Department of Computer Science

Adopting Agile Methods in Large-scale Organizations using Scaling Frameworks

Abheeshta Putta

Aalto University

DOCTORAL
THESES

Aalto University publication series
DOCTORAL THESES 155/2022

Adopting Agile Methods in Large-scale Organizations using Scaling Frameworks

Abheeshta Putta

A doctoral doctoral thesis completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at Otakaari 1, U154 U1 on 18th November 2022 at 12 noon.

**Aalto University
School of Science
Department of Computer Science
Software Process Research Group**

Supervisor

Prof. Casper Lassenius

Thesis Advisor

Prof. Maria Paasivaara

Preliminary Examiners

Prof. Markku Oivo, University of Oulu, Finland

Prof. Eric Knauss, Chalmers and University of Gothenburg, Sweden.

Opponent

Prof. Xiaofeng Wang, Free University of Bozen-Bolzano, Italy.

Aalto University publication series
DOCTORAL THESES 155/2022

ISBN 978-952-64-0994-8 (printed)
ISBN 978-952-64-0995-5 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-64-0995-5

Unigrafia Oy
Helsinki 2022

Finland

NORDIC SWAN ECOLABEL
Printed matter
4041-0619

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Abheeshta Putta

Name of the doctoral thesis
Adopting Agile Methods in Large-scale
Organizations using Scaling Frameworks

Publisher School of Science

Unit Department of Computer Science

Series Aalto University publication series DOCTORAL THESES 155/2022

Field of research Software Engineering

Manuscript submitted 6 June 2022 **Date of the defence** 18 November 2022

Permission for public defence granted (date) 5 October 2022 **Language** English

□ **Monograph** ☒ **Article thesis** □ **Essay thesis**

Abstract

Agile methods were originally developed for small and co-located teams. The popularity and success of agile methods in small teams led to growing interest on agile adoption across large organizations as well. However, there are several challenges while adopting agile to large, e.g., coordination between large number of teams and integration of other non-development units e.g., HR, and marketing. Scaling frameworks, e.g. Scaled Agile Framework (SAFe) and Large Scale Scrum (LeSS) to support scaling agile to large have become popular in the recent past. Despite of popularity, there is very little scientific research on usage of the scaling frameworks.

The primary goal of the thesis is to investigate the adoption and usage of scaling frameworks in practice. The goal is divided into two parts: a) scaling frameworks usage and adoption and b) SAFe usage and adoption. In the first part, we conducted two surveys. The first survey aimed to explore why the frameworks were developed, and how they were evolved, their benefits and challenges directly from the practitioners who developed them. Later, in second survey, we collected data from 204 software practitioners using scaling frameworks to understand the reasons, expected benefits and satisfaction of using them. In the second part, we conducted a multivocal literature review (MLR) due to the lack of scientific evidence on SAFe, to understand the benefits and challenges of SAFe adoption. Next, we conducted an in-depth case study to explore the reasons, transformation process, benefits and challenges of SAFe. To get a wider overview of the benefits and challenges of SAFe we conducted a survey, to explore the benefits and challenges of SAFe.

Our results for the first part show that majority of the frameworks were designed to improve agility, collaboration, coordination, and synchronization between agile teams. The most common reasons for their adoption were to scale more people and deal with existing challenges and pain points. The benefits of adopting these frameworks were categorized into to business, product, organizational, and culture and the challenges were categorized to implementation, organizational, and scope. Our results for the second part show that reasons for SAFe adoption are related to organizational, business, and framework-specific. SAFe transformation activities typically map with the SAFe roadmap activities.
The most common benefits of SAFe adoption are improved transparency, collaboration and faster time to market. The most significant challenges of SAFe adoption are identifying value streams and forming ARTs, change resistance, and inculcating an agile mindset. More in-depth research on scaling frameworks is needed to establish the effectiveness of their usage in practice. We encourage researchers to conduct in-depth case studies on their usage and adoption.

Keywords Agile methods, agile scaling frameworks, SAFe, large-scale organizations.

ISBN (printed) 978-952-64-0994-8 **ISBN (pdf)** 978-952-64-0995-5

ISSN (printed) 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2022

Pages 236 **urn** http://urn.fi/URN:ISBN:978-952-64-0995-5

Dedicated to the Almighty

Preface

I thank my supervisors, Prof. Casper Lassenius and Prof. Maria Paasivaara for their advice, feedback and support during my doctoral studies.

I express my deep gratitude to Prof. Kai Petersen (my master thesis supervisor) for being an inspiration for pursuing this doctoral studies. His encouragement and support has helped to me stand where I am today, both in my career and personal life. I want to convey my special thanks to Prof. Pekka Abrahamsson for making me believe in myself during tough times and laying a path for my career in industry. I would like to thank Ömer Uludağ for a fruitful research collaboration over the past few years.

I would like to thank Prof. Marjo Kauppinen for being there for me, especially at the beginning of my doctoral studies and special thanks to Prof. Marko Nieminen for his support during end of my doctoral studies. I would like to thank my colleagues and friends, Nagadivya, Jose, and Anar for the good company from the beginning of my journey in Finland. My deep gratitude to Kiruthika Srinivasan for giving strength and courage to face the challenging situations and helping me to become what I am today. I want to thank Rama Krishna Sir, Anitha, Ramesh sir and Lalitha for good times and continuous support in past years. I never felt I was away from home when I was with them. I want to thank Dr. Deepika Badampudi for being my academic mentor.

Finally, I would like to express my gratitude to my parents, Dr. Mrunalini Talla and Dr. Damodhar Putta for their support, motivation and encouragement throughout my life, I would be certainly not here without them. They are the primary source of inspiration for my research career and role models. I would like to thank my brother, Advaith Putta, for his moral support and fun times. I would like to thank my cousins Lavanya, Divya, Srinivas, Dr. Sunitha, Dr. Shubha, Kapil, Sandeep, Shiva Teja, Prathyusha and Dr. Sthithi for their belief in me. I would like to thank my mother-in-law (Aliveni), father-in-law (Subrahmanyam), and Haritha for their support. I would express my deepest gratitude to my dear husband, Dr. Sai Prashanth Josyula, for always holding me during the ups and downs of this roller coaster journey, *"The PhD"*. Lastly, I would like to pay my tribute to the sacred syllable *"a-u-m"*.

Helsinki, October 6, 2022,

Abheeshta Putta

Contents

List of Publications

This thesis consists of the following publications which are referred in the text by their respective Roman numerals.

I **Evolution of the Agile Scaling Frameworks**
Ömer Uludağ, Abheeshta Putta, Maria Paasivaara, Florian Mattes
Proceedings of International Conference on Agile Software Development (XP), 2021, pp. 123-139

II **Why Do Organizations Adopt Agile Scaling Frameworks?— A Survey of Practitioners**
Abheeshta Putta, Ömer Uludağ, Shun-Long Hong, Maria Paasivaara, Casper Lassenius
Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2021, Article No.: 26, Pages 1–12.

III **Benefits and Challenges of Adopting the Scaled Agile Framework (SAFe): Preliminary Results from a Multivocal Literature Review**
Abheeshta Putta, Maria Paasivaara, Casper Lassenius
Proceedings of International Conference on Product-Focused Software Process Improvement (PROFES), 2018, pp. 334-351

IV **How are Agile Release Trains Formed in Practice? A Case study in a Large Financial Corporation**
Abheeshta Putta, Maria Paasivaara, Casper Lassenius
Proceedings of International Conference on Agile Software Development (XP), 2019, pp. 154-170

V **SAFe Transformation in a Large Financial Corporation**
Abheeshta Putta, Maria Paasivaara, Casper Lassenius
Under Revision in Empirical Software Engineering Journal

VI **Benefits and Challenges of Adopting SAFe-An Empirical Survey**
Abheeshta Putta, Ömer Uludağ, Maria Paasivaara, Shun-Long Hong
Proceedings of International Conference on Agile Software Development (XP), 2021, pp. 172-187

Author's Contribution

DC= Doctoral Candidate (Abheeshta Putta)

Co-Authors:

A= Maria Paasivaara

B= Casper Lassenius

C= Ömer Uludağ

D= Shun-Long Hong

Publication I: Evolution of the Agile Scaling Frameworks

DC was the second author of this paper. DC and C both were the main ones responsible for the study. DC collected around 20% of data, analyzed the data for three research questions (RQ) (study consists of four RQs) and wrote the research methodology, limitations, and results and discussions of three RQs in the article. C and D collected 80 percent of the data and designed the survey questionnaire. C analysed data for RQ1, and wrote results and discussions for RQ1 and was involved in analysis of other three RQs. A commented on final article and provided suggestions for different sections.

Publication II: Why Do Organizations Adopt Agile Scaling Frameworks?— A Survey of Practitioners

DC was the first author and main responsible of the study. DC collected majority of the data (around 70%), analyzed the data, and wrote methodology, results and discussions, abstract, and few parts of introduction and conclusion. D and C collected 30% of the data. C wrote some parts of introduction, conclusion, and complete related work section. A and B participated in the study design and commented on other parts of the work.

Publication III: Benefits and Challenges of Adopting the Scaled Agile Framework (SAFe): Preliminary Results from a Multivocal

Literature Review

DC was the first author and the main responsible for the study. DC analyzed all the included studies in the MLR, and wrote all the sections of the article. A and B were involved in the analysis, study design, and commented on the article.

Publication IV: How are Agile Release Trains Formed in Practice? A Case study in a Large Financial Corporation

DC was the first author and the main responsible for the study. DC and A collected the data together. DC analyzed the whole data and wrote all the parts of article. A and B participated in the study design, analysis and commented on different parts of the article.

Publication V: SAFe Transformation in a Large Financial Corporation

DC was the first author and the main responsible for the study. DC and A collected the data together. DC analyzed the whole data and wrote all the parts of article. A and B participated in the study design, analysis and commented on different parts of the article.

Publication VI: Benefits and Challenges of Adopting SAFe-An Empirical Survey

DC was the first author and the main responsible for the study. DC collected majority of the data (around 70%) and wrote all the parts of the article. D and C collected 30% of the data. A and C commented on different parts of the article.

1. Introduction

"Well begun is half done."
-Aristotle

1.1 Background

Software has become an integral part of many products and services. Several software development methodologies came to light over the years. The methodologies continuously evolved to meet the needs of customers and technological innovations. There has been a shift from traditional methods, e.g., plan-driven, to agile methods to handle fast-changing market demands in recent years [Hirsch, 2005]. Since the early 2000s, agile methods have been popular in the software industry [Boehm et al., 2004] and research studies have reported several benefits (e.g. productivity) of their usage and adoption [Diebold and Mayer, 2017, Vijayasarathy and Turk, 2008, Phalnikar et al., 2009, Stadler et al., 2019].

The original principles and practices of agile methods are suitable for small and co-located teams [Boehm and Turner, 2005]. To leverage the potential benefits also in large teams and systems, many organizations started to scale agile [Rolland et al., 2016, Dikert et al., 2016]. Scaling agile to large can be defined as [Fuchs and Hess, 2018a]: (i) agile methods employed by more number of people or teams (e.g., hiring more employees to extend the software development), (ii) expansion of agile methods to other units within the organization (e.g., applying agile to other units such as business, marketing, finance), and (iii) deepening the application of agile methods (e.g., integrating agile practices from different agile methods). In our thesis, we focus on the first two definitions (i.e. (i) and (ii)) of scaling agile.

