into use, SAFe was not explicitly mentioned. According to one interviewee *"...people here do not necessarily understand that they are doing SAFe..."*

SAFe is only recently being mentioned, but not consistently or very loudly, according to another interviewee. The interviewee had trouble recalling when having first heard about SAFe, and felt that at least on team level the adoption was still not visible, but had heard it mentioned when discussing the naming of a solution level architect. A third interviewee had no recollection of attending any meetings where SAFe would have been explicitly mentioned as being used.

4.3.4 Organization and cadence

Comptel

In Unit 2 the dependency matrix was a deciding factor in choosing to organize all teams in a single release train. Following the recommendation of a consultant, the RTE was recruited internally (from a previous product owner) in Helsinki, and was made a full-time role with the blessing of leadership. A consultant helped mentor and coach the new RTE, helped prepare and facilitate the first PI planning, and provided additional support during the first PI. The cadence of the PI is ten weeks, and all teams sprints are synchronized to start and end simultaneously.

In Unit 1, the RTE is in Kuala Lumpur but has frequently traveled to Helsinki, among other things on a SAFe "roadshow" to explain to teams why SAFe is being done. The cadence of the PI varies but is usually ten or twelve weeks, and all teams sprints are synchronized to start and end simultaneously.

In both business units, the releases happen according to a different plan than the cadence, mostly due to restrictions in the business environment as customers cannot take releases into use as frequently—as such, releases are still done every 9–12 months.

NAPA

The role of RTE at NAPA is somewhat unclear, as the original drivers of change were process owners and one other current driver is an agile coach. The current responsibility of improving the release and software development

processes still reside with the process owners even though the role of RTE has been introduced.

All teams except for continuous delivery teams are in the same release train, and the rationale for this was to have synchronized planning and release of all the different products. Teams decide their own sprint length, both due to logistical reasons (as product owners cannot attend all sessions if they have multiple teams) and to allow for customization. Sprint length is also flexible depending on the context.

NAPA employs an in-house agile coach. Scrum masters are recruited from teams and handle their responsibilities part-time.

4.3.5 PI planning

Comptel

The first PI planning at both Unit 1 and Unit 2 was done according to the SAFe handbook, and has later been customized based on feedback. For the first planning in Unit 2 the program and portfolio level items were defined by the product managers together with product owners and architects. The plannings of both business units take place during two days, and jointly across all sites with offsite members participating via Skype. One manager cites time differences between sites as *"very challenging"*, but also summarizes:

> *"Well uh, in Kuala Lumpur there's a five-hour time difference, so they sit there with the help of pizza."*

Sessions begin at 8:30 Finnish time with an introduction by the leader of the business unit presenting the vision and strategy. Next, product managers present their own roadmaps of upcoming functionality, tell about current client requirements and the portfolio items which have been formed based on these. About half of the first day is used for the introduction, and afterwards teams go into their own meeting rooms and plan together while stakeholders and product managers go from room to room. Scrum masters and product owners synchronize regularly in order to solve dependencies as quickly as possible. The items of the first sprint are also groomed as much as possible.

The second day begins with product managers reviewing the status of the current plans, to see if any prioritization needs to be changed or if some item planned further. Next, the final adjustments to plans are done, and in the evening the plans are reviewed together and a confidence vote is held in which everyone give their opinion of how well the PI will go. Both morning sessions and the review and vote are held in a joint workspace with everyone onsite participating.

In Unit 1, teams plan together with the PO, and the POs in turn have formed some preliminary plan before the session. One interviewee estimates that the length has been reduced from roughly fifteen to around ten hours total, mainly by eliminating unnecessary reviews and meetings from the schedule.

In Unit 2, the agenda has been loosened and specific meetings, e.g. scrum of scrums, have been removed. It is the scrum master's responsibility to organize ad hoc meetings whenever needed. There is a site-specific retrospective in which the entire business unit participates after each planning, the content of which was left unspecified in the interviews. This is held in the second evening of the planning in European sites and the following morning in Asian sites.

In Unit 2, the dependency management in the PI planning was done via a physical dependency matrix, with the different components of all products listed on a wall and the dependencies between them drawn. At Unit 1, a list of risks is presented at the end of each planning.

NAPA

The PI planning, or release planning as it is called at NAPA, was initially two full days as prescribed by SAFe. This was compressed to one day later on, and currently the planning is held during two mornings (Finnish time), so that time differences between sites do not hinder participation.

The planning is a joint session across all sites, with Skype being the main tool for other sites to participate. In Finland, everyone is free to participate in person, but the attendance rate has declined since the first planning with employees choosing to participate via Skype instead. All in all, one interviewee estimated that 130-140 employees of the around 200 in NAPA participate.

First, an introduction is held by senior leadership, e.g. the VP, telling everyone where the company is headed during the next three months and explaining the current situation of the industry. Next, the agenda is described, after which product managers present their roadmaps. Teams then make a draft plan for the next increment based on the roadmaps, which is reviewed with everyone. The plans are iteratively refined and dependencies and risks continuously resolved during the first day. During the second day, plans are presented and final adjustments made, and at the end a vote of confidence is held in which everyone can express their confidence in achieving the goals of the increment. Later on, the draft plan stage was skipped since teams and product owners began proactively forming the draft plans together with stakeholders before the actual planning session. One interviewee states that

it has since been further refined:

> *"After that we have moved to the release planning becoming more of a sort of plan presentation day, so everyone has done the real planning also with the teams and between the teams, and they attempt to do that before the session, well in advance."*

During the planning discussions and plans happen on a high level: stories are not planned or discussed, and no estimation of features happen. One employee described it as teams planning on their gut feeling.

There is no centralized dependency management, instead each team and PO is responsible for handling their own dependencies and risks.

4.3.6 Cross-functional month

NAPA

At NAPA, there is no separate IP sprint. Instead, teams are expected to do IP work alongside their regular work, and allocate enough time during planning to allow for it. One interviewee described that many teams and POs employ have one so-called rogue day in the sprint, in which team members can work on anything they want as long as it might somehow be beneficial to the team.

On the program level, the release cycle has been adjusted to releasing every four months, with the last month being a so-called "cross-functional month". During this time, the release is stabilized via testing and bug fixing, installation instructions are updated and the next release is planned, so this period also contains activities which are nominally assigned to the IP sprint. Release demos and hands-on and online workshops are held where everyone can familiarize themselves with the content of the release. This allows sales personnel to better market the content of the releases and create relevant training material. One interviewee added that while marketing and training material could be updated continuously when working on the content of the release, it is very difficult since things constantly change. It would require sales personnel to be constantly and closely involved with the development, which according to one leadership interviewee is not feasible at this point.

4.3.7 Customization

Comptel

Practices of SAFe have been modified throughout the adoption to better suit the organization in both Unit 1 and Unit 2. One interviewee from Unit 2

mentioned that most changes had been done to demos, but that also scrum of scrums and PI planning agendas had been customized. Teams are allowed to choose their own ways of working within a certain frame, as sprint length is set in both units. In Unit 2 teams can choose whether to use Scrum, Kanban, or something else, while in Unit 1 Scrum is mandated.

In both Unit 1 and Unit 2 the RTE created a "handbook" for SAFe in-house before the first PI, in which SAFe principles and terms are explained together with Comptel-specific information. It also contains practicalities for teams, such as different time zones. Drivers of the change in Unit 2 also used the knowledge gained in Unit 1 about their adoption and its challenges to help their own. However, the handbook does not contain a formalized vocabulary of all SAFe-related terms and their specific meaning at Comptel.

NAPA

Teams are allowed to customize their ways of working, being free to select the methodology and which metrics to employ. Retrospectives are mandatory, and every team has to have some sort of planning, but organizing these is up to the teams.

NAPA employs a chief portfolio officer, which they perceive as a release train engineer for the portfolio level, to guide the portfolio level and ensure smooth flow of ideas through the funnel.

4.3.8 Demo

Comptel

At Unit 2 there is a 4–5 hour long joint system demo, which is open for everyone, at the end of each PI where teams present their work in half-hour slots. The level of the demo is higher than sprint demos, as instead of code or test reports the integrated functioning system is shown.

In Unit 1 there is no such demo at the end of the PI, but there is a combined sprint demo: at the end of each sprint there is a joint session in which every team has a time slot assigned of around 20 minutes. Teams do not have to participate or demo, and the material of the demo is sent to all participants. A video is also recorded of the presentations and made available to everyone.

NAPA

There is no single joint system demo at NAPA, but there are four sessions in which teams present their work in smaller groups. Everyone is invited to each

sessions and they are recorded. There is a release team which creates a video which acts as a system demo, and this video is posted to the internal Yammer (a social networking services for enterprises) for comments and feedback. This system is also in use at team level, in which instead of public demos teams create short videos—starting from three minutes—and post them to Yammer for everyone in the organization to see.

4.3.9 Estimation

Comptel

In both business units one big change was normalizing teams' story points during the first PI planning. Previous to the planning organizers had asked teams to make a one story point story for reference: the team should be able to finish that story in one day. This was then used as a common base when estimating.

NAPA

As release planning happens on a very high level, no estimation is done at that time. Most teams use average velocities when planning for the sprint, but some do not estimate at all. According to one interviewee, this does not change the end result: the average number of stories done during the sprint remains relatively constant. The interviewee thought that it might be due to the maturity of the team, as in other teams which tried to skip estimation the end result was poor.

4.3.10 Metrics and follow-up

Comptel

An interviewee from Unit 2 mentioned that in both business units a specific set of key performance indicators (KPI) was taken into use for the adoption of SAFe: epics achieved from planned content shows how many items were completed and how many spilled to future PIs; total story point change describes how many story points were planned and how much unplanned work was added (with support work subtracted); number of changed epics described how many epics were either modified, removed or moved out of scope; and average cycle time shows the average time for a backlog item to move from being added to the backlog to being completed.

However, one interviewee from Unit 1 did not know about any specific metrics, and according to another these high-level metrics were the only ones used:

> *"Well we thought about those [metrics] too, so originally we had a lots of ideas about how we were gonna put up a huge amount of KPI's and statistics and whatnot. And I feel that it's more like, it sort of went towards just looking at what we actually get out, as like throughput. Looking only at completed epics and stories and trying to kind of leave everything else out of that, so ... if you ask me for example what the automated test coverage is, then we do not have a metric for that currently. Otherwise I'd probably know it right away."*

In addition to the metrics, teams follow standard Scrum practices. On the program level, Scrum of Scrums are held every week, while product owners have bi-weekly meetings. The Scrum of Scrums of Unit 1 is a bit different in that it also involves all product managers and product owners also, meaning that it is both a status and problem-solving session. Offsite scrum masters from both business units hold joint local round-tables and brownbag sessions every month. On the portfolio level a portfolio board meeting is held weekly, going through if there are critical conflicts or prioritization changes and how to respond by e.g. shifting work loads from one team to another.

NAPA

Metrics were initially introduced together with Scrum, for test coverage, team velocity, forecast hit rate (for how well sprint plans hold together) and cycle time (the total amount of time for issues to get Done). Together with the gradual adoption of SAFe a set of rewards tied to metrics were introduced at NAPA at a relatively early stage, around 2013–2014. These were set for the whole organization (tied to revenue), units (tied to e.g. automated test coverage) and teams (with more specific targets).

Table 4.3: Approaches to SAFe adoption

Adoption approach	**Description**
Main approach	The adoption at NAPA was a gradual process driven by key employees over several years trying out different ideas and practices. In Comptel, leadership and upper management spearheaded the adoption, which itself was quite quick with a ramp-up of two months in Unit 1 and around double that in Unit 2.
Introduction and training	All of the case organizations employed consultants. Product owners, middle management and leadership were trained prior to the adoption in NAPA and Unit 2 and after in Unit 1. Teams received a shorter in-house contextual introduction, but no formal training.
Communication about SAFe	In NAPA SAFe was almost never explicitly mentioned, and is only recently being mentioned in internal and external communication. In Unit 1, SAFe was likewise not mentioned much in the beginning and communication somewhat haphazard. In Unit 2, SAFe and the benefits of it were clearly explained but the rationale behind the adoption not.
Organization and cadence	In all case organizations the whole organization is arranged into a single agile release train. In Comptel, sprint length is set and releases follow a different cadence than the train, and the increments are 10–12 weeks long. At NAPA, the sprint length is flexible and releases happen after every increment, which are four months long.
PI Planning	All case organizations began with the recommended two-day agenda, and have since customized their planning sessions to e.g. take into account time differences. All follow roughly the basic structure, and are held across sites simultaneously, with participants gathering in workspaces or participating from their computer.

CONTINUED ON NEXT PAGE

Adoption approach	Description
Cross-functional month	At NAPA there is no IP sprint, instead one month is used to both stabilize the release and demo it to the whole organization, enabling sales personnel to better market the products and everyone to get an understanding of what functionality was completed.
Customization	SAFe practices have been customized in all organizations, with Comptel creating an in-house handbook for SAFe in their context. At NAPA, teams are free to select which methodology they follow but retrospectives are mandatory.
Demo	In Unit 2 there is a joint system demo held at the end of the PI. In Unit 1 there is no such demo, but sprint demos are held jointly. At NAPA there is no single demo, instead workshops and sessions are organized where everyone can familiarize themselves with the products.
Estimation	At Comptel, story points were normalized across teams to help with estimation and predictability. At NAPA, no systematic estimation is done at program level, with teams selecting features according to their own understanding.
Metrics & Follow-up	At Comptel, metrics for the adoption were developed and taken into use. At NAPA, metrics were introduced with the adoption of Scrum and enhanced with incentives when SAFe was being taken into use.