In the current literature, there is no exact agreement on what counts as large in large scale agile [Rolland, 2016]. The definition of large is primarily dependent on the context, and the person defining it [Beecham et al., 2021]. In the literature, the term large is characterized by several dimensions: the number of teams, team size, complexity, lines of code, time duration, and cost of the project [Dikert et al., 2016]. For instance, Torgeir et al. [Dingsøyr et al., 2014a], defined large as “2 or more teams" and Dikert el at. [Dikert et al., 2016] defined large as at least six teams or more than 50 people. Other interpretations of large described by Fuchs and Hess are [Fuchs and Hess, 2018b]: a) the use of agile methods in large organizations, b) adoption of agile methods in large projects or large teams, c) application of agile methods in large multi-team settings, and d) adoption of agile practices and principles in the organization as a whole, popularly known as enterprise agile [Dingsøyr et al., 2014b]. In our thesis, we did not define a specific number for large. However, we described large similar to the Fuchs interpretation of large as defined in (a), (b), and (d) above.

Scaling agile to large is challenging as it requires additional mechanisms for coordination, alignment, and collaboration to handle multiple teams or a large number of people [Dikert et al., 2016]. Large teams or organizations have several dependencies between the teams and projects, requiring more formal documentation and reducing agility [Lindvall et al., 2004]. In a large organization, the development teams need to coordinate with other organizational units such as business, HR which often do not work in an agile way [Boehm and Turner, 2005]. For instance, the HR unit may want to have strictly specified roles in projects [Lindvall et al., 2004], and the control board may oppose the use of continuous integration or refactoring [Boehm and Turner, 2005]. Therefore, the agile practices need to be modified or tailored based on the needs of the individual, organizational units [Boehm and Turner, 2005]. Agile adoption is more than the adoption of practices, and it is more of cultural and mindset change [Klünder et al., 2018]. Transforming the culture and traditional mindset is difficult during scaling agile in large organizations that are typically waterfall-driven [Dikert et al., 2016, Dingsøyr et al., 2012]. Regardless of the above challenges of scaling, there is an increasing trend of agile adoption in large organizations [Edison et al., 2021, Uludag et al., 2020].

Several agile scaling frameworks (e.g. Scaled Agile Framework (SAFe) [Leffingwell, 2007], Large-Scale Scrum (LeSS) [Larman and Vodde, 2010], Nexus [Framework,], and Spotify [Kniberg and Ivarsson, 2012]) started to emerge in order to support the scaling of agile methods to large organizations. These frameworks were developed by some custodians of existing agile methods and also by the practitioners who have worked to scale agile methods in large companies. For instance, the co-creator of the Scrum model developed Nexus [Framework,]. Most of the scaling frameworks have claimed to address the pressing issues related to scaling, such as aligning multiple teams and improving coordination and collaboration with other organizational functions such as human resources (HR), marketing, legal, and finance [Edison et al., 2021]. However, how far the scaling frameworks have addressed these issues is still unclear from the existing literature [Beecham et al., 2021, Conboy and Carroll, 2019a].

Researchers and practitioners started to conduct workshops focusing on scaling agile since 2013 at the International Conference on Agile Software Development (XP) [Moe et al., 2016, Moe and Dingsøyr, 2017, Bass, 2019, Bass and Salameh, 2020]. The workshops encouraged researchers and practitioners to do more in-depth research to extend the understanding of how to scale agile methods. Since then, the number of research-based studies in large scale and also agile scaling frameworks have increased. For instance, Dikert et al. [Dikert et al., 2016] in 2016 identified only six primary studies on large scale and no studies on scaling frameworks. Whereas Edison et al. [Edison et al., 2021] identified 191 primary studies on large scale agile across 134 organizations in 2021. Forty-one studies of out of those, 191 studies reported the use of scaling frameworks. SAFe alone was used in 19 organizations with 21 primary studies. More studies on agile scaling frameworks confirm growing interest in the research community to understand their usage and adoption.

Even-though research-based studies on scaling frameworks have increased in recent years (0 in 2016 [Dikert et al., 2016] to 41 in 2021 [Edison et al., 2021]), the existing literature on evolution, adoption, and usage of agile scaling frameworks is still scarce or inconclusive [Das and Gary, 2021, Beecham et al., 2021]. For example, the literature lacks research on why scaling frameworks are adopted, how satisfied the practitioners are with the use of scaling frameworks, how the scaling framework mitigates the scaling-related challenges, and the benefits and challenges of their adoption.

Investigating the benefits, challenges, reasons of scaling frameworks usage in one study enables us to provide an extensive overview to practitioners or researchers who want to examine challenges, benefits, or reasons. Such an overview could guide practitioners in selecting a suitable framework for their organization and understanding the common adoption challenges. Understanding the challenges of implementing scaling frameworks can prepare practitioners for the transformation journey and look for mitigation strategies for the obvious challenges related to scaling. The identified gaps in the research could be a starting point for researchers in creating a more in-depth analysis of scaling frameworks. Such future studies will hopefully help in creating a new body of knowledge in the domain of large-scale development, which can lead to more efficient outcomes compared to the present ones.

1.2 Research Problem and Questions

Scientific research exploring the usage and adoption of agile scaling frameworks is still limited. Therefore, in this thesis, we aim to understand the research problem: *Understanding the adoption and usage of agile scaling frameworks*. Both practitioners and researchers are interested in understanding why frameworks are adopted, how organizations transform to scaling frameworks, and the benefits and challenges of their usage [Moe et al., 2016, Moe and Dingsøyr, 2017, Dingsøyr et al., 2018].

In order to achieve the overall aim, we formulated two goals (G1 and G2) and related research questions below.

- **Goal 1 (G1): Understand the evolution, adoption, and usage of agile scaling frameworks.**
  - RQ1: What are reasons behind designing the agile scaling frameworks?
  - RQ2: How did the agile scaling frameworks evolve over the years?
  - RQ3: Why do organizations adopt agile scaling frameworks?
  - RQ4: What are the benefits and challenges of adopting agile scaling frameworks?
  - RQ5: How satisfied are practitioners after adopting agile scaling frameworks?
- **Goal 2 (G2): Understand the adoption and usage of SAFe.**
  - RQ6: Why do organizations adopt SAFe?
  - RQ7: What activities are involved in a doing a SAFe transformation?
  - RQ8: What are the benefits of adopting SAFe?
  - RQ9: What are the challenges of adopting SAFe?

The first goal of the thesis is broader, where we study scaling frameworks as a phenomenon, providing a more comprehensive overview of various agile scaling frameworks. In the second goal, we narrowed our focus to

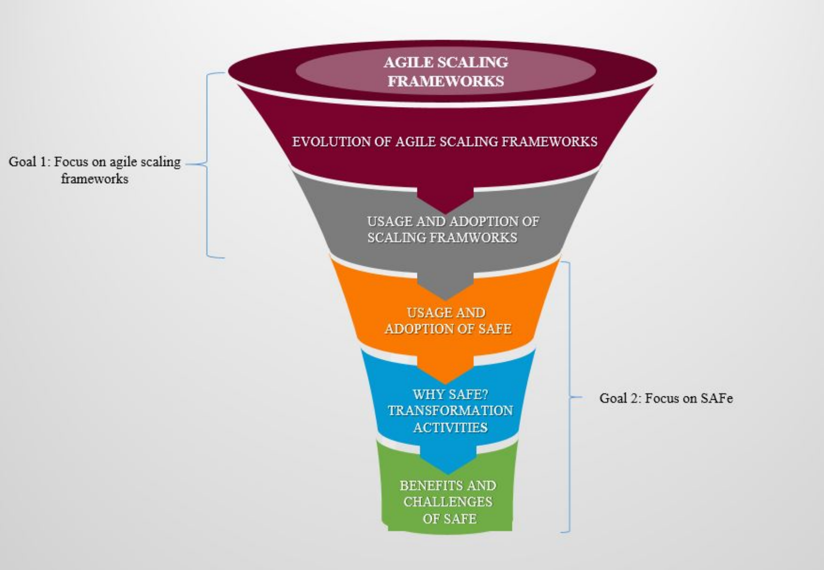
AGILE SCALING FRAMEWORKS
EVOLUTION OF AGILE SCALING FRAMEWORKS
USAGE AND ADOPTION OF SCALING FRAMWORKS
USAGE AND ADOPTION OF SAFE
WHY SAFE? TRANSFORMATION ACTIVITIES
BENEFITS AND CHALLENGES OF SAFE
Goal 1: Focus on agile scaling frameworks
Goal 2: Focus on SAFe

Figure 1.1. Research Goals

SAFe, see Figure 1.1. We choose SAFe due to its popularity across the software industry, with 37% of respondents adopting based on the latest non-scientific survey conducted by Digital.ai (earlier known as VersionOne) between February to April 2021 [One,]. Responses came from a broad range of software industries located all over the globe. The report also stated the following about SAFe popularity in the past five years, *"Over the past five surveys, we have seen the use of SAFe grow significantly to become the dominant approach, in use by more than a third of respondents"*. Another research-based study also indicated the popularity of SAFe over other scaling frameworks like LeSS and DAD [Kuhrmann et al., 2017]. Additionally, there is a growing interest among the academia and industry to understand SAFe usage and adoption [Edison et al., 2021, Beecham et al., 2021].

The research questions follow a logical sequence by first finding the reasons behind designing the scaling frameworks, how they evolved over the years, later why they are being adopted, followed by their benefits and challenges. Next, we focused on SAFe by investigating why it is adopted, the transformation process, and finally, finding the benefits and challenges of its usage in the software industry.

1.3 Structure of Thesis

The thesis is structured as follows. Firstly, we briefly discuss related work in Section 2. Secondly, we present our research problem and research methodology in Section 3. Thirdly, we give an overview of the results from the publications in Section 4. Fourthly, we discuss the implications and threats to validity in Section 5. Finally, in Section 6, we conclude the thesis and propose future work.

2. Related Work

"Research is creating new knowledge."

-Neil Armstrong

In this chapter, we review the previously published literature related to this thesis. Firstly, we start with a short description of agile methods and their importance in large scale. Later, we describe the agile scaling frameworks, which became popular in the recent past. Lastly, we review the research-based concerned with the most popular scaling framework, SAFe.

2.1 Agile Software Development

In following sections we briefly describe the history of agile methods, agile manifesto, and common agile methods used in software development.

2.1.1 History of Agile Methods

Waterfall development was the most commonly used process model in software development in the past. The waterfall model was developed by Winston W. Royce [Royce, 1987], and consists of the following phases: system requirements, software requirements, analysis, program design, coding, testing, and operations. It is a sequential model, where each phase is frozen before the next phase could begin, and every phase has an emphasis on documentation [Balaji and Murugaiyan, 2012]. For instance, it begins with collecting the requirements from the users, customers and end-users [Royce, 1987]. Later, these requirements are well documented and handed over to designers. After that, design documents are given to developers for development. The model looks perfect for constructing a bridge, where the needs of the end-user are relatively static [Cohen et al., 2004]. However, developing software is more complex due to rapid changes in technology and market needs. For instance, software requirements often change, customers may not be sure of defining all requirements at the beginning of the project, and new requirements may emerge in the middle of the project duration [Cohen et al., 2004].

The challenges above with waterfall development led to the development of other models such as V-model [Coad et al., 1999], and Rational Unified Process (RUP) [Booch, 1995]. V-model is an extension of the waterfall model by mapping the verification and validation activities to each phase of the model [Schuppan and Rußwurm, 2000]. Even though these approaches aimed to solve the waterfall problems, they were still heavyweight, documented, and plan-driven approaches [Abbas et al., 2008].