4.4 Success factors

In this section, we present the results related to the third research question of this thesis:

> *RQ3: What common success factors for adopting SAFe can be identified?*

Success factors are listed in table 4.4.

4.4.1 PI planning

Comptel

The first PI planning at Unit 2 was feared to be total chaos, with no-one knowing what to do and with no planning getting done—potentially leading to failure right at the beginning. However, as one interviewed manager puts it, the planning was a controlled chaos which went surprisingly well, leading to some astonishment amongst the team members:

> *"...at the beginning of the of the PI planning our Release Train Engineer kinda promised it to be kinda, ad hoc and chaotic, because it is a first PI but, you know like, at the end some people complained that there was no chaos, like they were looking for it, [but] everything went smoothly."*

The presented high level goals and business needs were appreciated by team members, some of which felt that they were seeing and understanding them for the first time. Organizers were also keen on noticing aspects which were not functioning well in order to tailor them better to Comptel for the following PI planning. In interviews by both Unit 2 and Unit 1, the first PI planning was described as a big success even if Unit 1's planning was described as more chaotic, especially when considering the short timeframe in which the adoption was made.

One manager at Unit 2 mentioned that dependency management worked *"right out-of-the-box"* during the first PI planning. Dependencies and risks were being listed—and solutions thought out—from the start, not requiring any special guidance or warm-up time (as managers had feared).

One more mundane aspect mentioned was the breakfast served at the beginning of each PI planning day in Finland, and the dinner served at the end of each PI planning in Kuala Lumpur (due to time zone differences, PI plannings could go on quite late). Though not officially mandated by SAFe, team members perceived the PI plannings more positively when they perceived the company was taking care of them. One manager stated that all logistical matters, including room bookings, IT equipment and connections in addition to provided food and refreshments, must be in order before an organization-wide PI planning can take place.

NAPA

The first PI planning went "surprisingly" well, with everyone having prepared themselves and their respective responsibilities well. While already described as a success the first time, the process of grooming backlog items previous

to the PI planning (or Release Planning as it's called at NAPA) has been continuously refined, with each team and PO having their own practices and processes for doing so. The main benefit of doing the majority of the grooming and actual planning before the session is that dependencies and risks are identified much earlier, requiring less time during the session itself and also reducing the amount of last-minute prioritization changes needed due to such problems. It has also reduced the amount of time needed to plan altogether.

The PI planning are held across sites at the same time, and one interviewed leader described having a joint simultaneous PI planning as *"an absurdly important thing."*

4.4.2 Consultants & training

Comptel

Using consultants to go through the potential adoption, determine how it might be adapted to the organization, and arranging practicalities was deemed as a prerequisite for successful adoption by one interviewed leader. Having a good coach to train program layer roles, and team roles interfacing with the program layers, in this case product managers and product owners, was mentioned as an important success factor. Off-site product owners were also brought to Helsinki for the training, enabling them to help the adoption at their respective sites. One manager stated that simply having someone to answer questions regarding SAFe and how to implement it was seen as helpful.

Training for teams was held internally, as the effects of adoption on team members and processes were seen as relatively small. The RTE later held SAFe refresher trainings of 2,5–4 hours for teams, going to each site to talk about the initial goals of the SAFe adoption, how they were hitting the goals and how the implementation of SAFe practices had changed since the initial adoption in January 2016. These trainings were seen as beneficial recaps and helped teams stay aligned and informed about how the organization was progressing, as well as discuss concerns and ideas for improvement. Being face-to-face was deemed as essential to ensuring proper communication.

The difference between Unit 1 and Unit 2 is that Unit 1 employed consultants very sparingly prior to the adoption and later using them for audits, while Unit 2 used them to train people and adapt SAFe before the adoption. Bringing the consultants back to perform an audit of the SAFe adoption and current state around 6–7 months after adoption was seen as very beneficial to the whole organization, providing concrete helping point for all stakeholders.

One main topic for improvement was to customize and adapt SAFe more to the organization instead of doing everything according to what's been prescribed. Enabling middle management to better understand the principles of SAFe was helpful, also helping to improve their grooming practices. One manager emphasized continuous learning, having participated in multiple conferences on agile since Comptel moved to SAFe.

NAPA

Having consultants on-site helping teams with impediments and improving their ways of working, as well as managing the change itself, was mentioned as a success factor in one interview.

4.4.3 Internal change agent

Comptel

Change agents was described by an interviewed scrum master as being absolutely central to the SAFe adoption at Comptel's Unit 2 business unit, stating that *"I like SAFe quite a lot, maybe 8 out of 10, I would say. But I would say all that is due to [name of change agent], not due to the SAFe in and of itself."*

Having someone from inside the organization organizing trainings, handling administrative problems, keeping people informed and answering questions were all aspects that were described as success factors in the adoption.

At Unit 1, the role of internal change agent was less clear: there was less systematicness in the adoption itself, and less mention of any specific person, so cannot be mentioned as a specific success factor.

NAPA

The main agent for change at NAPA was the software and release process owner who had discovered SAFe by himself. The fact that he was driving the change and "selling" the idea internally inspired developers, one of which later took on both the formal role of process owner and as change agent. Being able to see the needs and problems of the developers was cited as own success factor in driving the change across the developmental organization, and he was described as handling the change with the right attitude. One key internal change agent summarized the problem of external agents as:

> *"I do not believe that I could, like, do the equivalent now, go to another company and be a rainmaker and be like hey, let's put*

this thing of yours into shape."

4.4.4 Appropriate staffing

Comptel

Having a skilled and motivated full-time release train engineer (RTE), internally recruited from product management, was deemed as the most important success factor by three interviewees from both management and teams. While finding a driven person with a knowledge of the organization, its ways of working and its people was deemed the most important, one interviewee emphasized that the role of RTE cannot be handled as a side duty. When asked to describe whether any reservations towards SAFe existed, one senior developer answered no but elaborated that this was due to the RTE and the way RTE drove the SAFe adoption at Comptel. The developer concluded:

> *"I would say it's quite critical for the release train engineer to be generally likable by people, otherwise it's probably going to be fairly horrible for everybody involved."*

NAPA

Interviewees at NAPA stressed the skill level of their teams and finding and coaching motivated scrum masters. POs and teams work together efficiently on both sprint and release level, so the adoption of SAFe did not cause drastic changes to their daily work. The scrum masters of teams are motivated and actively try to improve the ways of working of the teams via retrospectives and experimentation, so stagnation is less frequent and problems addressed faster. The RTE and agile coach helped scrum masters reflect over different problematic topics to ensure that nothing was left out.

4.4.5 Leadership support for adoption

Comptel

One of the original drivers of the change was a member of the steering group of Unit 2's R&D, having previously been an internal Scrum coach at the time of Comptel's Scrum adoption. Having a higher-level employee explain the potential benefits of adopting SAFe—the potential for scaling, the possibility of increased throughput with the same amount of teams, etc.—to leadership and steering groups helped cement support for the adoption and gave a strong mandate for it.

One interviewee stressed that strong top level leadership support is vital for the adoption of SAFe, since any organizational change will generate resistance with SAFe generating even more, being more comprehensive. Having the leadership actually understand SAFe and why it should be implemented (e.g. through participating in trainings together with employees), and then helping implement it, greatly helped to increase motivation at Comptel.

Another aspect mentioned was the support of leadership towards middle management: product owners discuss the adoption and related matters weekly with the portfolio owner, so as to keep them in the loop and on track.

NAPA

As with Comptel, the change—stemming from middle management—had strong leadership support. No specifics were mentioned, but one interviewee summarizes it as *"Without it [leadership support], it would not have succeeded in any way."*

4.4.6 Customization of SAFe

Comptel

Taking the time to identify which parts of SAFe fit into Comptel and which parts do not, without shying away from some basic principles such as synchronization and the PI planning, was seen as a success factor during the adoption. The principle of a joint sprint demo, in which each team has a short time slot and after which all material is shared, was seen as having more pros than cons even though teams might not always see it as relevant. Having the principle that team retrospectives should always impact some practice was deemed important to prevent stagnation and change resistance.

NAPA

One interviewee stressed that while the SAFe-prescribed way of doing PI planning contains many good ideas and principles, no-one can achieve a perfectly working planning session directly. Rather, adapting and customizing it as feedback and experiences are gathered is the way to go. SAFe is aimed at firms of very variable size, and so taking the size and structure of the organization into account is important when adopting it. At NAPA, some roles, e.g. value stream engineer and release train engineer, were merged, and this was deemed to be working well. Allowing teams to customize their own practices was also mentioned as important, as some teams enjoy focusing more on developmental practices while others try new ways of working.

The four-month release schedule used by NAPA, in which the final month is the so-called cross-functional month, has generated a lot of positive feedback. During this month, teams can work on remaining work, new ideas, work together with marketing to create marketing material and so forth.

4.4.7 Retrospectives

Comptel

At Comptel, retrospectives were successfully used to customize aspects of SAFe that were not working well, such as removing one unnecessary PO-specific meeting during the PI planning. One scrum master mentioned that on the team level, retrospectives work very well: the team is getting better at what they're doing, trust is increasing, problems are solved etc. As stated by a manager:

> *"We have purposely strived towards having each retro leading to some change, because otherwise the retros become completely useless and nobody attends them and then begins, like, a lot of complaining."*

However, the scrum master felt that there was not much that teams could do to impact the practices of higher levels of the organization. Another interviewee mentioned that portfolio management had been improved through their own retrospectives on that level, but did not mention specifics.

4.4.8 Agile already introduced in teams

Comptel

The fact that Scrum had been adopted previously meant that the daily life of team members did not change all that much when adopting SAFe. Scrum was working well, and one manager stated that it was not so much about scaling development from one agile team to multiple, but about seeing how Comptel can adapt SAFe to its particular needs. In keeping with the spirit of continuous improvement and empowerment, teams do not have to follow the specific methodology as set out by SAFe but are free to adopt and adapt as suits them as long as the base practices—the interviewee mentioned planning sessions and reviews on both team and program level—are followed. One team chose to adopt Kanban after finding that the sprint cadence of Scrum did not fit them.

NAPA

At the start of NAPA's agile transformation a strict regime of Scrum was implemented, but as teams settled in with the practices the requirements were gradually loosened. This implementation of Scrum was very beneficial to help NAPA move towards team-based development and also structuring the work with the help of product owners and sprints. Later on, with leadership support for continuous improvement, some teams have moved to Kanban or Scrumban, and other left out or modified parts which do not benefit them.

> *"...Scrum was right up our alley for our situation in a way, like how to move from a sort of, as I described it, very individual development to a team type operation and so forth."*

Some interviewees mentioned the fact that the team level was already agile and working quite well as a success factor for the adoption, also speculating that the adoption would have been much more challenging if a wholesome agile transformation would have been needed.

4.4.9 Metrics

NAPA

Several interviewees mentioned metrics as a significant success factor, in that simply beginning to measure certain areas, such as code complexity and test coverage led to attention being focused on said areas, whereby they improved without any specific goals or performance incentives tied to these. Later on, relevant milestones were developed and performance-based rewards created in order motivate improvement.

On the team level, scrum masters are allowed and encouraged to develop their own metrics with their team in order to improve their velocity and ways of working. Since the metrics are developed by the team, everyone feels more motivated to keep measuring and improving them.

4.4.10 Appropriate tools

Comptel

Comptel employs JIRA in every layer of the organization, which has worked well, according to one interviewed leader.

NAPA

One interviewee mentioned that employees at NAPA did not use to switch on their cameras when talking to each other via video chat. This did not happen during meetings or otherwise, even though the employed video and chat tool (Skype) supports it natively. Using live video feeds did not markedly improve after Scrum adoption, but has since become established as the norm. As the interviewee puts it:

> *"Because it's, uh, I can see the difference because when I've been talking to people so much and also to the teams like it's, very very clear that when the cams are on, meetings are more productive."*

One tool which Finnish interviewees identified as a success factor when mitigating cultural differences was Yammer, a social networking service built for internal enterprise use. NAPA extensively employs Yammer to elicit comments and feedback from all employees, but especially Asian sites use it more actively as they either cannot participate in the live plannings or do not feel comfortable speaking up at the time. Product owners also use the service to gather input from team members across all sites, and channels can be created as needed.

4.4.11 Other success factors

Comptel

While not a success factor in itself, the lack of any true change resistance was mentioned several times during interviews.

One scrum master mentioned that DevOps practices had been introduced as part of the SAFe adoption, with automated tests running after every build, and that this had greatly helped code quality.

Weekly scrum of scrums were mentioned by one interviewee as useful, with real discussion happening organically without reverting to status meetings.

NAPA

One aspect, mentioned by a different interviewee, is the input given to teams by stakeholders during sprint reviews. As the stakeholders present are of varying backgrounds, teams get useful information related to e.g. coding, and sometimes even customers provide information that teams can use.

Table 4.4: Success factors

Success factors	**Description**
PI planning	The first PI planning proved essential in helping employees understand the potential benefits of SAFe at all case organizations.
Consultants & training	Employing external professionals to help train staff and customize the adoption was a major success factor especially at Comptel.
Internal change agent	At Unit 2 and NAPA, a vocal internal evangelist helped cement acceptance for the change as well as drive it forward.
Appropriate staffing	Finding motivated and skilled persons for each role proved a success at NAPA and Unit 2.
Leadership support for adoption	Securing leadership support for the adoption was necessary in all organizations and created an air of validity.
Customization of SAFe	Customizing different aspects of SAFe to the organization and its context was important, at NAPA in particular.
Retrospectives	A systematic approach to continuous improvement helped identify pain points of the adoption at Unit 2.
Agile already introduced to teams	The adoption at all organizations was made less painful by the fact that teams were already familiar with agile.
Metrics	At NAPA, self-developed metrics increase motivation and help with management.
Appropriate tools	Using the right tools increased productivity, especially seen at NAPA's off-shore sites.

4.5 Challenges

In this section, we present the results related to the fourth research question of this thesis:

> *RQ4: What common challenges for adopting SAFe can be identified?*

A wide variety of challenges were reported, relating to many aspects of SAFe, but also to global software development and to general organizational transformation. The reasons have been listed in table 4.5. Each challenge is described in its own subsection, with minor challenges being described together in a separate section after these.

4.5.1 Change resistance

Comptel

Both business units at Comptel had relatively quick ramp-ups of the SAFe adoption, but the intensity of the change resistance was much more widely described in interviews from Unit 1 employees. There were challenges from the start: members of product management were unwilling to change their schedules and duties according to the roadmap for the adoption, and breaking up the organizational silos and walls between different roles which existed also generated conflict. A concrete example was that some product owners were very reluctant to interact with and comment on portfolio level items (in JIRA) since it was perceived as the product managers' responsibility. The product managers, on the other hand, were reluctant to participate in grooming sessions and work with product owners on backlog items and involve them in making roadmaps. One interviewee speculates that this initial change resistance in the program layer was why the adoption took longer than anticipated.

A leadership level respondent mentioned that there might have been a general sentiment amongst teams that the adoption of SAFe was something leadership just ordered teams to do, and that in some teams the change resistance became very personified, but that most of the resistance was concentrated to around one fourth of the teams. The respondent speculated that team change resistance could have been drastically reduced by involving the team leads more before the actual adoption began, in e.g. trainings and info sessions. This would also have helped create understanding of e.g. why the PI planning happens for ten weeks forward, as some employees now felt that SAFe was a return to waterfall practices.

A team level interviewee felt that initial skepticism existed, and that the rationale for the SAFe adoption was lacking. One manager also underlined the fact that teams initially saw SAFe as restrictive and bureaucratic, and that acceptance for the adoption was low. However, nowadays SAFe was just seen as part of the process.

In Unit 2, one interviewee stated that some still see SAFe as tacked-on bureaucracy, while others are genuinely pleased with the increase in communication and visibility between teams. A manager admitted that the own initial response towards SAFe was one of skepticism:

> *"... one has to admit that when looking at SAFe as a process from the outside, it looks extremely heavy. And that's the first things that jumps up. That, hey, there's an awful amount of management here, terribly large meetings; that who would be able to run this."*

As with Unit 1, the largest challenge was perceived to be in product management: how to change mindsets and ways of working towards those of SAFe and more specifically the program layer of SAFe. One team level interviewee stated that some managers still exhibit ignorance and apathy where SAFe is concerned, which does not help sentiments amongst team members.

Another respondent felt that while it existed in both business units, the criticism towards SAFe was more vocal in Unit 1 than in Unit 2.

NAPA

Change resistance at NAPA manifested itself especially during the first phase when all-out strict Scrum was taken into use. When aspects of SAFe were adopted, change resistance was stronger in more senior employees, some of which had been at the company for roughly 20 years. These had been accustomed to a less disciplined, more individual approach to software development and saw SAFe as horribly bureaucratic and even complained about such as aspect as mandatory (partial) test automation (even though this is not specifically part of SAFe). One respondent felt that very pessimistic individuals could easily "poison" the attitude of the entire team—or organization, as there was initially one vocal critic in the leadership team—and partly sabotage the change, and that dealing with such individuals was problematic. However, during the initial phase the turnover of employees increased, which the respondent saw as mostly beneficial:

> *"...since in a certain way constantly grumbling about how something is so stupid, then maybe the person changes scenery and is happier somewhere else as a result..."*

Several interviewees felt that a lot of the skepticism was not towards SAFe itself, but agile principles altogether. The necessity of retrospectives was questioned in at least one team, and the benefit of metrics were not obvious to all teams. Older employees were mentioned several times as more critical towards the new ways of working, and one respondent was amazed at how fierce resistance the use of certain terms generated. Some employees, for instance, could not stand the terms "Scrum" or "retrospective", but had no problem with discussing problems regularly or developing in iterations.

4.5.2 Cultural differences

Comptel

Cultural differences were cited as a challenge when discussing problems. One team had weekly reviews, but that teams in certain sites rarely reported any problems or failings, even when e.g. missing deadlines, leading to a lot of spilling.

One manager proposed that it was not so much about differences between countries, but between individuals and especially between more traditional and newer software development aspects. As SAFe aims to give power to teams and increase visibility of everyone's work, team members might feel embarrassment or discomfort with showing their own work or being open about problems. The manager also summed up the problem of plan-driven waterfall-style mindsets in agile organizations:

> *"... they want everything to be planned right to the minutes. They need to have proper plan, they can't um start, hit the ground running, and they can't accept the fact that they will fail, they can't understand the fact that it's okay to fail fast and improve later on. So those are the kind of mentalities, so especially PI planning I think is a big problem to them because they tend to kind of do planning in a very extensive detailed level."*

The manager also shared the view that differences in communication culture were quite clear, and something to be aware of across the organization. Finnish people were seen as blunt but honest, and Asian cultures as more polite but reserved with regards to reporting problems.

Cultural differences also posed a challenge to the cross-functionality of teams in Unit 2 even though Scrum had been in use since 2010. The main reason for this was the perception in some sites that testing was less valuable than programming, meaning that team members were less willing to help in that area in case of bottlenecks.

NAPA

Cultural differences were cited as very challenging, especially between different contintents. One respondent from Finland felt that teams in Romania were easier to work with, as the respondent felt that Asian culture was very dependent on management, with too little self-organization and initiative. In the respondent's view, Nordic work culture is based on giving the team responsibilities and a frame to work within, and then expect results. It felt difficult to "take away" the power to make decisions regarding one's own work, as the respondent felt was necessary to do when working with teams from Asian cultures.

The difference between collocated and distributed teams was also highlighted in the context of cultural differences. According to one interviewee, some distributed teams with members in India and Finland were at the brink of open war at times (during the period when NAPA had a development presence in India), especially in those teams in which there was a large competence gap between the senior ship industry experts in Finland and new recruits in India, who understandably did not possess the same knowledge of the shipping industry but who also needed a lot of guidance otherwise, something which the Finns were not used to.

4.5.3 Estimation, planning and management of work

Comptel

In Unit 1, planning, estimating and managing work on a program level proved to be challenging. One interviewee stressed that teams needed to be aware of what items are next in line, since showing completely new epics for the first time during the PI planning without any prior input or grooming might not be very motivating. A team level respondent simply stated that grooming cannot realistically happen in two days, but rather that as much as possible should be done beforehand. The main problem, according to the interviewee, was that there is no-one in the organization who can actually do the grooming, and much of the responsibility is on the team. Another team level interviewee supported both of the points, stating that it was rare to receive good business requirements or groomed portfolio-level items, and that the teams had trouble grooming:

> *"... in the very early part of SAFe, times of SAFe we run into the PI planning days, and we see the items for the first time, it's impossible physically for the teams in two days time to groom them*

> *to a technical level in such a detail that we can actually commit to the efforts. And to the ten weeks ahead, it's just impossible."*

Tying in with grooming, estimation was mentioned as a major challenge by interviewees from both Unit 1 and Unit 2, with one Unit 1 team level interviewee going so far as describing estimation as something which they did "very poorly" and as their weakest area altogether. In both cases, the underestimation of stories was frequent and quite severe, leading to continuous spilling and to an IP sprint which was mainly dedicated to work on stories. In some cases—and in both business units—some teams add stories to the IP sprint already during PI planning, or are otherwise overly optimistic in their estimates of the sprint work. Especially during the first PI there was little understanding of how much buffer should be allocated for ad hoc work and sudden changes, leading to severe spilling. Two interviewees mentioned that product managers can be quite aggressive during the planning sessions, sometimes pressuring teams into taking more work than would be feasible.

Respondents across the business units, from team to manager to leadership, all mentioned the same two-fold problem of stories spilling coupled with using the IP sprint as a buffer or regular work sprint, meaning that very few teams get to use all, or any of, the IP sprint for the prescribed innovation. One interviewee from Unit 1 had heard of three times over two years in which some team had managed to do roughly one full sprint of innovation, but that otherwise the IP sprint is not used for that:

> *"We are not using it, exactly that's the problem. Because there's always, because the customer commitment is always more priority than anything else, so whatever technical depth or possible innovation that would improve processes or whatever, is lowered in the priority so the final innovation sprint is always used to finish up things."*

The team in question was not planning anything to the IP sprint, and using a 15 % buffer on their estimates, but still had severe problems with over-commitment. One interviewed leader stated that the continuing problem of over-commitment was a combination of several things, such as a lack of grooming and prioritization before the planning (leading to a lack of understanding of the work required to complete a feature), an overly optimistic attitude of some teams and product managers, and of downright pressure to over-commit in some teams.

NAPA

One interviewee stated that release planning cannot happen during one or two days, and that it was a significant challenge to get people to collaborate on backlog items enough beforehand. Spilling was severe during in the beginning, but has since reduced. On the team level, one interviewee mentioned that spilling and especially scope change varied drastically between teams, with some teams having up to 170 % scope change during a single sprint. As with Comptel, some product owners and managers had challenges prioritizing backlog items and adhering to SAFe principles, leading to teams filling up their IP sprint with backlog work.

4.5.4 Lack of communication about adoption

Comptel

One recurring challenge mentioned during the interviews was a lack of communication. Interviewees from both business units, and especially from team level, did not feel that the rationale or justification behind the SAFe adoption was explained. One interviewee had a vague feeling that it was due to a need for coordination and follow-up on the program level, while another lamented that there was not really anything that showed a particular need to adopt SAFe, summarizing that:

> *"Honestly, well... Honestly I would say that it was probably one or two people kinda coming up with it or something like that, uhh... I didn't see any justification for it [the selection of SAFe], any at all..."*

A manager in Unit 1 mentioned this as a mistake during the initial adoption, stating that there should have been a conscious effort to explain the reasons behind choosing to adopt SAFe, and to constantly remind everyone of why it was being done, how it was affecting everyone and what the potential benefits were. As of now, changing the mindsets of employees was proving to be challenging:

> *"I think our message on why we are doing SAFe is not strong enough. If I would, given a second chance, I would spend more time to talk about SAFe."*

Another challenge mentioned was that (some) teams simply do not want to communicate, with some team members complaining that joint demos etc. are not useful since they do not want to know what other teams are doing.

4.5.5 Lack of continuous improvement

Comptel

The lack of continuous improvement was mentioned especially in interviewees from Unit 1, being mentioned in all but one interview. One interviewee mentioned that it felt as if the adoption was treading water: sessions were becoming ritualistic with little focus on dependency or risk management. Another leadership interviewee mentioned that the approach to continuous improvement had been slow and unsystematic, even though there has been significant negative feedback from retrospectives. A third interviewee from team level agreed, stating that there were some issues which apparently had been circulating in retrospectives since the beginning with no-one taking the initiative to address them. The difficulty of customizing SAFe to the context of Comptel was apparent to the interviewee, who stated that everyone wanted to participate in improving the ways of working but that no-one was really taking action to systematically and continuously do it (or help facilitate it for teams). The interviewee summed it up somewhat dramatically:

> *"I have been here just for three PI's but what have I heard is that, people just talk to me as if nothing has changed in the past year. This is what others say to me; I mean, I'm not even on their level because I've been here just for three increments so I still have hope, but other people don't really have hope."*

One interviewee mentioned that some issues might require escalation or additional resources, and that these are often not addressed since customer concerns are prioritized. One concrete example mentioned was test automation: the team had testers with harsh work loads, and simply running the tests take all of their time. Although allocating time for automating tests has been raised during several retrospectives, it is always pushed lower on the list of priorities. The interviewee mentioned that many similar issues concerning technical debt are often prioritized too low to be able to address them. This, in turn, has led to teams simply ignoring certain problems when discussing in retrospectives, and has in some teams created a feeling of pointlessness where retrospectives are concerned. In one team, retrospectives had not been held at all for two sprints, nominally due to a high workload but also due to retrospectives being seen as not that useful since nothing happens as a result of them. One interviewee mentioned that the work load has a big impact:

> *"Currently I think some teams have been really good, because they have been responding to the need and to the context, um, they*

> *really changed their way of working. For some teams, at least the more established ones, I think they are just too busy with all these customer escalations and what not, so they are kind of "just" doing their work ... They might have not realized that they didn't improve their way of working."*

Disciplined risk and dependency management was mentioned as challenging in one interview: risk matrices and dependencies had been part of the PI planning for several PI's but the benefits were still unclear, as there is no real ownership of said risks and dependencies. Demos were mentioned as less beneficial, as they are more for the process and show rather than demonstrating fully integrated slices of functionality.