The iterative development model and the spiral model also emerged in response to handle the volatile requirements [Boehm, 1988, Larman and Basili, 2003]. Incremental and iterative development (IID) methods adopt the process behind the waterfall model and repeat them in increments. This model reduces the development time by breaking the whole development cycle into overlapping increments [Larman and Basili, 2003]. In the waterfall model, all the requirements are analyzed at the beginning of the project. In contrast, the requirements in this model are broken down into several increments and are delivered at the end of each increment [Cohen et al., 2004, Larman and Basili, 2003]. IDD approach was more flexible than the waterfall method in handling the change in needs and technologies. However, this model still has a detailed analysis phase and detailed documentation of the requirements before the coding phase.

The spiral model is similar to IID model [Boehm, 1988]. However, they have an additional advantage over IID methods, as they prioritize requirements based on the risk analysis. Spiral and IID models gave greater flexibility in handling the requirements over the waterfall models. However, many practitioners felt that these methods still did not respond to change as rapidly as necessary in the evolving business world [Misra et al., 2012, Cohen et al., 2004]. These methods still had lengthy analysis and extensive documentation involved [Cohen et al., 2004, Misra et al., 2012]. As a result, practitioners and consultants wanted to develop methods that respond to the changes rapidly and are flexible and people-oriented. This quest led to the emergence of the agile movement, and the methods are popularly known as agile methods. Many of the agile methods are based on IID [Larman and Basili, 2003]. The agile methods are a collection of different practices that share common values and principles or characteristics, e.g., communication, iterative development.

These methods became popular after the formulation of Agile Manifesto in 2001. The manifesto was created when Cockburn invited a group of 17 software engineering professionals who were following some lightweight software development models to discuss their common philosophy and pressing issues in software development and termed their shared philosophy as “agile”. [Laanti et al., 2013, Misra et al., 2012]. The manifesto became an essential piece of information during the agile movement, where it describes the core values and principles of the agile methods. The agile manifesto states the following core values [Beck et al., 2001]:

- V1: Individuals and interactions over processes and tools
- V2: Working software over comprehension documentation
- V3: Customer collaboration over contract negotiation
- V4: Responding to change over following plan

The twelve agile principles related to the aforesaid core values are [Beck et al., 2001]: P1. Customer satisfaction, P2. Welcome Change, P3. Frequent deliveries, P4. Work together, P5. Motivated individuals, P6. Face-to-conversation, P7. Working software, P8. Sustainable pace, P9. Technical excellence, P10. Simplicity, P11. Self-organising teams, and P12. Continuous reflection.

Since 2001, several agile methods started to evolve, see Figure 2.1. Figure 2.1, contains the most common agile methods along with their practices. The most popular agile methods are Scrum and extreme programming (XP) [Dikert et al., 2016]. Typically, the methods are applied in small teams with less than ten people as the principles and values are suitable for small teams [Misra et al., 2012, Cohen et al., 2004]. Agile has been both advocated and criticized in the literature. The following are some of the criticisms reflected towards agile methods: agile methods are over-hyped and misunderstood [Irons, 2006], limited support for global distributed settings [Turk et al., 2002], and limited support for large settings [Turk et al., 2002, Mahanti, 2006].

However, there is also empirical evidence that claims that agile has bought in several benefits, e.g., improved productivity, predictability, and team morale [Vijayasarathy and Turk, 2008, Solinski and Petersen, 2016, Shankarmani et al., 2012]. Due to such potential benefits, many organizations showed interest to scale agile to large scale settings and projects [Fuchs and Hess, 2018b, Dikert et al., 2016]. The following section will explain how the above-defined four core values and twelve principles of agile are contradicted in large scale settings.

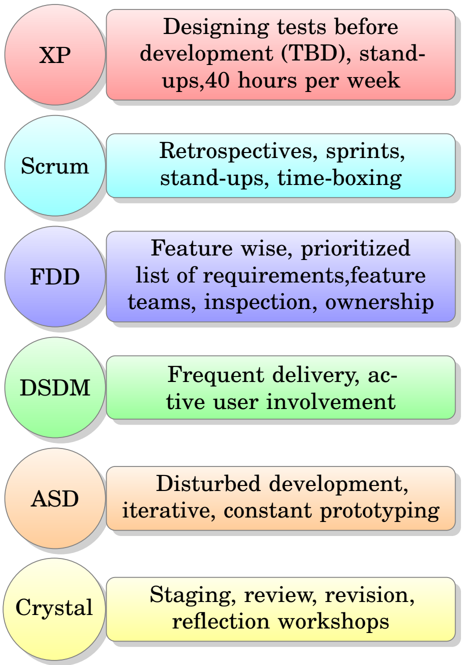
XP
Designing tests before development (TBD), stand-ups,40 hours per week
Scrum
Retrospectives, sprints, stand-ups, time-boxing
FDD
Feature wise, prioritized list of requirements,feature teams, inspection, ownership
DSDM
Frequent delivery, active user involvement
ASD
Disturbed development, iterative, constant prototyping
Crystal
Staging, review, revision, reflection workshops

Figure 2.1. Agile Family [Dybå and Dingsøyr, 2008]

2.2 Agile methods in large scale

In the following section, we present definition of scaling, large in large scale and agile scaling frameworks.

2.2.1 Scaling of agile methods

Adopting agile in large is popularly known as scaling agile [Kalenda et al., 2018]. According to Fuchs et al. [Fuchs and Hess, 2018b] scaling agile to large has different contexts, they are described as follows: (i) agile methods employed by more number of people or teams (e.g., hiring more employees to extend the software development), (ii) expansion of agile methods to other units within the organization (e.g., applying agile to other units such as business, marketing, finance), and (iii) deepening the application of agile methods (e.g., integrating other agile practices from different agile methods). In our thesis we focus on the first two definitions (i.e. (i) and (ii)) of scaling agile.

Now, we shall explain what does this large mean in large scale. In the current literature, there is no exact definition of what does large mean in large scale agile [Rolland, 2016]. The definition of large is dependent mainly on the context and the person defining it [Beecham et al., 2021].

The definition of large typically includes the number of people or agile teams involved in the project or associated costs or duration of a project [Dikert et al., 2016]. For example, Berger and Beynon-Davies [Berger and Beynon-Davies, 2009] categorize a project as a large scale agile project if the project costs exceed ten million GBP. Bjarnason et al. [Bjarnason et al., 2011] consider a project as large scale if it exceeds more than two years.

Other definitions consider the number of teams or people involved in the development. For instance, Dikert et al. [Dikert et al., 2016], defines large, when there are six or more teams or more than 50 people, and Paasivaara et al. [Paasivaara et al., 2008] defines it as 40 people or seven or more agile teams. To provide a more concrete definition for large scale, Dingsøyr et al. proposed a taxonomy, which consists of three categories: (i) small-scale agile projects with one team that use traditional agile practices such as daily stand-ups for intra-team coordination, (ii) large-scale agile projects between two to nine agile teams that use approaches like such as a Scrum-of-Scrums (SoS) for cross-team coordination, and (iii) very large-scale agile projects with at least ten agile teams that require additional coordination mechanisms for cross-team coordination, such as multiple SoS. Based on the above taxonomy, large is defined as least, two agile teams.

Fuchs and Hess extended the definition of large by taking other interpretations described as follows [Fuchs and Hess, 2018b]: a) the use of agile methods in large organizations, b) adoption of agile methods in large projects or large teams, c) application of agile methods in large multi-team settings, and d) adoption of agile practices and principles in the organization as a whole, popularly known as enterprise agile or organizational agility, where other units such as HR, business work in an agile way [Dingsøyr et al., 2014b]. In our thesis, we did not define a specific number for large. However, we described large similar to the Fuchs interpretation of large defined in (a), (b), and (d) above.

2.2.2 Difficulties of scaling agile

Agile methods were originally designed for small and co-located teams [Rolland et al., 2016]. Due to the benefits in the small teams (e.g. improved productivity), several organizations started to adopt agile in large teams and projects. The fundamental principles and practices of agile are challenged when applied in large scale settings. For instance, a large setting consists of multiple teams or large teams, *face-to-face communication* (P6) becomes complex when the size of the team increases [Lindvall et al., 2002]. Also, having multiple teams also reduces the effectiveness of *communication* (V1) [Dingsøyr et al., 2014a]. There is a need for additional coordination mechanisms, e.g., the daily stand-up meeting could extend to include a daily intra-team project meeting to coordinate multiple teams under the project [Reifer et al., 2003]. Large projects also require more *formal documentation* to manage the dependencies and requirements between multiple teams or people, which contradicts with the core values of agile (V2) [Lindvall et al., 2004].

Typically, a large scale project involves a significant number of external experts in addition to team members, which means larger projects are multidisciplinary, thus making it more difficult to share knowledge across involved stakeholders and more coordination efforts to *work together* (P4) [Rejab et al., 2015, Carlile, 2002]. Larger projects also need extensive integration efforts due to a large number of stakeholders, e.g., customers, suppliers [Ambler, 2007]. Large organizations have different departments or units other than IT or development, e.g., business, finance. Such units require additional practices and coordination mechanisms, and also sometimes these units may not be suitable to work with agile ways, e.g. business units and HR needs strict roles [Fuchs and Hess, 2018b]. Large scale projects often involve legacy systems, which are monolithic and hard to integrate into the *short-cycles, iterative workflows* which are supposed to be core principles of agile [Fuchs and Hess, 2018b].

Moreover, adopting agile is more than just applying practices; instead, it is a change in mindset and culture at the organizations [Klünder et al., 2018]. Large organizations typically use traditional approaches for software development, e.g., waterfall, which makes it even more challenging to inculcate an agile mindset [Dikert et al., 2016].

2.3 Agile Scaling Frameworks

We have previously discussed the difficulties to implement agile in large scale settings. To address the scaling challenges, different agile scaling frameworks started to emerge. The primary purpose of these frameworks is to align or support multiple teams to work together in an agile way.

Uludag et al. [Uludağ et al., 2017], identified a total of 22 scaling frameworks based on the structured literature review. The majority of frameworks emerged from very fundamental approaches of agile development, such as XP and Scrum, and they were enhanced in order to apply to the large settings [Uludağ et al., 2017]. There has been an increase in the adoption and usage of these scaling frameworks in the recent past [Edison et al., 2021].

Most popular scaling frameworks based on the latest agile survey by Digital.ai [One,] include Scaled Agile Framework (SAFe), Large Scale Scrum (LeSS), Disciplined Agile Delivery (DAD), Spotify, and Scrum at Scale (S@S). Each scaling framework has a set of characteristics, e.g., roles, practices. The practices of the scaling frameworks could be categorized into the following: (i) practices used at the team level, most of the frameworks use basic scrum for the teams [Theobald et al., 2019], e.g., daily stand-ups, (ii) practices used for scaling, to coordinate multiple teams, e.g., scaled planning, scaled retrospectives, and (iii) practices that are used for teams as well as for scaling, estimation, backlog refinement. There are some standard practices among the scaling frameworks, such as sprint planning, sprints, retrospective, review/demo [Theobald et al., 2019]. Teams of teams formation for coordination between teams are also common among the frameworks but known with different names or titles, such as Agile Release Trains in SAFe and Tribes in Spotify model. Some practices are specific to certain frameworks. For example, product owner sync, an architectural runway is only found in SAFe, architecture envisioning in DAD and managing impediments in S@S [Theobald et al., 2019]. Each framework has its advocates and critics, but to date, there is very little empirical evidence about their efficacy in general [Beecham et al., 2021]. A more detailed description of each framework is presented in the following sections.

2.3.1 Overview of the popular scaling frameworks

We will give an overview of the popular frameworks identified based on the latest agile survey conducted by Digital.ai [One,] in the following sections. The section will briefly describe who developed the framework, essential practices, roles, different configurations of the frameworks. We arranged the frameworks based on the year of their first formal publication.