In Unit 2, there was tendency for some product managers and owners to see the PI planning as some sort of ritual, recycling presentations and speeches across planning sessions without actually explaining e.g. the terminology or the reasons behind some backlog item. However, there were no mentions of a lack of continuous improvement on team level nor on program level.

NAPA

A lack of discipline and focus was something that one interviewee mentioned as a challenge in some teams. In some cases, product managers might try to get teams to work on items outside of scope or irrelevant to the current release, and even though the team might refuse the ensuing discussion would still waste a significant number of work hours. This was an issue that had been prevalent since the beginning of the adoption, and was still not entirely solved.

4.5.6 Lack of drivers of change

Comptel

No agile coach was present at either business unit of Comptel currently, and some interviewees cited a lack of internal change agents as a challenge to the adoption. If there had been some evangelist or change agent to help create acceptance for the change then the transformation might have succeeded more easily. One respondent mentioned that the structure in the Finnish part of Unit 1 was lacking a proper on-site RTE, as the program managers were basically supervisors with tens of subordinates and a lot of other tasks. As such, they act in more traditional managerial roles rather than acting as coordinators and enablers for the teams. One program level interviewee

lamented that more change agents would have helped Comptel change its established ways of working, and that some had misunderstood SAFe and agile altogether:

> *"SAFe is not a process. Agile is not a process. It's a way of working. The moment you see it as a process, I think it defeats the objective, because when seen as a process, then you need to, then you wait for someone tell a clear instruction how to do it. But being agile means that you challenge the status quo."*

NAPA

NAPA originally took Scrum into use somewhat religiously, but as the grip was eased and teams allowed to experiment more freely it reportedly led to problems. In one team, scrum masters rotated too quickly with no one single person ever taking on the full responsibilities of the scrum master. Two interviewees mentioned that in some teams with fixed scrum masters, there were problems with the scrum masters becoming set in their ways, with no experimentation of new ways of working and no real drive to help the team change for the better. One respondent mentioned that there was a lot more change resistance in those teams.

4.5.7 Lack of training or support

Comptel

The adoption at Unit 1 suffered from a lack of training. One employee felt that having weeks of training for such a large organizational change would not have been unrealistic, but that even holding the two-day certification training for everyone—which was not the case—would have been very beneficial. As a result, the planning sessions felt more like ceremonies. This lack of engagement was mentioned by one respondent from leadership, in whose opinion the planning of the adoption did not involve every layer of the R&D organization sufficiently to generate commitment to the principles behind SAFe. As such, change resistance grew unnecessarily. Another manager mentioned that the impact of the adoption was not realized, or at least communicated, well enough to all organization members, and that this had only recently been remedied through "roadshows".

4.5.8 Organizational challenges

Comptel

No agile team operates in a vacuum, and an aspect brought out in multiple interviews is the effect of the business environment and surrounding organization on the adoption of SAFe. For instance, one leader in Unit 2 mentioned that HR processes for career development, in which discussions happen once per year, were not in line with the agile aspect of developing in much shorter cycles, be they PI's or sprints. Budgeting was also mentioned as a challenge to implementing "pure" SAFe or agile in general, since currently budgeting still occurs according to products instead of for the ART as a whole. This meant that getting the portfolio level to work as intended was highly challenging, and the line between program and portfolio levels hazy, since product managers had to make decisions that should have resided with portfolio management.

Organizing teams and roles around the concept of the Agile Release Train also proved challenging, as some teams were much less interdependent then others, at least in Unit 1. One interviewee speculated that there might have been less change resistance if the teams had been organized into several smaller but highly interdependent ARTs. There were also mentions of dependencies within teams, as there might be one or two key persons in that team tied up with a specific item, setting the team back for that whole sprint.

One challenge concerning the entire R&D organization reported in Unit 1 was the lack of any clear goal for the adoption of SAFe, which would help create alignment toward it and foster motivation. Another is the amount of work required just for the logistics of the adoption, as teams are not collocated. A third challenge mentioned in both sub-units was unsuitable tools: the same JIRA was still in use in some cases without proper adaptation to SAFe, leading to overhead and frustration.

NAPA

One interviewee mentioned that there were challenges with cross-functionality in teams. Not all teams are truly self-organizing in that they push decisions and responsibilities to other people, teams and roles, and this causes some issues or decisions to circulate without any true owner. In other cases, the responsibilities of the team are so diverse that e.g. daily stand-up meetings were almost useless, as everybody talked about their own area without any real understanding of what their co-workers were doing.

4.5.9 PI planning

Comptel

While the PI planning was widely reported as a success factor, there was some criticism of the work load involved with planning and participating in it. In Unit 1, tying in with the change resistance, there is still some vocal criticism of the PI planning: some people do not feel they gain anything from synchronizing with other teams, and in some cases the dual planning (for the PI and for sprints within the PI) is seen as overhead. Mentioned by two interviewees, the repetitiveness of the PI planning and the intensiveness of the two planning days were mentioned as challenges. One interviewee stated the following:

> *"Then again we felt disconnected with the other teams, so it did come up a lot that we are wasting time by listening to things that do not impact our daily work."*

Getting the PI planning days to go smoothly is challenging, especially coupled with the challenges related to backlog item grooming. As some items are shown to the teams for the first time during the planning, they might feel frustrated at having to work with so little information and might not be able to estimate and plan optimally. One interviewee in Unit 2 felt that the planning sessions were more beneficial for leadership and management than for team work, since the teams know what they are going to do for the next two weeks but managers and leaders want to see where the whole train is going. As such, getting people and teams to interact with each other is a constant challenge, and many participants do not actively discuss at all.

NAPA

Two interviewees mentioned that release planning was perceived as overhead or a waste of time by some teams, and that one challenge was to get people to actually participate instead of listening to the session from their own computer.

4.5.10 Poor technical base

Comptel

While SAFe should enhance cross-team collaboration, the haphazard code base, lack of common platform, lack of common coding standards and lacking DevOps and CI practices were all mentioned as initial challenges to receiving

the full benefits of SAFe. As the release schedule is separate from the PI cadence, one interviewee lamented the amount of overhead this caused, since doing releases is not trivial (due to the aforementioned issues), and suggested a cadence tied to releases as a possible solution.

NAPA

In the beginning of the adoption, test automation was completely lacking, leading to a large amount of time being dedicated to regression testing.

4.5.11 Portfolio management and prioritization

Comptel

One interviewee from Unit 1 made the point that business and customer requirements are too decoupled from the technical implementation of them, and felt that it was one major challenge of the upper levels of the organization. If customers were required to submit their ideas and requirements through a formally defined funnel, with portfolio level prioritization and architectural input, it would be easier to align the entire development towards those requirements. A second interviewee supported this, stating that items are not nearly groomed enough when arriving to teams, lacking architectural input and lacking a holistic view of how each item fits into the bigger picture.

Previously mentioned under Organizational challenges, the lack of a proper division into portfolio and program levels was mentioned by an interviewed Unit 2 employee as problematic since it pits various backlog items of different sizes and levels against each other. With no single person or group who could prioritize them properly (as constructing a proper portfolio team was also a challenge), business needs are often prioritized above technical ones, especially if the choice is left to product managers. In Unit 1, the portfolio level was originally almost non-existent—which proved very challenging—but has since been built up and improved upon.

NAPA

The portfolio level, from ideation to prioritization to decision making etc. was mentioned as challenging to get to work properly, since it requires input from several parts of the organization and sometimes teams also. An interviewee also mentioned that teams do not really understand the portfolio level, and that communication about it has been lacking. Team members are technically able to add ideas to the funnel, but they have no insight into

how the ideas move forward, how they are accepted or rejected, how the prioritization happens and so forth. Since teams cannot work on items which have not been formally approved, the time it takes for an idea to become reality is very long. The interviewee mentioned that the topic had come up during several retrospectives and that steps had been taken to mitigate the problems, but that upper management or leadership were still not quite agile.

The lack of a portfolio level architect, or even an understanding of the value of architectural epics, was named a challenge by an interviewed leader, as business driven development ideas and items inevitably win over architectural and technical ones even though the latter ones could enable more value to be built. An interviewed manager supported this, stating that the lack of an architectural runway or common high level architectural practices was hurting team collaboration in the long run. Getting prioritization to work smoothly was challenging also due to it being somewhat person-dependent, with each person handling it differently. One interviewed leader summed up portfolio management as follows:

> *"Well that's very challenging, I mean portfolio management is the most difficult thing of them all. Of course there's extreme challenges and the most difficult decisions and analyses and those kinds of things..."*

4.5.12 Miscellaneous challenges

A challenge mentioned in both NAPA and Comptel was the Scrum of Scrums. In NAPA, it was an organization-wide session for discussing the adoption, but it quickly reverted into a reporting session where no-one was interested in what the others were saying, so it is no longer held. At Comptel, it is still in use but is also somewhat of a reporting session, with everyone reporting possible problems but no real discussion about how to solve them.

4.6 Effects of adoption

In this section, we present the results related to the fifth research question of this thesis:

> *RQ5: What effects has the adoption of SAFe had in the case organizations?*

This section is split into two parts, the first concerning positive effects and the second negative effects.

Table 4.5: Challenges

Challenges	Description
Change resistance	Change resistance was widespread at Comptel and to a lesser degree at NAPA, and still persisting in some teams especially in Unit 2.
Cultural differences	Differences in ways of working and in reporting issues and obstacles between cultures led to friction in distributed teams at all case organizations, especially NAPA.
Estimation, planning and management of work	Teams at Comptel suffered from a lack of proper business requirements and grooming prior to planning, and from pressure to overcommit, with similar lesser challenges at NAPA.
Lack of communication about adoption	The lack of a proper rationale for the adoption of SAFe at Unit 2, and a general lack of communication about it at Unit 1, led to confusion and change resistance.
Lack of continuous improvement	Teams lacked the means to prioritize solutions to retrospective issues high enough in all case organizations, and some issues were recurring since the beginning at Unit 1.
Lack of drivers of change	The lack of agile coaches or drivers of change led to poor understanding of the adoption and agile principles at Unit 1.
Lack of training or support	Not everyone was trained before the adoption in Unit 1, leading to a lack of understanding of SAFe.
Organizational challenges	Rigidity in other organizational parts such as budgeting hindered the adoption, and cross-functionality was challenging especially at Comptel.
PI planning	Reducing wasted time and overhead and getting everyone to participate in the planning session was challenging at both Comptel and NAPA.
Poor technical base	When starting the adoption test automation coverage was low and standard practices for CI or DevOps lacking at Comptel in particular.
Portfolio management and prioritization	The lack of a properly implemented portfolio level led to challenges in grooming requirements and prioritization of technical features at all case organizations.

The effects following a SAFe adoption are not easily separated from the successes and challenges concerning the adoption itself, especially since an adoption—as with any organizational change—does not really have a fixed end date. As both companies already employed Scrum at team level it could be difficult to affirm that these reported effects stem from SAFe and not only a general agile transformation.

Benefits

4.6.1 More disciplined agile practices

Comptel

One interviewee from Unit 1 mentioned that SAFe has helped teams to integrate better with each other, in part by helping them refine their agile practices. In Unit 2, one manager shared the results of the latest employee satisfaction survey, in which roughly 50 % of the respondents felt that the topic "Common work practices are good, and they are supporting my daily work" had improved since the adoption of SAFe. In Unit 2, an interviewed manager listed, among other things, that Scrum of Scrum had become more useful, in that it is used as a discussion forum instead of a reporting session.

NAPA

An interviewed leader stated that adopting SAFe has helped with the processes of business driven software development, as SAFe encompasses a much broader range of functions than pure Scrum, which has the perspective of only the team:

> *"But the most essential thing is that it [SAFe] has built a kind of handbook for the big picture, or a sort of holistic approach that actually applies for this kind of business drive software development. Scrum was in a way just the perspective of that one team, and it does not, it only helps that team."*

One direct benefit of the adoption (although not necessarily restricted to SAFe) was that every team got a team-facing product owner. Previously some teams only had a product manager to convey customer requirements, which proved problematic since the product managers spent very little time, or no time at all, with the teams.

Two interviewees mentioned that teams have more freedom of choice as they are not restricted to using pure Scrum. Teams are free to choose from and customize other agile methodologies as well.

4.6.2 Better alignment across the organization

Comptel

Interviewees in both Unit 1 and Unit 2 stated that it is clearer for teams what the priorities for the entire business are currently. Teams have begun to understand that the whole organization moves towards a specific direction, and that this direction can be changed according to the cadence. The specifics behind the priorities, such as what some stakeholder is trying to achieve with their item or the combination of multiple items, is explained much better and helps teams understand and implement them better. Architectural runways, business goals and drivers of these goals are explained better and teams understand them. On the team level, one interviewee states:

> *"I think from my level I can see that we are more cohesive, we have a common focus, so if we are doing something then we know why it's being built. Everyone knows what's going on."*

In Unit 2, in the latest employee satisfaction survey 65 % of respondents felt that the topic "Our product portfolio reflects the company's strategy" had improved. Interviewees from all levels feel more involved and have gained a better understanding of what their role is and why, how their daily work links to the strategy, and how it creates value for the stakeholders. There is also a general feeling of knowing what the common goal is, what different product visions are and how each team is contributing to the goals and visions. Some interviewees specifically mentioned the transparency of business goals and the value it creates, stating that e.g. knowing what some customer is paying for a specific feature is interesting and adds motivation.

NAPA

The PI planning has enabled more parts of the organization as well as external stakeholders to help keep the organization aligned towards its final goal by giving their input and expert opinions during the planning. It also acts as a checkpoint for stakeholders to make their final adjustments.

4.6.3 Better communication

Comptel

Interviewed leaders in both Unit 1 and Unit 2 described that the adoption of SAFe, and more specifically that of a portfolio level and board, has helped with external stakeholder communication. Having the clear cadence helps

with expectation management towards the customers, and having the PI planning allows everyone to get their voice heard.

Internal stakeholder communication has improved as well, and five interviewees stated that especially communication between teams has increased and improved. Meetings have a more open atmosphere, and teams are more likely to discuss with other teams of their own initiative, and as a result e.g. dependency management is much improved.

NAPA

One interviewed manager described cross-team communication as improved since the adoption of SAFe.

4.6.4 Better dependency discovery and management

Comptel

Better management of dependencies was mentioned as a benefit of the SAFe adoption in interviews across both business units. Teams proactively discuss and try to address dependencies, and one leader described that the improvement was dramatic during the first three PI planning sessions:

> *"In my opinion, one clear victory or benefit of this SAFe has been that these dependencies, dependency management, that dependencies have been visibly brought up already in the planning phase and then they are followed up on a weekly basis, like what has happened with this and that. And within the scope of it teams have even discussed proactively with each other, and we noticed a big difference between the first program increment plan and the second PI planning and then even the third one, like a huge difference in how proactively teams are bringing up dependencies. In the first PI planning you had to kind of fish it out of them, like might there be dependencies here or in this feature, but now teams have begun to think in the way that what do we need or what does someone else need from us."*

The leader reported that the feedback regarding dependencies had been very positive, stating that teams felt motivated to take ownership of dependencies since they felt it was more in "their hands". The decision making is more democratic in that managers do not dictate any one approach to dependencies and risks. One interviewee mentioned that even though dependencies and risks are not followed up as strictly in Unit 1 as in Unit 2, simply making them public made it easier to handle them.

4.6.5 Better software quality and DevOps practices

Comptel

In Unit 1, SAFe has had a positive effect on test automation coverage and continuous integration (CI) practices, as the framework has underlined the necessity of these. One interviewee mentioned that especially DevOps practices concerning the entire organization have improved, so teams working on the same product have a much better technical base for collaboration, such as integrated test environments and common (instead of team-specific) CI guidelines and resources. SAFe was mentioned as a good "excuse" to invest in and build up these practices better, as it helped point out the importance of those things to higher-ups in the company.

NAPA

An interviewee mentioned increased quality, and an improved focus on quality, as a continuous benefit of adopting aspects of SAFe. Through increased visibility people were more careful in what they presented as a finished product, and at the same time increased automation and better release processes helped teams achieve higher quality in a shorter time.

> *"If I think back a couple of years, the software kept crashing completely at random from almost anything [laughs]. And that does not happen any more; the quality awareness grew considerably."*

Another interviewee describes that previously, as everything was developed in projects, the pace was very fast but the quality suffered immensely. Even though developing features in SAFe might take a little longer, on a whole everybody is more satisfied and the end result is of better quality. In that interviewee's opinion, a greatly improved level of quality was the biggest difference that the adoption of SAFe brought.

4.6.6 Better synchronization and collaboration across teams

Comptel

An interviewed manager from Unit 2 saw that teams were collaborating better, with each other and with their stakeholders. Product owners had previously been responsible for making long-term road maps while being quite separated from the teams, while now they work more closely together with their team and plan according to the cadence. Team members reportedly also

enjoyed hearing about company strategy and the work of other teams during the PI planning. Another interviewed manager from Unit 1 stated that even forced synchronization between teams is good, since it has reduced dependency on product management for solving dependencies. One interviewee stated that cross-functionality has improved, and features requiring input from across the entire business unit and different locations are much easier to do currently than before adopting SAFe. Another stated that teams are working more cohesively as a unit when compared to pre-SAFe times. There is a common focus, and implementation happens in the same cadence instead of teams developing certain features at vastly different times. The interviewee felt that it brought more control to the development.

NAPA

Virtual teams across the organization work better together, according to one interviewee. Another interviewed leader stated that a blueprint for collaboration across the entire organization, not just within a team, was one of the most important things that SAFe has brought with it.

4.6.7 Better visibility

Comptel

Linked to the benefit of improved communication, an improvement in visibility was reported by interviewees in both business units and from all levels. Leaders and managers reported that teams felt more motivated since they could see the connection between the company's strategy and their daily work. The improved visibility also helps with reactivity, as problems, challenges and changes to e.g. business requirements are identified and brought up sooner across the organization. Accountability of every individual and teams has increased due to more visibility, according to one manager, and this helps not only with quality but also with motivation and a feeling of ownership. Interviewees reported that more visibility to other teams was beneficial even though they might not be working on the same product, as it helps align everyone to the same common goal and helps with dependency management:

> *"So it's cool to see what other people are doing because there are, our components are like, you know, connected to each other under the hood, so it's important to know what other people are planning to do, and then when we are maybe gonna use those things in the future. So that's one thing that we're finding that is very useful."*

NAPA

Improved visibility across the organization was reported in interviews from all organizational levels. Teams have better visibility towards other teams and the program and portfolio levels, and other parts of the organization such as marketing have a better understanding of the developmental organization and its processes. One leader felt that the increased visibility also increased trust in the organization, since people could better see what everyone else was doing. It makes it easier for managers to give freedoms to teams to work how they want, and teams better understand why e.g. certain prioritization decisions are done. However, one interviewee pointed out that visibility to the portfolio level was still lacking when compared to other levels, and that steps were being taken to address this.

4.6.8 Better, more frequent and more predictable releases

Comptel

The cadence helps bring predictability to the development when compared to the previous practice of working towards one major release roughly once every 10 months. One interviewee stated that even though things might spill from one PI to the next, it is still a much shorter window in which the feature will be developed when compared to having to guess at what time during the year the feature would be finished. One leader from Unit 2 reported that velocity of teams has steadily improved, and that the metric for achieved Epics per PI is going up every PI. There are fewer changes mid-sprint or mid-PI, and this amount of unplanned work has decreased constantly during the first four PI's.

A more agile product management is another benefit of SAFe, which helps with planning since product managers groom items earlier and more actively. While there are still high level plans made for each 10 month cycle, the focus has been "lowered" towards the teams. Adopting SAFe has also helped with prioritization, which in turn helps with producing more predictable releases as priorities are known to everyone. Previously team structures and responsibilities were rigid while nowadays they can be adapted to current priorities, and if need be the whole organization can pivot according to new priorities every PI.

NAPA

After the release process was implemented the frequency of releases has become faster and with better quality. One interviewee stated that releases previously were done once per year, with bug fixing beginning immediately after and continuing for a long time. Nowadays releases happen once every three months and the quality of them is much better. Coupled with the increase in frequency and quality, releases are more structured than previously since they follow a set process. Release planning was not something that was done before adopting SAFe, and one interviewee felt that they were previously more "rag-tag".

4.6.9 Miscellaneous benefits

One interviewed leader from Unit 2 in Comptel reported that teams really liked the principle behind IP sprints, since it officially sets aside time for improving e.g. CI practices and reducing technical debt.

The use of Kanban boards prompted the following comment from a program level interviewee at Comptel Unit 1:

> *"Yeah, one thing I might want to add is that, um, when it comes to the product management and budgeting I guess everybody currently is using the Kanban portfolio [board]. Which is great. Because through that, we are defining how our business case can be used. I think the business cases are now prioritized on a sort of regular basis when they arrive on the board, and they have a physical board in Helsinki now. So that, so we can see some product managers, some persons kind of doing their daily stand-up there some times. I think that's kind of a fun sight to see. "*

Drawbacks

Reported drawbacks were significantly fewer than benefits, somewhat similarly to most case studies reviewed. Whether this is due to a reluctance to report problems or due to the relative newness of SAFe in the case organizations is hard to determine, but all drawbacks reported are listed here even though they might not be specific to SAFe. Some drawbacks are in conflict with reported benefits, e.g. communication was mostly reported to have improved but in some instances and contexts it was reported as being worse.

Table 4.6: Benefits

Benefits	**Description**
More disciplined agile practices	At NAPA, each team has a PO and the organization as a whole is more agile, and at Comptel Unit 2 Scrum of scrums are more useful.
Better alignment across the organization	Everyone at Comptel has a better understanding of the common goal and underlying company strategy, and how their own work ties into these. Stakeholders at NAPA are better aligned.
Better communication	Communication with stakeholders and between teams has increased and improved at both units at Comptel, and cross-team communication at NAPA has improved.
Better dependency discovery and management	Dependencies are discovered, discussed and addressed more proactively and in better cooperation at both Comptel business units.
Better software quality and DevOps practices	SAFe has helped set a focus on quality by mandating the improvement of DevOps and CI practices as well as test automation at Unit 1 and NAPA.
Better synchronization and collaboration across teams	Teams and PO's work together more closely, and teams collaborate more cohesively together towards a common goal, especially at Unit 2 but also to some degree at Unit 1. At NAPA, virtual teams are collaborating better.
Better visibility	Visibility across all of the case organizations has improved, leading to increased accountability and trust.
Better, more frequent and more predictable releases	Shortened cycles and better prioritization practices have led to more predictable releases and higher and more stable development velocity at Unit 2 and NAPA.

4.6.10 Less autonomy for teams

Comptel

One interviewee in Unit 1 reported that one minor drawback of SAFe and the cadence is that teams cannot choose their own sprint lengths.

NAPA

One challenge reported was getting the portfolio level to work well, and as a result one drawback is the long time it takes for ideas and suggestions to pass through the funnel. If a team member suggests some feature or technical solution it might take months for the portfolio level to reach decisions, instead of empowering teams to decide for themselves.

4.6.11 More overhead

Comptel

Two interviewees stated that SAFe brings additional overhead to teams and to scrum masters. In Unit 1, there was mention of new metrics which had to be gathered manually and collected into e.g. Excel and PowerPoint. While it had been done before SAFe, the interviewee felt this work had increased.

The PI planning was mentioned by three interviewees as wasting a lot of time since the amount of active participants at any one time is quite low:

> *"... probably the biggest one [drawback] that comes to my mind right now is... is that these two day plannings, uh, honestly usually it is so that, I don't mean the presentations, but in the planning session usually it's like three people from a team participating in the actual planning and then the rest are just sitting there looking at their cell phones. So that's a bit of waste of time."*

Some team members feel that the increased visibility is simply information overload, and would rather continue working, but all interviewees also mentioned in the same breath that as a whole, the PI planning has been beneficial.

One recurring theme in Unit 1 retrospectives is that the program backlog is used as a bureaucratic tool for control by some product managers. Instead of allowing teams to decide how to implement some feature by themselves some product managers attempt to constrict what teams work on at any given time.

4.6.12 Reduced communication

NAPA

Two interviewees mentioned that getting the communication between teams and customers to work is challenging. It should flow via product managers, but some of these spent too much time with customers and too little time with teams, or vice versa. Previously some teams communicated directly with the customers, but many are now relying on product managers to handle it when in reality some issue could be solved more quickly via direct communication. While one interviewee stated that this is not something SAFe prescribes, the lack of flexibility or the current division of responsibility was problematic in that it reduced communication. One interviewee felt that the communication between teams and upper levels of the organization, especially product managers, is worse in SAFe, and that this was the case in all teams, but did not specify further.

4.6.13 Work satisfaction decreased

Comptel

In Unit 1, employee satisfaction surveys showed that work satisfaction had gone down since adopting SAFe. Linked to change resistance, one interviewee reported that some employees felt that *"... the autonomy had been lost and that top-down dictation politics had return and some are even of the opinion that this isn't agile in the way that increments have led to a return of the waterfall model."* Product management had always been strong in Unit 1, and the interviewee stated that in some cases SAFe might have strengthened it, leading to pressure on teams to overcommit which might explain the lowered work satisfaction.