Large Scale Scrum (LeSS)

Large-Scale Scrum was formally published in 2008 by Craig Larman and Bas Vodde [Larman and Vodde, 2016]. LeSS accommodates the Scrum practices and principles to larger projects [Uludağ et al., 2019], therefore it also called as a multi-team scrum. Multiple scrum teams develop a

Minimum Viable Product (MVP) at end of each sprint. At the beginning of every sprint all the team members discuss about the features and they start with highest prioritized features from the product backlog. Later, the team works on the assigned features [Larman and Vodde, 2016].

Currently, there are two versions of LeSS: “normal” LeSS, for up to eight Scrum teams, and LeSS Huge, for more than eight Scrum teams [Larman and Vodde, 2016]. To manage more than eight teams, LeSS introduced concepts such as requirements areas (RAs), area POs (APOs) and area product backlogs (APBs) [Larman and Vodde, 2016].

Scaled Agile Framework (SAFe)

SAFe was designed as a blueprint for scaling agile to large enterprises. The first version was released in 2011. The framework is frequently updated. At the time of writing, the current version is 5.1 [Inc.,], representing the sixth major revision of the framework. Figure 2.2 gives an overview of SAFe.

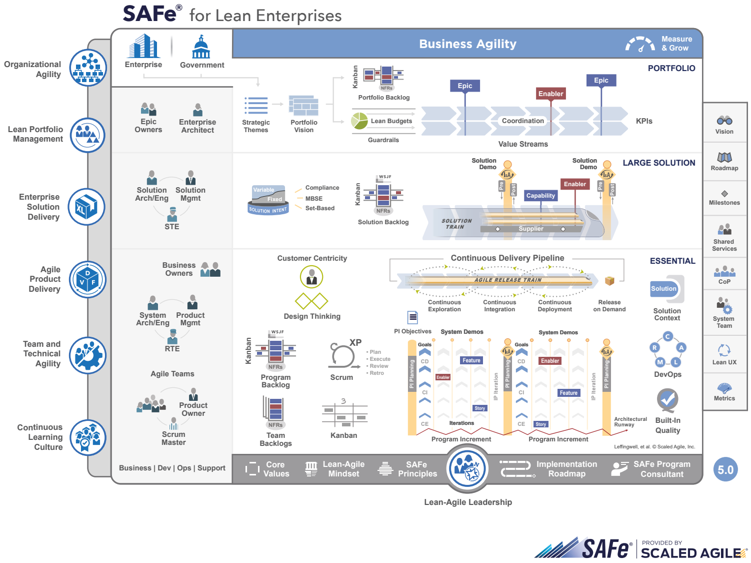

Figure 2.2. The Scaled Agile Framework, Version 5.0 [Scaled Agile Inc., e] (Used with permission from Scaled Agile)

The framework evolved as an approach for developing software and complex systems in an agile and lean manner [Scaled Agile Inc., e]. It incorporates practices from Scrum, Extreme Programming (XP), Kanban, and lean into four different configurations: the *essential SAFe* [Inc.,], *portfolio SAFe* [Inc, a], *large solution SAFe* [Inc, b], and *full SAFe* [Scaled

Agile Inc., e]. Each configuration comprises different roles, processes, and events to achieve a targeted solution. Organizations can adopt one or more SAFe prescribed configuration(s) based on their needs. In the following we would define important roles and events in each of the configurations.

The *essential SAFe configuration* [Inc.,] contains roles, activities, and events to deliver solutions through the central concept of agile release trains (ARTs). An ART is a long-lived organizational entity composed of agile teams [Scaled Agile Inc., a]. An ART typically includes 50 to 125 people, and delivers various solutions incrementally, by using fixed length *program increments* (PIs) as time-boxes. PIs are typically 8 to 12 weeks long [Scaled Agile Inc., a]. Each PI starts with 2-day PI planning session, where all the teams plan features and identify the dependencies between them for the upcoming PI.

The *large solution SAFe configuration* [Inc, b] consists of different roles, artifacts, and processes to build complex solutions. The *solution train*, is the key organisational construct of this configuration which aligns the people and the work towards a common solution vision, backlog, and mission [Inc, b].

The *portfolio SAFe configuration* [Inc, a] comprises of a set of roles, practices, and principles to initiate and govern the development of value streams. A *value stream* is the sequence of steps used to build a solution that generates continuous value for the customer. A value stream may directly deliver customer value, or it may support an internal process [Scaled Agile Inc., b]. The portfolio SAFe configuration is said to contain the necessary processes to build and develop enterprise needs that meet the strategic objectives. This configuration in SAFe is responsible for defining the strategy and investment funding for value streams and their solutions.

The *full SAFe configuration* includes all the three aforementioned configurations [Scaled Agile Inc., e].

SAFe provides a twelve step roadmap to guide organizations on how to successfully implement SAFe. The twelve steps are presented in the Figure 2.3. Many participants in a recent workshop on large-scale agile development (at XP2020 conference) [Bass and Salameh, 2020] agreed that SAFe provides all the necessary steps for its implementation. However, they mentioned that the steps could be difficult to implement. Other participants considered SAFe implementation as complicated, as it includes unnecessary processes that are plan focused, bureaucratic, and

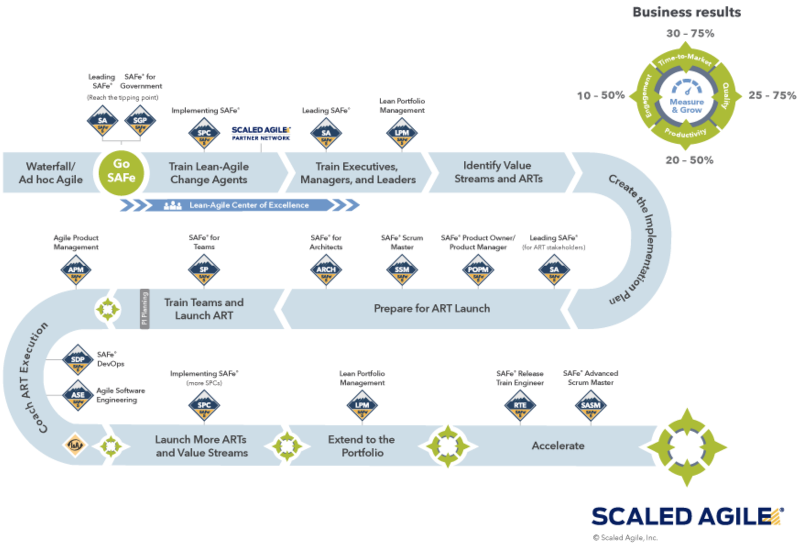
Business results
30 - 75%
Time-to-Market
10 - 50%
Engagement
Measure & Grow
Quality
25 - 75%
Productivity
20 - 50%
Leading SAFe®
SAFe® for Government
(Reach the tipping point)
SA
SGP
Implementing SAFe®
SPC
SCALED AGILE® PARTNER NETWORK
Leading SAFe®
SA
Lean Portfolio Management
LPM
Waterfall/ Ad hoc Agile
Go SAFe
Train Lean-Agile Change Agents
Train Executives, Managers, and Leaders
Identify Value Streams and ARTs
Lean-Agile Center of Excellence
Create the Implementation Plan
Agile Product Management
APM
SAFe® for Teams
SP
SAFe® for Architects
ARCH
SAFe® Scrum Master
SSM
SAFe® Product Owner/ Product Manager
POPM
Leading SAFe® (for ART stakeholders)
SA
PI Planning
Train Teams and Launch ART
Prepare for ART Launch
Coach ART Execution
SDP
SAFe® DevOps
ASE
Agile Software Engineering
Implementing SAFe® (more SPCs)
SPC
Lean Portfolio Management
LPM
SAFe® Release Train Engineer
RTE
SAFe® Advanced Scrum Master
SASM
Launch More ARTs and Value Streams
Extend to the Portfolio
Accelerate
SCALED AGILE®
© Scaled Agile, Inc.

Figure 2.3. SAFe Roadmap of Transformation (Used with permission from Scaled Agile)

dis-empower team autonomy [Bass and Salameh, 2020].

Participants in the workshop reported SAFe as one of most complex frameworks, some also said there is a lot of guidance on how to implement and start SAFe [Bass and Salameh, 2020]. However, very little guidance is provided once implemented in the organizations [Christopher and De Vries, 2020]. This reinforces the need to understand the various activities involved in SAFe transformations and analyze how these activities are mapped with the SAFe implementation roadmap.

The Spotify Model

The Spotify model [Kniberg and Ivarsson, 2012] was first published in 2012 by Henrik Kniberg and Anders Ivarsson. The framework is based on Scrum and Lean practices. The primary goal of Spotify framework is to create autonomous cross-functional teams. The structure consists of Squads, Chapters, Guilds, and Tribes [Kniberg and Ivarsson, 2012].

Teams are called Squads. Squads can use different methods like Scrum, Kanban or a mix of both. Each Squad has a long term mission and develop the product based on the mission [Salameh and Bass, 2019b]. The collection of Squads is called a Tribe. Chapters are a small group of people with similar skills, e.g., testers or developers. Guilds consist of small groups of people who would like to share their knowledge on coding, practices, and tools. Typically, the Chapters are created within the Tribe, while the Guilds are formed across the organization [Kniberg and Ivarsson, 2012].

Disciplined Agile Delivery

DAD is defined as a people-first, learning-oriented hybrid agile approach to IT solution delivery. DAD tool kit provides two categories of roles, (i) Primary roles, which are commonly found in DAD teams regardless of the level of scale, and (ii) Supporting roles, are filled on temporary basis, to address scaling issues [Alliance,].

DAD had defined nine principles based on and lean and flow principles [Petersen and Wohlin, 2011]. DAD projects or products follow a life cycle, which has several critical features: (i) It is a delivery life cycle: The DAD life cycle extends the Scrum construction life cycle, (ii) There are explicit phases: The DAD life cycle is organized into three distinct, named phases, inception, construction and transition that map with the agile coordinate-collaborate-conclude (3C) rhythm, (iii) The delivery life cycle is shown in context: The DAD life cycle recognizes that the process of identifying and selecting projects takes place years or even decades before the project's start date, and (iv) There are explicit milestones: The milestones are an integral part of DAD's governance and risk reduction strategy [Alliance,].

Scrum at Scale (S@S)

The Scrum@Scale framework is an extension of the Scrum framework [Scrum Inc.,]. It was developed by Dr. Jeff Sutherland based on the fundamental principles of Scrum, Complex Adaptive Systems theory, game theory, and object-oriented technology. Its a lightweight framework, that can customized to different kinds of industries. There is an increase in the adoption of this frameworks in past two years [One,].

Nexus

Nexus is a framework for used for developing and sustaining scaled product and software development initiatives [Framework,]. It consists of roles, events, artefacts to knit together three to nine scrum teams. Nexus is an exoskeleton that is build on top of multiple Scrum teams. Roles, events, and artefacts are similar to scrum in general. Some new roles such as Nexus Integration Team and Nexus Integration Team have been added [Framework,].

2.4 Research on Agile Scaling Frameworks

In this section, we will provide an overview of the research conducted on different scaling frameworks. We found four types of research-based studies in the domain of scaling frameworks: (i) primary studies that are based on several scaling frameworks, e.g., case studies conducted in various industry settings using different scaling frameworks [Conboy and Carroll, 2019a], (ii) primary studies on individual frameworks, e.g., case studies or surveys on SAFe [Paasivaara, 2017], (iii) secondary studies on scaling frameworks, e.g., SLRs on comparing different frameworks based on the primary studies, literature review on LeSS and SAFe [Kalenda et al., 2018] (iv) review based studies, e.g., studies that compare practices [Theobald et al., 2019], artifacts [Wińska and Dąbrowski, 2020] between frameworks based on the information present on official websites of the frameworks.

We summarize the first category of studies in the following. We identified two studies in this category. The first study identified the different challenges of using scaling frameworks [Conboy and Carroll, 2019a] and subsequent study illustrated how SAFe and DAD addressed software development risks in global software development [Beecham et al., 2021]. The study reported that SAFe and DAD were able to address most of the risks associated with GSD. Even though the frameworks were successfully able to mitigate the risks of GSD, the process of the framework adoption is challenging [Conboy and Carroll, 2019a]. Some of the critical challenges reported in [Conboy and Carroll, 2019a] are as follows: difficulties to understand the *terms and concepts* defined under the frameworks, choosing between several frameworks without a *comparison model* to contrast and compare between different frameworks was problematic, lack of *readiness* among the developers or team level across the organization, difficulties to balance between the framework structure and the organizational structure. The above two studies show that the frameworks have their advocates and critics. Also, the context of the organization is an essential factor that could influence the efficiency of the frameworks [Beecham et al., 2021]. Both studies raise a need for more in-depth studies on the scaling frameworks to understand their use and effectiveness.

Now, let us move to the second category, i.e., the primary studies published on individual frameworks. There is an increase in literature on studying individual frameworks such as SAFe, LeSS, and Spotify Model.

For instance, Paasivaara et al. [Paasivaara and Lassenius, 2016] and Uludag et al. [Uludağ et al., 2019] reported the challenges and of LeSS adoption. Alsaqaf et al. [Alsaqaf et al., 2020] studied the applicability of LeSS practices in addressing the quality requirement challenges in distributed settings. Studies on Spotify include: [Smite et al., 2019, Salameh and Bass, 2019b, Salameh and Bass, 2019a, Gerster et al., 2020]. For instance, Šmite et al. [Smite et al., 2019], reported the success criteria for Guilds and how they help in cultivating knowledge sharing in large organizations. The existing scientific literature lacks empirical studies on S@S, Nexus, DAD, even though there is an increase in the adoption of these frameworks [Edison et al., 2021]. However, framework official websites have been actively publishing success stories and case studies on framework adoption. Except for Spotify, the majority of the popular frameworks, e.g., SAFe [Scaled Agile Inc., d], DAD, Nexus, have publicly available case studies on their official sites. The case studies have been published to bring the frameworks into the limelight rather than provide insights on adoption and usage. Therefore, the publication bias is extreme across these studies.

Based on the above studies, we understand that individual studies on scaling frameworks were related to challenges and understanding specific practices, only one study reported regarding the transformation process [Paasivaara, 2017]. Many studies also report to do more in-depth studies to understand the usage of the scaling framework in practice, e.g., to transform to SAFe [Paasivaara et al., 2008]. Moreover, the current literature is more problem-centric, where researchers are keen to report the adoption challenges. More in-depth research on how to select a framework, transformation process, and solutions to the existing scaling challenges are needed [Gerster et al., 2020].

The next category of studies is the secondary studies conducted on scaling frameworks. We identified two studies [Edison et al., 2021, Kalenda et al., 2018]. [Edison et al., 2021]. Both the studies were included towards reporting challenges and success factors between several scaling frameworks, e.g., SAFe, LeSS, DAD. For example, Edison et al. [Edison et al., 2021] identified 31 challenges grouped into nine categories, e.g., inter-team coordination and 27 success factors grouped into four categories, e.g., management buy-in. The other study was more specific towards SAFe, and LeSS [Kalenda et al., 2018]. Most of the challenges related to scaling frameworks have also been persistent in large-scale agile [Dikert et al., 2016]. The secondary studies indicate an increase in the number of primary studies in the domain of scaling frameworks. However, the SLR also found that many companies choose to develop their customized method for scaling, even after many frameworks have been published in the market. The reasons can be that organizations are more comfortable scaling what they already know rather than switching to a large-scale unknown frameworks. However, when we compare the number of organizations that use scaling frameworks (32 organizations) versus using customized methods (13 organizations), the scaling frameworks overpower the other category [Edison et al., 2021].

The above study ([Edison et al., 2021]) also mentioned that there is a lack of longitudinal studies on the adoption and application of agile large-scale. Conducting such studies is challenging due to a lack of proper resources, e.g., funding and access to longitudinal data. Despite such challenges, the need for such studies to understand the complex phenomenon of agile adoption on large scale is still in demand [Moe and Dingsøyr, 2017]. We also found that none of the existing literature reviews has included the grey literature from the official website of the scaling frameworks, e.g. SAFe, LeSS. To bridge the gap between practice and research, there is a need to include the grey literature studies [Garousi et al., 2019a].

We summarize the last category of studies on scaling frameworks, i.e. review based studies that compare different frameworks. These review based studies rely both on primary studies and information from the official websites of the frameworks. We identified six studies that compared different frameworks based on different characteristics [Wińska and Dąbrowski, 2020, Diebold et al., 2018, Theobald et al., 2019, Christopher and De Vries, 2020, Alqudah and Razali, 2016, Uludağ et al., 2017], e.g., based on practices [Theobald et al., 2019], artifacts [Wińska and Dąbrowski, 2020]. Theobald et al. [Theobald et al., 2019] compared the practices among twelve agile scaling frameworks. The study concludes that the majority of the frameworks at the team level implement Scrum practices and the common practices for scaling include scaled plannings, meetings and retrospectives, e.g., PI plannings. However, the study calls for more in-depth research on the practices and validation from the framework creators or authors. Winska et al. [Wińska and Dąbrowski, 2020], compares the artifacts among five different scaling frameworks, namely, LeSS, Nexus, S@S, SAFe, and Agile PgM. Based on the analysis, the authors conclude that SAFe and Agile PgM has focused on strictly defined artifacts that are mostly related to planning activities. Additionally, the above methods are duplicating the problem of documentation, which could contradict the core values of agile, e.g., continuous improvement Whereas, the scrum derived frameworks, S@S, Nexus, S@S have more emphasis on continuous process improvement and self-organization concepts. All the above frameworks have emphasized more on processes and working culture than on the actual products. More detailed research on artifacts and their usage in practice is required.

Other studies compared the frameworks based on its benefits [Christopher and De Vries, 2020], challenges [Christopher and De Vries, 2020], flexibility [Christopher and De Vries, 2020, Diebold et al., 2018], inter-team communication practices [Christopher and De Vries, 2020], coaching and support [Uludağ et al., 2017], and complexity [Christopher and De Vries, 2020]. The key results from the studies are as follows: SAFe is rated as highly complex and less flexible [Christopher and De Vries, 2020, Diebold et al., 2018]. LeSS and DAD have medium complexity and flexibility [Christopher and De Vries, 2020, Diebold et al., 2018]. There was no information on Nexus regarding the complexity. Inter-team communication is high in SAFe, LeSS, and DAD and medium in Nexus [Christopher and De Vries, 2020, Diebold et al., 2018]. More information on framework comparison is given in Table 2.1. The above studies aimed to provide comparison criteria to support the decision making on which framework is best suited for the organizations. Even though the studies aimed to provide information to support the decision making, they still miss several components such as context, culture, and practical cases on usage. The studies are still in the preliminary stages and need more research to guide the selection process of scaling frameworks in practice.

Table 2.1. Comparison between different scaling frameworks

Framework/Characteristics	LeSS	SAFe	DAD	Spotify Model	Nexus	Scrum@scale
Publication date [Uludağ et al., 2017]	2008	2011	2012	2012	2012	2015
Category [Uludağ et al., 2017]	Framework	Framework	Framework	Model	Framework	Framework
Underlying agile methods [Diebold et al., 2018]	Scrum	Scrum, Kanban, Lean, specific XP practices	Scrum, Kanban, cultural emphasis	Scrum	Scrum	Scrum
Flexibility [Diebold et al., 2018]	Medium	Low	High	Medium	Medium	Medium
Training and support [Uludağ et al., 2017]	Yes	Yes	No	Yes	Yes	Yes
Certification [Uludağ et al., 2017]	Yes	Yes	No	Yes	Yes	Yes
Case studies on official site [Uludağ et al., 2017]	Yes	Yes	No	Yes	Yes	Yes

2.5 Research on SAFe

In the following sections, we start describing the popularity of SAFe in the software industry followed presenting research-based studies conducted on SAFe related to our research goal (G2).

2.5.1 Popularity of SAFe

Out of all frameworks, SAFe has been the most popular according to a current state of the agile survey, with 37% of respondents adopting it [One,]. The report also stated the following about SAFe popularity in the past five years, *"Over the past five surveys, we have seen the use of SAFe grow significantly to become the dominant approach, in use by more than a third of respondents"*.

The popularity of SAFe and the marketing of the benefits by the framework influenced several companies to take SAFe into use. At least more than 100 companies have already taken SAFe into use according to the official website [Scaled Agile Inc., d], and many companies are considering taking SAFe in future according to a recent survey published in 2020 [Petri Kettunen and Männistö, 2020]. We have also observed organizations who already implemented a particular framework for a while (e.g., Spotify Model) also transitioned to SAFe [Carroll et al.,]. The research-based studies on SAFe have been increasing in the recent past [Edison et al., 2021]. However, there are a limited number of studies that have information on SAFe usage and adoption. The probable reasons could be due to the complex nature of transformation and complexity of the framework, making it challenging to study and comprehend its usage in large-scale settings [Bass and Salameh, 2020]. The grey literature from the official SAFe website still stands as a dominant source of information for SAFe usage, and adoption [Scaled Agile Inc., d].

2.5.2 Reasons for adopting SAFe

The reasons for SAFe adoption were reported in two studies [Paasivaara, 2017, Pries-Heje and Krohn, 2017]. The reasons could be divided into two categories: (i) to improve the current state of the organization and (ii) to solve the existing challenges. The category (i) reasons from the previous study include improving collaboration between management and IT and faster reaction to changes. Category (ii) include: overcome the delays in spreading the features across teams to solve the challenges related to visibility, poor morale of employees, complexity, and under-estimated dependencies.

The reasons found in the literature were similar to the reasons for agile adoption in general. The existing literature lacks framework-specific reasons, e.g., why SAFe was chosen over other frameworks, what attributes of SAFe were considered for the selection process? This calls for more research on understanding the specific or prominent reasons for SAFe adoption.

2.5.3 Activities involved in SAFe transformation

To our knowledge, only two studies provided information related to the SAFe transformation process [Paasivaara, 2017, Pries-Heje and Krohn, 2017]. The adoption of SAFe was a top-down approach in both of the aforementioned cases, i.e., it started from product management to the teams.

The transformation in [Paasivaara, 2017] proceeded by adopting SAFe training's, change agents and external coaches supporting transformation, and PI planning events. In [Pries-Heje and Krohn, 2017] and [Paasivaara, 2017], help from external consulting companies during the transformation was reported. In-depth information on how the SAFe transformation proceeds are not reported in the current literature. Need for detailed studies which provide information on transformation steps would be necessary to guide the practitioners during the SAFe transformation.

2.5.4 Benefits and challenges of adopting SAFe

We identified several studies that reported benefits and challenges of SAFe usage. The studies used different research methods, such as case study [Paasivaara, 2017, Gustavsson, 2020, Gustavsson, 2019d, Gustavsson, 2018, Heikkilä et al., 2015], survey [Laanti and Kettunen, 2019, Salikhov et al., 2020, Gustavsson and Bergkvist, 2019], action research [Pries-Heje and Krohn, 2017], and design science [Turetken et al., 2017].

The most significant benefit reported in the studies is improved transparency, and visibility [Gustavsson and Bergkvist, 2019, Salikhov et al., 2020, Pries-Heje and Krohn, 2017, Laanti and Kettunen, 2019]. Other benefits of SAFe include improved productivity, and employee engagement [Paasivaara, 2017].