4.6.14 Unplanned work not handled well

Comptel

The pressure to commit to work items and the cadence has led to worse handling of work that happens without planning, such as working on technical items or debt. While the IP sprint is nominally reserved for that purpose, it is often used for actual work. One interviewee from Unit 2 mentioned that especially during the first PI the lack of a proper buffer for ad hoc work led to challenges with support items and bug fixes, but that the situation has since improved a bit. The cadence means that items and bugs inevitably have to

wait at least for some time before they are taken into a PI and sprint. This has proved problematic in some cases, and the had even been a couple of situations in both Unit 1 and Unit 2 in which teams have been "removed" from SAFe and all agile practices in order to focus all effort on critical items.

NAPA

One interviewee mentioned that things which are outside the rhythm or cadence set by SAFe receive less attention, such as proper communication with other parts of the organization.

Table 4.7: Drawbacks

Drawbacks	**Description**
Less autonomy for teams	Some team members in Unit 1 would like to set their own sprint length.
More overhead	In Unit 1, manual reporting of metrics has increased, and large parts of the PI planning feels like wasted time.
Reduced communicated	Communication between teams and customers, and between teams and product managers, has become worse at NAPA.
Work satisfaction decreased	Team members at Unit 1 feel that there is less agility and more control after SAFe adoption.
Unplanned work not handled well	SAFe does not address ad hoc work properly, leading to sporadic task forces at Comptel Unit 1 and 2.

Chapter 5

Discussion

In this chapter, the results of the interviews are discussed through each research question: existing literature and interview results are compared, and similarities and disparities discussed. Limitations and threats to the validity of the study are presented at the end of the chapter.

5.1 RQ1 – Reasons for Adoption

The first research question this thesis set out to answer is *"Why have the case organizations chosen to adopt SAFe?"* The main reason for the adoption of SAFe in both the case organizations and the reviewed literature is that the organizations needed to scale their operation (Brenner and Wunder, 2015; Gusch and Herbai, 2015; Holdorf, 2011; Scaled Agile, 2012, 2015b; SEI Global Wealth Services; Vaje, 2014; Weltsch-Coen, 2014): the organizations employed agile methodologies at team level, but lacked coordination and alignment towards a common vision. In our case organizations, plans were made by management and leadership in a waterfallish process for up to a year at a time, and as such releases were infrequent and of low quality. Teams, especially ones distributed across different regions and cultures, suffered from all of this.

Our research supports the idea that leadership and management—from which the initiative to adopt SAFe was taken—look to SAFe since it provides such a comprehensive solution to scaling agile and especially to the organization above the team level, where they themselves operate. While it cannot be concluded from the interviews, it might be that the abundance of information and case studies about SAFe—coupled with the seeming popularity of the framework—in itself led Comptel to choose to adopt it. This was also seen in the literature review, as one prominent reason for the selection of SAFe

was that it was a "proven, public framework in widespread use" (Gusch and Herbai, 2015; Scaled Agile, 2014a, 2015b, 2016b; Weltsch-Coen, 2014). It is interesting that none of the case organizations seriously considered any other alternative to SAFe.

For NAPA, where the discovery of SAFe and its principles happened earlier and independently through the book *Agile Software Requirements* (Leffingwell, 2011), the main reasons were still the same: releases happened once per year and were of poor quality, all planning took place behind closed doors and between higher-ups, and teams were individualistic and sometimes disconnected from the customer. Independent discovery of SAFe in the reviewed literature was also somewhat prominent (Janisse, 2016; Pôle emploi, 2016; Rally Software, 2015; SEI Global Wealth Services), indicating that not all SAFe adoptions stem from higher leadership or consultant recommendations.

5.2 RQ2 – Adoption Approaches

The second research question of this thesis is *"How have the case organizations adopted SAFe?"* Here the approaches differ: NAPA's adoption was slow and gradual, with a few change agents piloting SAFe ideas and principles and adapting them to their own organization. At Comptel, both business units planned and transformed the organization according to SAFe during 2–4 months. Common for all cases was that they customized the approach to the organization, even though the means differed: at NAPA, piloting one or a few principles at a time meant they could be changed based on feedback and results; at Comptel, both adoptions were planned in detail together with consultants before the first PI planning. In all the case organizations—perhaps due to the fact that the team level was already employing Scrum and that managers and leadership felt that SAFe brought most of the changes to middle management—teams were left without formal certification training while leadership and middle management were trained, in Unit 2 more so than in Unit 1, and in Comptel more so than in NAPA. However, in both units at Comptel teams received internal training. The reviewed literature revealed a similar trend, where management and leadership training was more prominent (Holdorf, 2011; Ivar Jacobson International, 2014; Janisse, 2016; McMaster, 2014; North, 2013; Scaled Agile, 2014a, 2015b, 2016b; SEI Global Wealth Services) than formally training everyone.

5.3 RQ3 – Success Factors

"What common success factors for adopting SAFe can be identified?" was the third research question in this thesis. The most prominent success factor reported in both the reviewed literature and in the case organizations is the PI planning (Brenner and Wunder, 2015; Campbell-Pretty, 2016; Ivar Jacobson International, 2014; Lam and Raman, 2014; Pitkänen, 2015; Scaled Agile, 2012, 2015b, 2016b), called release planning at NAPA. Bringing the whole organization together, presenting and discussing the near future, planning cooperatively across all organizational levels and proactively visualizing dependencies and risks was something that convinced employees of the viability of SAFe at all levels, both in the reviewed literature and especially at our case organizations. At Comptel, most were pleasantly surprised by the planning session despite the relatively common fear that the first planning would devolve into chaos. Team members appreciated hearing about the company's strategy and the roadmaps for different product, managers liked getting a better overview of what was going to be built in the upcoming months, and leadership liked the addition of dependency and risk management.

Reacting to feedback is vital to the success of probably any organizational change, whether the change happens quickly as at Comptel or slowly as at NAPA. The importance of retrospectives and the actions taken based on these cannot be understated, and was a frequently reappearing topic in both the reviewed literature (Gusch and Herbai, 2015; Scaled Agile, 2015b, 2016c) and the case organizations. If adapting to the context and feedback is left too low on the list of priorities a feeling of stagnation and resentment is created, as seen in some interviews at Unit 1. Developing relevant metrics supported the continuous improvement at NAPA: while metrics are not an end goal in themselves, when applied correctly and when improved upon they can help support the adoption.

While consultants were mentioned quite seldom as a success factor in the interviews or the reviewed literature (Gusch and Herbai, 2015; Weltsch-Coen, 2014), they were prominent in adoptions in both literature (Cobb, 2012; Gat, 2006; Gusch and Herbai, 2015; North, 2013; Rutzen and Roy, 2014; Scaled Agile, 2014a, 2016c; SEI Global Wealth Services) and an integral part of planning the adoption and organizing the first PI planning, at least at Comptel. It seems likely that the feared chaos would have materialized without the training provided to middle management and leadership by consultants, the meticulous brainstorming and planning of the initial stages of the adoption together with consultants, and the subsequent follow-up actions by consultants. Our research suggests that securing leadership support

and training middle management, ensuring that they understand their new role and responsibilities, what is expected of them, and how the organization works in the future is key to a successful adoption in case the team level already employs agile methodologies. This is specifically supported by some case reports (Campbell-Pretty, 2016; Gusch and Herbai, 2015; Scaled Agile, 2014b).

5.4 RQ4 – Challenges

The fourth research question, *"What common challenges for adopting SAFe can be identified?"*, garnered much more answers in our study than in the reviewed literature. The most prominent challenge in literature was staffing (Gat, 2006; Holdorf, 2011; McMaster, 2014; SEI Global Wealth Services), which wasn't present as a separate challenge in our cases but rather as a success factor, indicating that our case organizations succeeded where others had failed and that there are benefits with getting the right people to each role. The second most prominent challenge in reviewed literature cases was change resistance (Janisse, 2016; McMaster, 2014; Pitkänen, 2015). As change resistance occurs in any organizational change which is not trivial, the most interesting results are the ones which lead to the creation of said resistance.

A lack of people actively driving the change, especially prominent in Unit 1, is heavily tied to a lack of continuous improvement, which in itself is a much more visible challenge. The roles of the RTE at program level and the scrum masters in teams are especially important in promoting the new ways of working. Simply coordinating and guiding teams is challenging, as reported in the literature (Gat, 2006; Scaled Agile, 2015b). If no-one facilitates—or even forces—challenging the status quo then no improvement can happen. Even if retrospectives are held, if the problems reported stay the same month after month with no apparent action taken then dissatisfaction will grow and change resistance with it, as happened at Comptel Unit 1. Unit 1's situation was not improved by a lack of communication about the adoption, leaving team members to try to figure out the reasons and rationale behind it themselves.

Not only does stagnation prove challenging to employee satisfaction and motivation, it also directly retracts from the work. Poor planning, estimation and management of work proved to be a major challenge in all case organizations, stemming especially from product managers and owners not understanding, or adhering to, the requirements set by the framework for program level work. Work spilling from one sprint to the next, and from

one increment to the next, was frequent, severe and—most alarmingly—continuous at Comptel, often leading to the IP sprint being filled up; this sprint being the one time during the increment when the team is supposed to focus solely on improving their own ways of working.

Blame for the challenges cannot be laid entirely at the feet of traditional middle management, even though the reviewed literature supports the idea that finding and staffing motivated and trained product managers and owners as well as scrum masters is highly challenging and crucial to the adoption (Gat, 2006; Holdorf, 2011; McMaster, 2014; SEI Global Wealth Services). Both the program and team levels depend on a properly functioning portfolio level, something which all case organizations initially lacked completely and still struggle with. If decisions on epics are taken too late or not at all, if no clear strategic roadmap exists for the organization, if no architectural aspects are sufficiently prioritized via architectural and technical epics, then the program level and ultimately teams will struggle to deliver consistent, reliable and valuable results.

5.5 RQ5 – Effects

"What effects has the adoption of SAFe had in the case organizations?" was the fifth and final research question. As with the reviewed literature, the benefits far outweighed the drawbacks; this might be due to a difficulty in separating negative effects of the adoption from challenges to the adoption, but all in all the tone of respondents was positive when asked to consider how they perceive the situation before and after SAFe.

In the reviewed literature, adopting SAFe led to both increased release frequency (Campbell-Pretty and Richards, 2013; Gat, 2006; Holdorf, 2011; Scaled Agile, 2014a) and improved quality (Ball et al., 2015; Campbell-Pretty, 2016; Campbell-Pretty and Richards, 2013; Gat, 2006; Gusch and Herbai, 2015; Lam and Raman, 2014; McMaster, 2014; North, 2013; Scaled Agile, 2015a, 2016b,c; SEI Global Wealth Services; Vaje, 2014). Our findings from NAPA support this: quality had improved drastically with more stable releases and fewer detected defects in production, coupled with a release frequency of three releases per year compared to one major release per year previously. The underlying factors behind this improvement are varying, as the multitude of reported other benefits are tightly interconnected. While improved quality was not reported at Comptel, the other benefits reported support the reviewed literature: improving visibility (Cobb, 2012; Gusch and Herbai, 2015; Ivar Jacobson International, 2014; Janisse, 2016; Lam and Raman, 2014; McMaster, 2014; Rutzen and Roy, 2014; Scaled Ag-

ile, 2014b; Vaje, 2014; Weltsch-Coen, 2014) and communication helps with building trust between teams (Cobb, 2012; McMaster, 2014; Pitkänen, 2015; Scaled Agile, 2014b), all of which in turn help with synchronization and collaboration when working jointly on products. Improved dependency management was mentioned especially at Comptel, while NAPA interviewees reported a more sustainable pace of development and a more disciplined use of agile practices.

The effect of SAFe on employee satisfaction at Comptel was conflicting, as it had been in the reviewed literature, but there was not enough information regarding the situation at NAPA to draw any conclusions for all organizations. At Unit 2, employee satisfaction had improved, in agreement with most of the case reports (Campbell-Pretty, 2016; Campbell-Pretty and Richards, 2013; Cobb, 2012; Gat, 2006; Holdorf, 2011; Rutzen and Roy, 2014; Scaled Agile, 2015a). At Unit 1, however, the satisfaction of employees had decreased since adopting SAFe; the only reviewed literature reporting the same was the case study by Pitkänen (2015). Interviewees from Unit 1 felt that SAFe decreased agility, which is somewhat supported by a couple of other identified drawbacks: worse handling of unplanned work, and more work which felt like overhead.

5.6 Limitations and Threats to Validity

There are several limitations concerning the literature review and the multiple case study.

Firstly, the reviewed literature showed signs of publication bias, being mostly case reports produced by the adopting organizations or consultants assisting with the adoption. The reports presented the adoptions as successes and listed far more successes and benefits than challenges and drawbacks.

As all our case organizations had previously adopted Scrum at team level, the results should be viewed while keeping in mind that a complete agile transformation would probably have faced more challenges and with different intensity than our study.

While we attempted to interview employees from across the organization the study could have benefited from more interviews from the team level. Our team level interviewees were all scrum masters referred to us by leadership and management, and the thesis lacks the viewpoint and experiences of a broad range of team members, such as senior developers and off-shore team members. A larger number of interviewees from teams would also have been warranted due to the fact that they make up most of the organization employing SAFe.