The common challenges of SAFe adoption are described as follows. Several studies mentioned challenges for conducting the PI planning sessions [Paasivaara, 2017, Heikkilä et al., 2015, Gustavsson, 2019b, Gustavsson, 2020, Gustavsson, 2019a]. Change resistance was mentioned in [Paasivaara, 2017, Pries-Heje and Krohn, 2017, Gustavsson, 2019d]. Organizations also felt using SAFe as moving away from agile [Turetken et al., 2017, Paasivaara, 2017, Gustavsson, 2019b].

Not all of the above studies explicitly investigated the benefits and challenges of SAFe usage; instead, they reported benefits and challenges as part of studying another topic of SAFe such as coordination and maturity of SAFe practices. Only three studies had an explicit focus on benefits and challenges.

None of the studies had reported how they measured the benefits and what practices of SAFe led to those benefits. Also, the challenges of SAFe look similar to challenges of adopting agile in large scale settings [Dikert et al., 2016]. More in-depth research on the benefits of challenges of SAFe is required to understand its efficiency in large scale environments.

2.6 Research Gap

Even-though the research on agile scaling frameworks is increasing, there is still a grave lack of empirical research on usage and adoption of these frameworks [Conboy and Carroll, 2019a]. For instance, research on why scaling frameworks have been designed, how did they evolve, why do organizations adopt frameworks, and satisfaction with framework usage is not yet investigated in the current literature. Thus, several researchers have highlighted the need for more empirical research on these frameworks [Cram, 2019, Conboy and Carroll, 2019a, Paasivaara and Lassenius, 2016]. Therefore, the first goal of this thesis we aim to explore the above mentioned topics related to agile scaling frameworks.

We observed that SAFe has been the most popular framework of all the existing frameworks [One,]. Despite of its popularity, only a few studies have mentioned about the reasons for adopting SAFe. However, in-depth information on reasons is lacking. The reasons were mentioned only as a part of case organization description. It is necessary to understand why organizations adopt SAFe over other frameworks.

Several academics and practitioners are interested to understand how is SAFe adopted in an organization [Moe et al., 2016, Moe and Dingsøyr, 2017]. This reinforces the need to study the various activities involved in SAFe transformation in practice due to the lack of information from the current literature.

The framework adopters have claimed several benefits of its adoption; in practice, do we have any empirical evidence for these benefits? Only a few studies have reported the benefits of SAFe, and literature on benefits is still dominated by grey literature from the SAFe webpage, making it biased. It is also deemed important to understand if the framework is able to mitigate the obvious challenges of scaling agile and also to identify if it brings in new challenges. Literature fails to provide sufficient information related to SAFe challenges [Conboy and Carroll, 2019b].

Many researchers and practitioners are interested to explore aforesaid topics on SAFe [Moe and Dingsøyr, 2017, Moe et al., 2016, Dingsøyr et al., 2018, Fuchs and Hess, 2018b, Beecham et al., 2021, Paasivaara and Lassenius, 2016]. Therefore, in this thesis we aim to fulfill the above research gaps by investigating on: (i) reasons of SAFe adoption, (ii) transformation process to SAFe, (iii) benefits and (iv) challenges of SAFe adoption.

3. Research Methodology

For a billion problems, there are a zillion solutions.
-A Telugu Proverb

This thesis is an exploratory study of understanding the usage and adoption of agile scaling frameworks with a keen focus on SAFe. Exploratory research is an approach for studying topics that have very little scientific knowledge and studying such topics is also relevant for the uncover the existing gaps in the current research [Stebbins, 2001]. The primary goal of this thesis is to create an understanding of the unexplored topic in the domain of large-scale agile, i.e., usage of the agile scaling frameworks. Many researchers and practitioners are interesting in understanding how scaling frameworks are adopted in practices and also learn from the industries that have already adopted scaling frameworks. Thus, exploratory research seems to be an appropriate approach for the thesis [Patton, 2002]. This research could also be categorized as inductive, as we did not employ any theory or hypothesis before conducting the research.

In this chapter, we firstly, present the research goals and research questions. Next, we describe the research methodology, data collection, data analysis, and validation.

3.1 Research Goals and Research Questions

The high-level research problem of the present thesis aims to address the following: *"Understanding the adoption and usage of agile scaling frameworks across software industry"*.

Scaling frameworks have gained much popularity and attention in the recent past. As described in the previous chapter, the current research on scaling frameworks is scarce and inconclusive. We aimed to address the research gap by investigating the adoption of several scaling frameworks in general and later study a specific framework, e.g., SAFe, to understand in more detail its adoption. Therefore, we divided our research problem, into two different research goals. The first goal has a broader scope, which investigates several scaling frameworks, e.g., SAFe, LeSS, as phenomenon to understand their adoption and usage in practice. While the second goal is focused on only one framework, i.e. SAFe, due to its popularity and relevance in the software industry. The goals are summarized as the following:

- **Goal 1: Understand the evolution, adoption, and usage of agile scaling frameworks.**

  - RQ1: What are the reasons behind designing the agile scaling frameworks?
  - RQ2: How did the agile scaling frameworks evolve over the years?
  - RQ3: Why do organizations adopt agile scaling frameworks?
  - RQ4: What are the benefits and challenges of adopting agile scaling frameworks?
  - RQ5: How satisfied are practitioners after adopting agile scaling frameworks?

  This thesis's first goal focuses on providing empirical data regarding the evolution, adoption, and usage of different scaling frameworks, e.g. SAFe, LeSS, Spotify Model. The goal is twofold: first, to understand the evolution and adoption directly from the methodologists who created the agile scaling frameworks. Understanding evolution would contribute towards the history of agile methods in large scale. Investigating why frameworks were created and what benefits and challenges the methodologists claimed could help get an overview of framework adoption and usage and guide practitioners during adoption.

  The second is to understand the reasons, expected benefits, and satisfaction of the adopting scaling frameworks from industry practitioners. This would guide the practitioners to select a appropriate frameworks based on what other practitioners have reported on aforesaid topic.
- **Goal 2: Understand the adoption and usage of SAFe.**

  - RQ6: Why do organizations adopt SAFe?
  - RQ7: What activities are involved in doing transformation using SAFe?
  - RQ8: What are the benefits of adopting SAFe?
  - RQ9: What are the challenges of adopting SAFe?

The second goal of this thesis focuses on producing empirical data regarding the adoption and usage of SAFe in the software industry. Till

Table 3.1. Mapping of Research Questions and Research Articles. X denotes article directly answers the research questions, whereas (X) denotes the article indirectly answers the research questions.

Research Questions	I	II	III	IV	V	VI
G1: Understanding evolution, adoption and usage of scaling frameworks						
RQ1: What are the key reasons for creating the agile scaling frameworks?	X					
RQ2: How did the agile scaling frameworks evolve over the years?	X					
RQ3. Why do organizations adopt agile scaling frameworks?		X				
RQ4: What are the benefits and challenges of adopting agile scaling frameworks?	X	X				
RQ5: How satisfied are practitioners after adopting agile scaling frameworks?		X				
G2: Understanding adoption and usage of SAFe						
RQ6. Why do organizations adopt SAFe?		(X)			X	
RQ7. What different activities are involved in doing a large-scale agile transformation with SAFe?				X	X	
RQ8. What are the benefits of adopting SAFe?	(X)	(X)	X		X	X
RQ9. What are the challenges of adopting SAFe?	(X)		X	X	X	X

now, 21 primary studies have been published on SAFe [Edison et al., 2021]; however, they do not provide any detailed information on reasons, transformation and benefits and challenges of adoption. This information could help practitioners get a deeper understanding of SAFe and guide organizations during the SAFe transformation process.

The results of the two goals are answered in or more than one research article that are compiled in this thesis. The mapping of the research questions and the research articles is presented in Table 3.1.

3.2 Mixed-methods Approach

In this thesis, we have adopted a mixed-method approach that combines three important methods: literature review, case study, and survey. The mixed-method helps approach helps to achieve a more complete picture of the empirical reality [Creswell, 1999]. We choose mixed methods approach for the following reasons [Creswell, 1999]: 1. Better or in-depth information could be identified from triangulation or converging the results from different methods e.g. benefits of SAFe adoption were investigated using survey, MLR, and case study, this helped to triangulate our findings and gain deeper and wider insights on benefits, for instance, MLR provided a list of benefits, while case study provided what practices led to the benefits and how, and survey helped understand how strongly practitioners

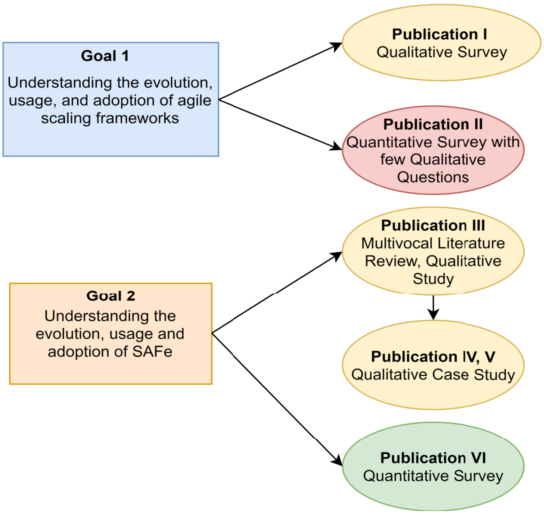
Goal 1
Understanding the evolution, usage, and adoption of agile scaling frameworks
Goal 2
Understanding the evolution, usage and adoption of SAFe
Publication I
Qualitative Survey
Publication II
Quantitative Survey with few Qualitative Questions
Publication III
Multivocal Literature Review, Qualitative Study
Publication IV, V
Qualitative Case Study
Publication VI
Quantitative Survey

Figure 3.1. Research methods and mapping their to the publications

have agreed towards the benefits, 2. Results from one method could be extended by using an another method, for instance, the MLR results for benefits were extended to case study to get more deeper information related to benefits. Having both qualitative and quantitative methods together could help in better understanding of research problem being studied, e.g., survey helped us to get list of benefits and challenges, while case study helped us to get in-depth information of each benefit and challenge 3. We have research questions that could be explored both quantitatively and qualitatively, e.g., benefits were explored using both type of methods.

Our research follows a funnel approach, i.e., we start from a broader goal (like the wider end of the funnel), to more narrower goal (like the narrower end of the tunnel), Figure 3.1. Our first goal has a wider focus; to address the first goal, we conducted two surveys by collecting both qualitative and quantitative data. We narrowed our focus to SAFe due to its popularity and industrial relevance 3.1. To address our second goal, we conducted a literature review, case study, and a survey.

In the next section we would describe the research methods adopted to investigate our two goals, Figure 3.1. First we explain about the methods that were adopted to answer goal 1, followed by goal 2.

3.3 Research Methods for Goal 1

To investigate the goal 1, we adopted survey method. We had two surveys, one survey was qualitative and another survey was quantitative with few qualitative questions. The below section would explain in detail about why the method was selected and how it was carried out in order to answer our research goal 1.

3.3.1 Survey

Survey is defined as the selection of a small sample of people from a large population, which is followed by collection of data from individuals [Fowler, 1995]. Publication I, II, and VI are based on survey method. Publication I and II answer our goal 1, Figure 3.1.

We conducted two surveys (survey I and survey II) in this thesis. Publication II and VI are based on survey I, and publication I is based on survey II. We conducted the both the surveys by following the guidelines from [Linåker et al., 2015]. The survey method allows reaching a broader population and capturing the state-of-art. In the survey I and II, we collected data from the methodologists who created the scaling frameworks and software practitioners adopting scaling frameworks. The survey method was suitable as we needed to reach several methodologists and software practitioners to understand perspectives behind creating the frameworks and usage of scaling the frameworks. Also, it helps in deriving conclusions from a broader population, hence more generalizability, and presenting the current state of the art of scaling framework adoption and evolution. Therefore, the survey method has been chosen.

The description of both the surveys is as follows.

3.3.2 Survey I

Questionnaire Design. The questionnaire consisted of five sections. Only practitioners were asked to respond to our survey. The questionnaire included questions on the organization's transformation background, such as the development methods used before adopting a scaling framework. Questions on reasons, expected and realized benefits, and challenges, which were consolidated based on three previous studies on agile and large-scale agile development [Uludağ et al., 2017, Uludağ et al., 2018, Version One, a]. In each of the three sections, we included an open-ended question to write-in other reasons, benefits, and challenges experienced/witnessed respectively to limit the anchoring effect. In the fifth section, we captured the participants' background information, such as respondent's primary role in the organization, and the location of the organization.

Survey Validation. The questionnaire was initially carefully reviewed by the authors. Next, it was reviewed by domain experts, two survey experts and industrial expert. After incorporating all suggestions, we conducted a pilot survey with three respondents from our target audience and asked for their feedback.

Sampling and Target Audience. We used non-probabilistic convenience sampling. The target audience for our survey included software professionals from various roles, e.g., managers, agile coaches, product owners, and developers, who use agile scaling frameworks in their organizations.

Data Collection. Data collection took place between May and September 2019 using the online tool *"LimeSurvey"*[1]. To reach our target population, we used the following media to promote the survey: (1) international, agile conferences, (2) agile meetup groups, (3) social media groups, and (4) personal networks. We received a total of 204 responses from respondents using different agile scaling frameworks, of which 100 were SAFe adopters.

Data Analysis. We conducted both descriptive and inferential statistics for analysing our data. We started the data analysis by running basic descriptive statistics for contextual information, e.g., reasons, benefits, such as frequencies, to get an overview of the data and insights on how to proceed with inferential statistics. Then, we calculated the mean values for the reasons, expected benefits, and satisfaction.

Next, we conducted the *Kolmogorov-Smirnov test* [Lilliefors, 1967], to the normality of the data. The test showed that our data had a non-normal distribution. Thus, we adopted non-parametric tests to perform inferential statistics. We used the *Kruskal-Wallis H test* [Conover, 1998] to compare the differences between more than two independent groups, e.g., primary framework, industry sector, when the dependent variable is either ordinal or interval/ratio, e.g., reasons, expected benefits, and satisfaction level.

We had two open-ended questions, *other reasons and other expected benefits* and the data from these questions was imported into Excel for qualitative coding. We coded the data by following the guidelines from [Corbin and Strauss, 2008]. We started by open coding, which included breaking down the data into meaningful labels. After that, we grouped the open codes into axial codes based on the similarities and differences.

[1] https://www.limesurvey.org/

3.3.3 Threats to validity

We describe the limitations of our survey method in this section [Wohlin et al., 2012].

Internal Validity. This threat relates to factors that influence the relationship between the research process and the obtained results, e.g., respondent bias. We mitigated the respondent bias by collecting data from only reliable sources, i.e., our responses (more than 90%) came either from people we met during the conferences and agile Meetups or via personal contacts, and we knew that they were using scaling frameworks, which helped us to avoid unreliable responses. As the questionnaire consisted of separate sections investigating the benefits and challenges of adopting agile scaling frameworks, respondents could not overemphasize the positive elements of the SAFe adoption.

External Validity. This threat is concerned with the generalizability of the survey results. Due to our opportunistic sampling strategy, we are unable to identify in what way our sample might be skewed. We had respondents with different roles, using different scaling frameworks, from various domains and geographical locations. Our demographic data are similar to other surveys in the field, giving us some degree of confidence that our sample at least does not completely misrepresent the population. Our respondents are likely biased towards positive answers through their roles, e.g., agile coaches. Further, our respondents likely represent firms and people who are active agile community participants.

Construct Validity. This treat is concerned whether the questions asked in the questionnaire represent the attributes being measured. We formulated the survey statements on the reasons, expected benefits, and satisfaction based on earlier findings in the realm of agile and large-scale agile. We had to limit the questionnaire length, and we could not include all the possible reasons or statements that measure satisfaction that we identified from the literature. We compensated for this through open questions, which we think helped get the most probable reasons and expected benefits for agile scaling framework adoption. We validated the questionnaire with a domain expert, survey experts and tested it by conducting a pilot study, which helped ensure that the questionnaire was clear and understandable to the respondents.

Conclusion Validity. This threat is concerned with the ability to draw the proper conclusion from the collected data. The survey data was mainly

Likert-scale, and we conducted appropriate non-parametric tests to identify differences between independent groups. We also verified our post-hoc analysis with the Mann-Whitney U test that confirmed that we did the proper tests to determine the significant differences for individual frameworks. The qualitative codes were thoroughly discussed among all the authors, which mitigated the misinterpretation of data. We also compared the results with the existing literature for validating our results.

3.3.4 Survey II

Questionnaire Design. The questionnaire consisted of four sections with a total of 22 questions[2]. We included questions on the framework background, e.g., reasons behind the framework creation, claimed benefits and challenges. The questions were compiled based on previous studies [VersionOne, 2020, Uludağ et al., 2017] and the Ask Matrix[3].

Survey Validation. Two experienced researchers validated the questionnaire from the software engineering research group at TU Munich. Their suggestions on length, language, and the order of questions were incorporated.

Data Collection. We collected data between August 2017 and September 2019 using the online tool *Unipark*[4]. We used various approaches to reach out to the inventors or organizations, i.e., methodologists, that created the frameworks: 1. Questionnaire link was sent to 22 methodologists by email. 2. We contacted some of the methodologists in two of the leading agile conferences: XP 2019[5] and Agile 2019[6]. 3. We reached a few methodologists via LinkedIn[7]. We received responses from 15 creators.

Data Analysis. We imported the data into excel sheet. Next, we started with breaking down the data into meaningful entities, i.e., open codes. Later, based on the constant comparison of similarities and differences, we grouped the open codes into higher categories of codes called axial codes. Finally, two authors of the study had a few discussions to compare the open and axial codes from their analysis. The majority of the codes matched between the two authors, and only a few adjustments were made by mutual agreement.

[2] Questionnaire link: https://bit.ly/2ZPl69S.

[3] http://www.agilescaling.org/ask-matrix.html, last accessed on: 03-10-2021.

[4] https://www.unipark.com/en/, last accessed on: 03-10-2021.

[5] https://www.agilealliance.org/xp2019/, last accessed on: 03-10-2021.

[6] https://www.agilealliance.org/agile2019/, last accessed on: 03-10-2021.

[7] https://www.linkedin.com/, last accessed on: 03-10-2021.

3.3.5 Threats to validity

We discuss the limitations of our study through the threats, as suggested by Wohlin et al. [Wohlin et al., 2012].

Construct Validity. This threat is concerned whether the questions presented in the questionnaire represent the attributes being measured. Two survey experts thoroughly checked the questionnaire and evaluated its' understandability, clarity, and readability to counteract this threat. Moreover, the questions were compiled based on previously published studies in the realm of agile software development.

External Validity. This threat is about the generalizability of the results. We aimed to collect responses from all existing scaling frameworks. Out of 22 frameworks, we received responses from 15 methodologists. We could not get responses from the methodologists of seven frameworks despite contacting them several times via email. Thus, this threat could not be completely mitigated. However, we received responses from the most widely adopted scaling frameworks, such as *SAFe*, *LeSS*, *DAD*, and *Spotify* [VersionOne, 2020].

Internal Validity. This threat is concerned with factors that can affect the relationship between the research process and survey results, i.e., the cause and effect relationship. We contacted the methodologists via emails found from the frameworks' official websites. We received confirmation from most methodologists after they filled in the survey, which ensured that the right persons answered the survey. We also met some methodologists during the agile conferences personally and asked them to answer the survey.

Conclusion Validity. This threat deals with the ability to conclude from survey data. The data was coded independently by two researchers. Both researchers compared the codes and drew conclusions together to avoid misinterpretation and misunderstanding of the data.

3.4 Research Methods for Goal 2

This section will briefly describe the research methods that were adopted to answer our goal 2. We adopted three methods to investigate our goal: literature review consisting of both peer-reviewed and grey literature, case study, and survey. The survey method (survey I) is already described in the above section. Other two methods are described below.

3.4.1 Multivocal Literature Review (MLR)

Systematic literature reviews (SLRs) and systematic mapping studies have been popular in the field of software engineering as they help to summarize the existing studies in a specific research domain [Garousi et al., 2019a]. According to the widely adopted systematic literature review guidelines [Keele et al., 2007], a *"fully systematic literature review"* should include both grey and the peer-reviewed literature. Grey literature is defined as, *"[literature] produced on all levels of government, academics, business and industry in print and electronic formats, but which is not controlled by commercial publishers, i.e., where publishing is not the primary activity of the producing body"* [Garousi et al., 2019a].

For many reasons, the majority of SLRs published in software engineering have not included grey literature [Garousi and Mäntylä, 2016]. Including the grey literature is challenging and unfortunately, the search strategy for grey literature has not been systematically addressed in the SLR guidelines [Keele et al., 2007]. This is unfortunate, as excluding this literature eliminates the voice and opinions of the practitioners who do not publish in the academic venues of forums [Garousi and Mäntylä, 2016, Ampatzoglou et al., 2015].

The inclusion of grey literature can, however, be considered a threat as the information reported is based on the opinions and experiences of practitioners rather than systematic collection and analysis of data [Garousi et al., 2016, Garousi et al., 2019b]. Thus, there are issues with grey literature, such as author and publication bias that needs to be accounted for when analyzing such literature. This threat, while severe, does not render the material unusable. Indeed, several systematic literature reviews published in the field of software engineering have included peer-reviewed experience reports (e.g., [Dikert et al., 2016, Rafi et al., 2012], and [Dybå and Dingsøyr, 2008]).

To gather the existing literature on SAFe, we chose to conduct an MLR due to a lack of scientific literature on SAFe usage and adoption. The important reasons for including grey literature are presented in Table 3.2. Such inclusion also helps to bridge the gap between academia and practice [Garousi et al., 2019a].

In our MLR, in the publication III, we searched articles across four different scientific databases: Scopus, Web of Science, ACM, and IEEEXplorer. We used the following keywords: ("safe" AND "scrum") OR "Scaled agile

Table 3.2. Reasons for Including Grey Literature in an MLR [Garousi et al., 2019a]

Attribute	Relevant to our study
Complex intervention	Yes
Complex outcome	Yes
Lack of consensus about the measurement of outcome in peer-reviewed studies	Yes (no in-depth information found)
Low volume of evidence in peer-reviewed literature	Yes (very few primary studies found)
Low quality of evidence in peer-reviewed literature	Yes (no in-depth studies found)
Context important in implementing the intervention	Yes (very few case studies found)

framework". We searched across the official SAFe website as a part of our grey literature search.

The search was conducted in the year 2018 by adopting the following inclusion criteria:

1. Only articles related to the Scaled Agile Framework.
2. Only primary evidence: experience reports, case studies, action research.
3. Publication type: Conference papers, journal papers, workshop papers, white papers from the Scaled Agile Framework's homepage.

Total of 88 documents were included based on the inclusion criteria, of which 82 were gathered from the grey literature and six from the scientific databases.