As we conducted the interviews in an open-ended fashion, the data collection exhibited significant variability, which we attempted to mitigate by conducting a large number of interviews using the same set of questions in each interview.

Chapter 6

Conclusions and Future Work

The existing body of scientific research on agile scaling frameworks is very limited, and this thesis set out to study the Scaled Agile Framework. In this qualitative multiple case study we have sought to understand why organizations choose to adopt it, how they adopt it, what success factors and challenges concerning the adoption can be identified and what effects the adoptions have on the organizations. We have gathered information through a set of semi-structured open-ended interviews at two Finnish companies.

The results of this thesis support many of the claims in the industry case reports. SAFe is adopted when organizations seek a comprehensive and proven approach to scaling their agile operations, helping to synchronize teams and align them to strategic objectives. The results seem to suggest that customizing the adoption to the organizational context, hiring external consultants to provide training—especially to managers transitioning into SAFe roles—and facilitation, and meticulously organizing the first PI planning are major success factors when adopting SAFe. Likewise, ensuring that there are dedicated internal change agents driving the adoption forward, consistently communicating about the adoption, and ensuring that higher organizational layers understand and commit to the new ways of working would seem to address the most severe challenges which the case organizations faced.

This thesis provides a base for future quantitative studies concerning the why and how of adoptions of SAFe. Studies comparing SAFe to other scaling frameworks, such as LeSS and DAD, would help create understanding of which challenges are common to all organizations aiming to become agile at scale, and which are more specific to each framework—potentially providing each framework with concrete suggestions for improvement.

Bibliography

Scott W Ambler and Mark Lines. *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*. IBM Press, 1 edition, may 2012. ISBN 0132810131.

Andrew Ball, Ajay Nair, and Mirko Hering. Key Accenture Learnings on Scaled and Distributed Agile Delivery, 2015. URL `http://www.scaledagileframework.com/?wpdmact=process{&}did=MTA2LmhvdGxpbms=`.

Kent Beck. Embracing change with extreme programming. *Computer*, 32 (10):70–77, 1999a. ISSN 00189162. doi: 10.1109/2.796139. URL `http://ieeexplore.ieee.org/document/796139/`.

Kent Beck. *Extreme Programming Explained: Embrace Change*. Number c. Addison-Wesley, Boston, MA, 2 edition, 1999b. ISBN 0201616416. doi: 10.1136/adc.2005.076794.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for Agile Software Development, 2001. URL `http://agilemanifesto.org/`.

Richard Brenner and Stefan Wunder. Scaled Agile Framework: Presentation and real world example. *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 1–2, 2015. doi: 10.1109/ICSTW.2015.7107411. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7107411`.

Em Campbell-Pretty. Banking on a SAFe QuickStart. pages 1–37, Melbourne, AU, 2016.

Em Campbell-Pretty and Mark Richards. Adopting Leffingwell's Scaled Agile Framework, the Theory and the Practice, 2013. URL `http://www.slideshare.net/ScaledAgile/agile-aus-2013final`.

Charles G. Cobb. Case Study – Valpak. pages 1–9, 2012. URL http://www.scaledagileframework.com/wp-content/uploads/2013/06/Final-Valpak-Case-Study-Short.pdf.

Comptel Oyj. Comptel Annual Report 2015, 2015. URL http://www.comptel.com/docs/default-source/annual-reports/comptel-annual-report-2015.pdf?sfvrsn=2.

Kim Dikert, Maria Paasivaara, and Casper Lassenius. Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, 119:87–108, sep 2016. URL http://www.sciencedirect.com/science/article/pii/S0164121216300826.

Torgeir Dingsøyr and Tore Dybå. What Do We Know about Agile Software Development? *IEEE Software*, 26(5):6–9, 2009. ISSN 0740-7459. doi: 10.1109/MS.2009.145.

Israel Gat. How BMC is scaling agile development. *Proceedings - AGILE Conference, 2006*, 2006:315–320, 2006. doi: 10.1109/AGILE.2006.33. URL http://www.scaledagileframework.com/?wpdmact=process{&}did=MTIxLmhvdGxpbms=.

Lars Gusch and Petrine Herbai. The trains have left the Central Station – A snapshot of Elekta's Agile transformation. pages 1–39, 2015.

Chad Holdorf. Case Study – John Deere – Scaled Agile Framework, 2011. URL http://www.scaledagileframework.com/john-deere/.

Ivar Jacobson International. Nordea – A Uniform Heartbeat with Help from Scaled Agile Framework and IJI. pages 1–5, 2014. URL http://www.scaledagileframework.com/nordea/.

James Janisse. Driving SAFe. page 32, 2016. URL http://www.scaledagileframework.com/tomtom-case-study/.

Johnny Lam and Sanjeev Raman. SKHMS: SAFe Adoption for Chip Development. pages 1–8, 2014.

Craig Larman and Bas Vodde. Large scale scrum (less), 2005. URL http://less.works/resources/about.html.

Dean. Leffingwell. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 1 edition, 2011. ISBN 0321635841.

Mikael Lundgren and Tomek Pająk. Downscaling SAFe, 2015. URL `https://www.infoq.com/articles/downscaling-SAFe`.

Maddy McMaster. Going agile in student administration projects – The view from the business side. pages 1–37, Gold Coast, AU, 2014.

Ryan North. Mitchell's Big LEAP Enterprise Transformation. pages 1–46, 2013.

Michael Quinn Patton. *Qualitative research and evaluation methods*. Sage Publications, Thousand Oaks, CA, USA, 3rd edition, 2001. ISBN 978-0-7619-1971-1.

Antti Pitkänen. *Agile Transformation: A case study*. G2 pro gradu, diplomityö, Aalto University, 2015. URL `http://urn.fi/URN:NBN:fi:aalto-201512165681`.

Pôle emploi. Case Study – Using SAFe on a program for the French national employment agency Pôle emploi. 2016. URL `http://www.scaledagileframework.com/pole-emploi/`.

Rally Software. The Travins Perkins Group Improves Business Value of IT. pages 1–3, 2015. URL `http://www.scaledagileframework.com/travis-perkins/`.

Mark Richards. A is for Agile, not Anarchy: Scaled Agile Framework Applied, 2013. URL `http://www.agilenotanarchy.com/2013/01/scaled-agile-framework-applied-part-15.html`.

Allen Rutzen and Sunil Roy. Waterfall to Agile – A Case Study. Number February, page 16, 2014. URL `http://www.scaledagileframework.com/from-waterfall-to-enterprise-agility/`.

Scaled Agile. Case Study – Swisscom – Scaled Agile Framework, 2012. URL `http://www.scaledagileframework.com/swisscom-case-study`.

Scaled Agile. Case Study – Amdocs – Scaled Agile Framework, 2014a. URL `http://www.scaledagileframework.com/amdocs-case-study`.

Scaled Agile. Case Study – Infogain – Scaled Agile Framework, 2014b. URL `http://www.scaledagileframework.com/infogain-case-study/`.

Scaled Agile. Case Study – Cisco Case Study – Scaled Agile Framework, 2015a. URL `http://www.scaledagileframework.com/cisco-case-study/`.

Scaled Agile. Case Study – LEGO – Scaled Agile Framework, 2015b. URL http://www.scaledagileframework.com/lego-case-study/.

Scaled Agile. About – Scaled Agile Framework, 2016a. URL http://www.scaledagileframework.com/about/.

Scaled Agile. Case Study – AstraZeneca – Scaled Agile Framework, 2016b. URL http://www.scaledagileframework.com/astra-zeneca-case-study/.

Scaled Agile. Case Study – HPE – Scaled Agile Framework, 2016c. URL http://www.scaledagileframework.com/hpe-case-study/.

Scaled Agile. Implementing – Scaled Agile Framework, 2016d. URL http://www.scaledagileframework.com/implementing/.

Scaled Agile. Value Stream Level – Scaled Agile Framework, 2016e. URL http://www.scaledagileframework.com/value-stream-level/.

Ken Schwaber and Jeff Sutherland. The Scrum Guide. *Scrum.Org and ScrumInc*, (July):17, 2013. ISSN 00195847. doi: 10.1053/j.jrn.2009.08.012. URL http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf.

SEI Global Wealth Services. Case Study: A Launch in Two Weeks. URL http://www.scaledagileframework.com/sei/.

Toivo Vaje. NAPA Agile Story: From Zero to Hero in Two Years. pages 1–24, 2014.

VersionOne. The 10th annual State of Agile Report, 2016. URL https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf.

Kim Wallis. *Tillämpning av ramverket Scaled Agile Framework i organisationer*. Bachelor's thesis, Aalto University, School of Science, 2016.

Yariv Weltsch-Coen. Implementing SAFe: MDO (Intel) test case. page 28, 2014. URL http://www.scaledagileframework.com/wp-content/uploads/2014/09/Implementing-SAFe-MDO-test-case.pdf.

Robert K. Yin. *Case Study Research : Design and Methods*. Sage Publications, Thousand Oaks, CA, USA, 4th edition, 2009. ISBN 978-1-4129-6099-1.

Appendix A

Code Families

Table A.1: Code Families

Code Family	Codes
RQ1 - Dependency management was challenging	Dependency management was challenging
RQ1 - Lack of portfolio level	Business environment mandated change, Lack of systematicness in portfolio level, Needed prioritization transparency, No top-level prioritization, Organization and planning in silos, Product management in waterfall, Wanted portfolio level as defined by SAFe
RQ1 - Lack of supportive structure above team level	Chaotic processes, More predictability, Needed cohesion, Needed structure, PO's not enough to handle both teams and customers, SAFe also provides agility to rest of organization, Scrum not enough
RQ1 - Lack of transparency	Disconnect between daily work and company/product vision, Disconnect between teams and middle management, GSD challenging, More visibility to other parts of organization, Needed transparency, Understanding of the bigger picture on team level
RQ1 - Misc	Agile methodology already in use at team level, Wanted better quality

CONTINUED ON NEXT PAGE

Code Family	**Codes**
RQ1 - SAFe was deemed a "good fit"	Discovered SAFe internally first, Existing organizational parts fit into SAFe, Leadership mandate for change, Lots of information available, SAFe also provides agility to rest of organization, SAFe in widespread use, Wanted portfolio level as defined by SAFe
RQ1 - Wanted capability for more frequent releases	Fixed scope releases not optimal, Need for simultaneous delivery across organization (scaled agile), Needed better synchronization and collaboration between teams, Needed more reactivity, less up-front planning, Needed shorter cycles, Wanted better quality, Wanted continuous delivery instead of project-based, Wanted internal cadence, Wanted more frequent releases
RQ2 - Adoption approach	Big bang adoption, Bottom up adoption, By the book, Quick ramp-up, Top-down approach
RQ2 - Cross-functional month	Cross-functional month, IP sprint used for release stabilization, IP work done during sprints, No IP Sprint
RQ2 - Customization	Adapted portfolio level, Constant reprioritization, No formalized vocabulary, SAFe handbook created in-house, SAFe practices modified throughout the adoption, Teams allowed to customize
RQ2 - Demo	Joint sprint demo, No system demo at end of PI, System (release) demo at end of PI, Workshop instead of system/release demo, Yammer videos instead of public sprint demos
RQ2 - Dependency management	Dependency matrix, No visible dependency management
RQ2 - Estimation	No estimation during PI Planning, Some teams don't estimate at all, Story point normalization
RQ2 - Gradual adoption	Began with joint plannings before SAFe, Learned from other organization, SAFe adoption only in developmental organization, SAFe in use by other sub-organization, Team and Program levels working before Portfolio level
RQ2 - In-house evangelism	Designated change agents, In-house agile coach, In-house introduction for teams, Local agile evangelist group, Scrum master from team

CONTINUED ON NEXT PAGE

Code Family	**Codes**
RQ2 - Leadership & Management	Gain leadership buy-in first, Middle management doesn't care about SAFe, No leadership or portfolio training, Top-down approach
RQ2 - Limited communication	Didn't call it SAFe in the beginning, Limited communication about change, Limited communication about SAFe adoption, No formalized vocabulary, SAFe not explicitly mentioned anywhere
RQ2 - Metrics & Follow-up	Created team and unit targets affecting compensation, Metrics, New KPI for measuring SAFe progression, No KPI's or goals for SAFe adoption, Program-level weekly follow-up, Scrum master roundtable, Weekly Scrum of Scrums, Weeky portfolio board meeting
RQ2 - Misc	JIT requirements specification, New tools for communication, No PI Planning retrospectives, No separate portfolio level, Separate PO and technical PO, Site-specific retrospectives, Team retrospectives mandatory, Team structure, Time zone adaptations, Yammer videos instead of public sprint demos
RQ2 - No training prior to adoption	Middle management trained after adoption, No leadership or portfolio training, No team-level training, Training after adoption
RQ2 - PI Planning	Adapted PI Planning, Joint PI Planning across sites, Joint retrospective after PI Planning, No estimation during PI Planning, PI planning, PI Planning via Skype, Portfolio and program level items defined before first PI, Time zone adaptations, Dependency management
RQ2 - Release cycle different from PI cycle	PI cycle, Release according to different plan
RQ2 - RTE	External agile coaching for RTE, No RTE, Recruited full-time RTE internally, SAFe roadshow
RQ2 - Sprint synchronization	Flexible sprint length, Synchronized sprints (cadence), Teams allowed to customize, Teams decided sprint length, Unsynchronized sprints

CONTINUED ON NEXT PAGE

Code Family	**Codes**
RQ2 - Training	External agile coaching for RTE, Further training later on, In-house agile coach, In-house introduction for teams, In-house training for teams, On-site training for off-site personnel, On-site training for off-site PO, SAFe certification for middle management, Training for middle management
RQ2 - Use of consultants	Consultant recommendation, External agile coaching for RTE, External consultancy, SAFe certification for middle management, Training for middle management
RQ2 - Whole organization in one ART	Whole organization in single ART
RQ3 - Appropriate staffing	Full-time RTE with PO&PM experience, Good people in team level, Good RTE, Motivated scrum master
RQ3 - Consultants & Training	Audit after adoption, Consultancy, External agile coach for middle management, On-site training for off-site PO, SAFe refresher training, Training for middle management
RQ3 - Customization	Cross-functional month, Customized SAFe, Joint sprint demo useful, Teams allowed to customize
RQ3 - Empowered agile teams	Team empowered, Team level was already agile
RQ3 - Internal change agent	Internal change agent
RQ3 - Leadership support for adoption	Gain leadership buy-in first, Gain leadership buy-in first, Leadership support, Leadership support for middle management
RQ3 - Misc	Agile Release Train concept, DevOps helps quality, Facilitation works, Getting requirements from customer, Quick adoption, Scrum of Scrums, Sprint reviews force teams to communicate with stakeholders, Surprisingly little change resistance
RQ3 - PI Planning	Breakfast during PI Planning, Dependency management, First PI Planning, Joint PI Planning, Teams groomed backlogs before PI Planning

CONTINUED ON NEXT PAGE

Code Family	**Codes**
RQ3 - Retrospectives	Focused retrospectives, Retrospectives impact practices, Separate team retrospectives, Team retrospectives mandatory, Team retrospectives work
RQ3 - Use of appropriate tools and metrics	Live video during meetings, Same tool (JIRA) in use throughout organization, Simply beginning to measure led to improve metrics, Yammer works for asian culture
RQ4 - Change resistance	Middle management doesn't care about SAFe, Change resistance, Change resistance in middle management, Pessimists can poison entire teams, SAFe is "heavy" for newcomers, SAFe seen as overhead, Skepticism in upper leadership
RQ4 - Cultural differences	Cultural differences, Cultural differences challenging to cross-functionality
RQ4 - Estimation, planning and management of work	Backlog items not groomed enough before planning, Estimation, IP Sprint filled in planning already, IP Sprint used for spilled stories, Less time for development backlog, Management of spikes, No buffer for ad hoc work, Overcommitment, PI commitments too big, Pressure to overcommit from middle management, Sporadic task forces
RQ4 - Lack of communication	Communication with customers, Lack of knowledge about SAFe, Lack of rationale for PI Planning, Little vertical communication, No rationale for SAFe, Not enough communication about adoption to teams, Not enough evangelism, Not enough evangelism at team level, Obsession over terms (SAFe, Scrum, etc.), Teams don't want to communicate
RQ4 - Lack of continuous improvement	Demos previously more relevant, Joint retrospectives with all teams not working, Lack of continuous improvement, Lack of discipline, No power to change due to retrospectives, No reaction to retrospective results, Retrospectives, Retrospectives skipped due to workload, Risk management unclear, Scrum master rotation, Scrum Master seen as administrative role, Unmotivated Scrum master
RQ4 - Lack of drivers of change	No agile coach, No internal change agent, No RTE, Unmotivated Scrum master
RQ4 - Lack of training or support	Insufficient training, Middle management trained after adoption

CONTINUED ON NEXT PAGE

Code Family	**Codes**
RQ4 - Misc	Avoiding responsibility, Demos previously more relevant, Dependencies not raised proactively before PI Planning, Difficult to customize, Fixed scope for releases sometimes meant lowered quality, Increments perceived as waterfall, Initially lack of focus for common goal, Integrated system demo challenging, Joint retrospectives with all teams not working, Low cross-functionality, Low predictability of releases, Releases not in sync with PIs, Risk management unclear, Scrum of Scrums, Scrum of Scrums used as reporting session, Teams might not see bigger picture, Teams too busy to adopt agile culture
RQ4 - Organizational challenges	Avoiding responsibility, Can't collocate everybody, Commitment to 2-day PI Planning from middle management, HR processes impairs use of SAFe, Lack of clear goal of adoption, Logistics, Low cross-functionality, No separate portfolio level, Non-agile or Non-SAFe budgeting impairs use of SAFe, Not all teams in ART interdependent, Rest of organization not in SAFe, Stakeholders can't participate in every separately held demo, Team forming, Tools unsuited for SAFe, Whole business unit in a single ART
RQ4 - PI Planning	Organizing first PI Planning very laborious, PI Planning, PI Planning eventually boring, PI Planning feels like overhead, PI Planning feels rushed, PI Planning not very interactive for team members, SAFe is becoming ritualistic
RQ4 - Poor technical base	Haphazard code base, Lack of test automation, Low DevOps, automatization and/or CI, Releases not in sync with PIs, Tools unsuited for SAFe
RQ4 - Portfolio management and prioritization challenging	Large gap between business requirements and technical implementation, No know-how to groom backlog items before team, No portfolio level architect, No separate portfolio level, Portfolio management, Prioritization is ad hoc and person-dependent, Product portfolio management and standardization, Roadmap changes done too close to PI Planning, Teams see backlog items for the first time during planning
RQ4 - Restrictive business environment	Business environment restricts use of SAFe/agile

CONTINUED ON NEXT PAGE

Code Family	**Codes**
RQ5 - Better agile practices	Benefit - Better agile practices, Benefit - Daily work practices improved, Benefit - Every team got a PO, Benefit - More comprehensive approach to software development, Benefit - Scrum of Scrums improved, Benefit - Teams have more freedom, Democratized decision making
RQ5 - Better alignment across the organization	Benefit - Better alignment, Benefit - Product portfolio more in line with company strategy, Benefit - Understanding of company/product vision improved, Benefit - Understanding of link between own work and strategy improved, Benefit - Understanding of own role improved, Benefit - Understanding of value creation improved
RQ5 - Better communication	Benefit - Better stakeholder communication, Benefit - Cross-team communication improved, Benefit - Increased communication
RQ5 - Better dependency discovery and management	Benefit - Dependency discovery and management improved
RQ5 - Better quality and DevOps practices	Benefit - Improved CI practices, Benefit - Improved software quality, Benefit - SAFe process mandated implementation of DevOps, Benefit - Understanding of importance of DevOps improved
RQ5 - Better synchronization & collaboration across teams	Benefit - Collaborative approach, Benefit - Cross-functionality easier, Benefit - Cross-team collaboration improved, Benefit - Development synchronization across teams improved
RQ5 - Better visibility	Benefit - Improved visibility across organization
RQ5 - Better, more frequent and more predictable releases	Benefit - Better predictability, Benefit - Fewer mid-sprint/mid-PI changes, Benefit - Less plan-driven, Benefit - Less unplanned work, Benefit - More frequent releases, Benefit - More structured releases, Benefit - Prioritization improved, Benefit - Quicker delivery, Benefit - Responsiveness to change, Benefit - Shift to portfolio-level thinking
RQ5 - Less autonomy for teams	Drawback - Lack of autonomy for teams

CONTINUED ON NEXT PAGE

Code Family	**Codes**
RQ5 - Less communication	Communication with customers, Conflicting experiences in teams, Drawback - Decreased communication between teams and upper levels, Drawback - Decreased communication between teams and upper levels (PM), Drawback - Less customer interaction for teams
RQ5 - Misc benefits	Benefit - IP Sprint nominally gives time for self-improvement, Benefit - Project management development process improved, Benefit - Synergies with other parts of organization
RQ5 - More overhead	Drawback - More administrative overhead for Scrum master, Drawback - More visibility not always welcome, Drawback - More work for technical PO, Drawback - PI Planning overhead for many participants, Drawback - Program backlog seen as more bureaucracy
RQ5 - Work satisfaction decreased	Drawback - Work satisfaction decreased
RQ5 - Worse handling of unplanned work	Less time for development backlog, Low predictability of releases, Management of spikes, No buffer for ad hoc work, Sporadic task forces, Drawback - SAFe doesn't handle ad hoc work well, Drawback - Things outside the cadence get less attention
RQ5 - Worse quality	Fixed scope for releases sometimes meant lowered quality

Appendix B

Interview questions

This appendix includes the questions used to guide the semi-structured interviews.

Interviewee role and background

- What is your current role?
- How long have you been in this project role?
  - Have you been in the project since before SAFe?
  - Have you been in the same role since before SAFe?
- How many people are working in your team?
- What is the team structure and role division?
  - What is the skill level of team members?

Project

- What is the size of your organization?
- What type of project are you in?
  - Complexity
  - Criticality
  - Size
- How large is the project?
  - How many teams and people?

  - How has it changed?
- How was the project managed?
  - How were plans managed before starting SAFe?
  - Did the plans/roadmap adapt to challenges/successes?
- Are all team members in the same location?
  - Do you feel this has affected the adoption? How?
- Could you draw a picture of the project and surrounding organization?

Reasons

- What did you use before SAFe?
- When did your organization start implementing SAFe?
- For what reasons did your organization select SAFe?
  - Did you select it yourselves?
- Did you explore any alternatives?
  - Which ones?
  - Why was SAFe chosen over them?
- What problems did you face when choosing to adopt SAFe?
- What benefits were you hoping to bring about with SAFe?
- Did you have any reservations about SAFe or agile?
- Was there a driving force, a change agent, behind the adoption?
  - From the organization or external?
  - How do you feel this affected the adoption?

Beginning

- What was your starting point?
  - What was the development method/process maturity?
  - At what level was your understanding of agile?
- Do you feel your business environment affected the adoption?
- How did the implementation start?
  - Piloting in small team, or all at once?
  - Formal training first, or as you went along?
  - How was the change communicated?
    - How was it received? How did you feel about it?
    - Alignment and involvement of employees
    - Common objectives (product vision etc.)
  - How did you adjust the work load during the adoption?
    - Did you feel there was pressure to deliver as much as usual even though you were in the middle of the transformation?
- Did you view the adoption as a project or as continuous improvement?
- Did you implement SAFe "by the book"?
  - Did you do changes or develop your own ways of working from the start, or later on?
  - What differences to the handbook did you have?
  - How has self-organization worked?
- Did you employ an agile coach?
  - External or internal?
  - Is he or she still active? Or just in the beginning?
  - What did the coach do?
  - Do you feel it was useful to have a coach? Why?
- Change resistance
  - How did you originally feel about SAFE and agile?

  - What were the attitudes of your co-workers / managers / employees?
  - How was the change management done?
- Management support
  - How has management supported the implementation?
  - Does management exhibit skepticism?
  - Agile "on paper" vs reality?
- How has the surrounding organization affected the adoption of SAFe?
  - Theory vs organizational reality (HR requirements on roles and staffing, legal requirements, upper management lack of understanding, pressure)
- How do you measure the success of the transformation?
  - What kind of objectives did you use?
  - What kind of metrics?

Current state

- Where are you currently regarding the adoption?
- What agile practices other than those advocated by SAFe are you using, if any?

Effects

- Has the adoption caused changes in the surrounding organization / organizational structure / behavior?
  - For the better or worse?
  - Has the organization become more agile / flexible?
- What benefits has the adoption of SAFe brought?
  - Improved quality
  - Fewer defects
  - Faster TTM
  - Employee engagement

  - Alignment

- What drawbacks does SAFe have?
  - "Chaos"?
  - More work? E.g. double reporting, rigid production processes

Future

- What are the next steps in the adoption?
- Where is the organization going?
  - Will you continue with SAFe?
  - Will you use some other method with SAFe?

Challenges

- How do you feel the adoption went/is going?
  - What difficulties did you encounter?
    * How did you react? (Retros, I&A)
    * What did you change?
  - Change resistance
  - Unfamiliarity of agile
  - Fall back to old ways?
- Did you notice any practices which didn't work so well?
  - Which ones? Why?
  - Did you change anything based on this?
- Has anything changed for the worse since the adoption of SAFe?

Successes

- What went well in the adoption of SAFe?
  - Were there some aspects which inspired people (more than others)?
    * What was quickly adopted?

- What benefits has SAFe brought?
  - How are these measured? KPI's: Improved delivery, software quality, etc.

Opinions

- What is your opinion about SAFe?
- How is the organization doing in your opinion? Are you better or worse off after SAFe?
- Software quality
  - What is the state of your software now (vs before implementation of SAFe)?
- Do you feel you are truly agile?
- Would you do anything differently?

Advice

- Do you have any advice for other people and companies intending to adopt SAFe?