3.4.2 Data Analysis

Total 88 documents selected were imported into the qualitative coding tool Nvivo. We analyzed the data by following the guidelines from [Corbin and Strauss, 2008]. We started by breaking down the data into meaningful labels, known as open coding. The open codes were grouped into axial codes based on similarities observed between the open codes. For example, for the research question benefits of SAFe adoption, we identified the following open codes, *no missing dates, and schedule slips*, which were coded into the axial code *on time delivery*.

3.4.3 Threats to Validity

The threats to validity [Zhou et al., 2016, Garousi et al., 2019b], and the steps taken to mitigate them are as follows.

Selection Bias. This threat occurs during the selection of primary studies based on interpreting the inclusion and exclusion criteria. It was mitigated by involving all authors in designing the criteria. Two researchers independently filtered the abstracts and titles of peer-reviewed articles.

For the grey literature, each document from the SAFe website was discussed among the authors and included based on inclusion-exclusion criteria.

Subjective Bias. This threat occurs during the qualitative coding of data. Coding was meticulously performed by two of the authors. There were a series of discussions during the coding process among all the authors regarding the naming of the axial codes and the categorization of the open codes into axial codes. The process is traceable.

Publication Bias. Including the grey literature could be seen as a limitation, as they usually describe only positive results [Zhou et al., 2016]. As expected, the majority of the cases considered in this study gave attention to the benefits of the framework. This could have been a serious threat to the results if this study had focused on reporting the framework's benefits, rather than the transformation process.

Grey Literature Challenges. The challenges related to grey literature, presented in [Garousi et al., 2019b], are discussed as follows:

Inherent nature of grey literature materials: There are huge quantities and variabilities of the grey literature materials available on the internet. In our study, we did not use any search engines as the same keywords in the Google search engine did not identify either link to various case studies published on the SAFe website or any primary studies related to SAFe[8]. However, the search results identified the official SAFe website, which acted as our central repository. Having all the content under this helped us deal partially with this threat.

The scaled agile team of experts reviewed the drafts and customised all the case studies to a definitive format. The data from the organizations was collected from a standardised questionnaire, that helped us to deal with variability of information from the white papers. Labeling the addi-

[8] We checked the first ten pages of the search results.

tional supplementary documents case wise helped to manage the quantity. However, the additional supplementary documents had varied information and formats, that took a lot of time to code.

Methodology adopted: Grey literature is based on opinions and experiences, while, the scientific literature adopts systematic data collection. The threat due to this has been partially mitigated, as Scale Agile, Inc. claimed to publish the information by adopting systematic data collection procedures. However, for the blog articles, PowerPoint presentations and white paper from organisations, we could not find anything related to how the information was collected. Thus, this threat could not be mitigated for those reports.

Quality assurance: There are several quality challenges related to the usage of grey literature. Garousi et al. provided a checklist for accessing the quality [Garousi et al., 2019a]. However, in [Garousi et al., 2019b], the same authors mentioned a lack of a well-defined checklist for assessing the quality. Unfortunately, we could not completely mitigate this threat in this study. We verified the credibility of the case studies on the SAFe website by searching for the names of people that were mentioned in case studies, e.g., directors and agile coaches on LinkedIn professional network [In, , Garousi et al., 2019b]. We found similar information was specified on their profile, which was reported in the case studies regarding their role and experience in the organization. However, this cannot completely mitigate this risk related to the quality of the information reported by the case studies.

Limitations of keyword search. For peer-reviewed studies, we have only included the primary sources which explicitly mentioned the keyword “SAFe" or “Scaled Agile Framework". The earlier versions of SAFe, may not have used the term SAFe as the term came into use only in the year 2011. It was difficult to find such studies, as *scaling agile* phrase could result in thousands of search results [Dikert et al., 2016]. It is very difficult to identify if they have used the earlier version unless and until they mention it as an earlier version. To avoid this, we have only included the studies which explicitly mention the keywords SAFe or Scaled Agile Framework.

For the grey literature, we included the earlier variants of SAFe as well, e.g., version 0.98, as the SAFe website only publishes the case studies of the organizations which have adopted their framework.

Information Loss. The codes with less than three organizations were not reported. The keyword search could have missed some studies. We mitigated this by going through the references of all three selected primary studies and did not find any new study that met our inclusion criteria. Some documents had lot of images as well, which were difficult to understand and code that information due lack of detailed description. In such cases some information have been lost while coding the data.

3.4.4 Case Study

The case study is defined as *"an observational request that explores a contemporary wonder inside of its genuine connection; when the limits in the middle of marvel and setting are not unmistakably obvious; and in which numerous sources of proofs are used"* [Yin, 2013]. Case study as a research method helps to investigate a contemporary phenomenon in a specific organizational context [Yin, 2009]. The case study is an appropriate method to investigate the usage and adoption of SAFe in the industry setting, as this phenomenon cannot be replicated in a laboratory setting. SAFe adoption and usage depend on many factors contextual and demographic factors, e.g., domain, number of teams. Therefore, the case study is the right method to investigate this phenomenon.

Publication IV and V are based on a case study conducted in a large financial corporation. Publication IV presents the ART formation and its challenges. Publication V presents reasons, transformation process, benefits and challenges of SAFe usage. This case organization was purposefully selected for the following reasons: 1. The case belongs to the financial sector; this sector has seen the highest number of SAFe adoptions in recent times, this helped to explore the reasons, challenges, transformation process, and benefits of SAFe adoption across the financial domain, 2. The organization reached a certain maturity level with multiple ARTs in progress and incrementally improving and expanding the practices of SAFe, 3. The case organization was one of the largest traditional corporations in Denmark that were taking SAFe, and 4. Availability and accessibility of case organization during the period of our study.

3.4.5 Case Organization Description

Our case organization (PFA) is located in Denmark. PFA develops large and complex insurance and pension products. It is a 1300 person organization with 32 teams involved in software development. The software development is distributed between two sites: *Denmark and Poland*. However, the main development takes place in Denmark. The organization hired some sub-contractors from Poland, as they could not find the right skill-set in Denmark.

At the time of interviews, PFA had four ARTs, namely, DCE[9] *digital and customer experience* (50 persons, 7 teams), DBI train: *data and business intelligence* (40 persons, 5 teams), IP train: *insurance and products* (120 persons, 11 teams) and DM train: *digitalisation and management* (80 people, 9 teams).

3.4.6 Data Collection

In the case study, the main data collection technique was qualitative interviews by following the interview guide approach. We interviewed 24 people from the case organization. We interviewed people from different roles that allowed us to have different perspectives and give us a holistic view of the SAFe usage and adoption. The interviewees were suggested by one of the core members of the transformation team, and some of the interviewees were selected by snowballing approach.

Interviews were audio-recorded and transcribed by a professional transcription company. After that, all the transcripts were exported to the qualitative coding tool Nvivo for analysis.

3.4.7 Data Analysis

The qualitative data was collected from the interviews was analyzed by following the guidelines from [Corbin and Strauss, 2008]. We started by open coding, which included breaking down the data into meaningful labels. After that, we grouped the open codes into axial codes based on the similarities and differences. For instance, in the publication IV, we coded the data related to ART formation and its challenges, and for publication VI, we coded the data related reasons, transformation activities, benefits and challenges of SAFe adoption.

3.4.8 Threats to Validity

The following threats to validity were identified from our case study [Yin, 2009].

Construct validity: This threat is concerned with the how well the case study results align to the reality. We selected interviewees from

[9] Names of the release trains have been slightly modified to retain anonymity

different roles and different levels, e.g., team level and program level. This enabled respondent triangulation. The analysis was carried out in several iterations and several discussions happened before finalizing the coding process. This helped in right interpretation of the data. The interviews were also held in a conversation manner, which enable interviewees to clarify the questions in case of misunderstanding and misinterpretation of the questions. The transcripts from the professional transcription company were also cross checked with the corresponding recordings to ensure the correctness of the data.

External validity: This threat is concerned with ability to generalize the results to other contexts. The results of the present study can be applicable when they applied to an organization having similar contextual information, e.g., pension and insurance systems, having tightly coupled products. However, we compared our findings with other case studies on SAFe and also with SAFe implementation road-map. We found many similarities between our cases and previously published cases, with respect to reasons, transformation process, benefits and challenges. This finding may be applicable for future case studies as well.

Internal validity: The threat is concerned with different factors that could effect the relation between the research process and obtained results. The different factors are: design the interview questionnaire, selection of interviewees. An initial draft of the questionnaire was made by the first author. After that, there were two iterations in which new themes, order of the questions was changed. The answers to these questions are highly dependent on the interviewees. We made of list of potential interviewees during one of the PI plannings and this list was checked by the one of the core member from LACE team. Then some other additional interviewees were suggested. This ensured were collected data from the right people.

Reliability: This threat is concerned with the extent of repetition of the study. The data was collected from multiple sources, ensuring the correctness of the data. The analysis of the coding was validated by conducting a feedback session with a positive response for the results presented.

4. Results

In this chapter, we present an overview of the results of the research questions. A detailed description of the results is given in the respective publications. The chapter is structured in accordance with two research goals formulated based on the existing research gaps.

4.1 Reasons for designing agile scaling frameworks

In publication I, we identified 12 key reasons behind creating the agile scaling frameworks from the 15 methodologists. These reasons were grouped into four categories: complexity, customer, market, and organization. The most commonly stated reasons were: *improving the agility/adaptability of the organization*, *improving the collaboration of agile teams working on same product*, *improving the coordination of agile teams working*, and *improving the synchronization of agile teams working on same product*.

4.2 Evolution of agile scaling frameworks

This question is answered as a part of publication I. Fig.4.1 shows a timeline of the 15 agile scaling frameworks whose methodologists participated in our survey. Grey rectangles (⬜) indicate the *start of development* of a framework, whereas green rectangles (🟩) show *current versions* and blue rectangles (🟦) symbolize *intermediate versions*. Fig. 4.1 also shows two types of dependencies between the frameworks and their versions: Dashed arrows indicate the influence between different frameworks, whereas solid arrows show a predecessor relationship.

Based on the survey answers, we identified that *Crystal* is the first created agile scaling framework which development started in 1997. *Nexus*, *eScrum*, and *S@S* were also relatively early designed compared to most other agile scaling frameworks.

The majority of the frameworks were published between 2011 and 2018. None of the methodologists indicated stopping the further development of their frameworks. Most frameworks have multiple versions, whereas four frameworks have only one version, namely *Nexus*, *LeSS*, *Spotify Model*, and *XSCALE*.

4.3 Reasons for adopting agile scaling frameworks

Results for six reasons from our survey from publication II are presented in Figure 4.2. The survey had respondents using different scaling frameworks. However, 49% of our survey respondents adopted SAFe. The reasons, *to scale more people* and *to remain competitive in the market* received highest mean values.

We found 34 additional reasons (from the open-ended question) from our survey; see Table 4.1. The 34 reasons were categorized into framework-specific, organizational/business reasons, and other reasons. The most commonly mentioned reasons were helps in dealing with pain points of the organization, to inculcate agile mindset, popular and proven successful, and provides flexibility.

We also wanted to find if there were any difference in reasons between different frameworks. We choose the five most common frameworks based on our survey. We calculated the mean values, Table 4.3. Later, we performed Kruskal-wallis H test and found statistically significant differences for the reasons: (i) *because it is widely adopted* (p= 0.000), (ii) *because the framework is well defined and clearly documented* (p= 0.000), (iii) *because it is well supported by coaching, training, and guidance* (p= 0.000). We conducted post-hoc analysis to identify statistically significant differences between individual frameworks for the three reasons mentioned above, see Table 4.2.

Based on the inferential statistics, we understood that SAFe is adopted for its (i) popularity, (ii) because it is well supported by coaching, training, and guidance, and (iii) well defined and documented when compared to other frameworks, such as internally created methods, LeSS, and Spotify.

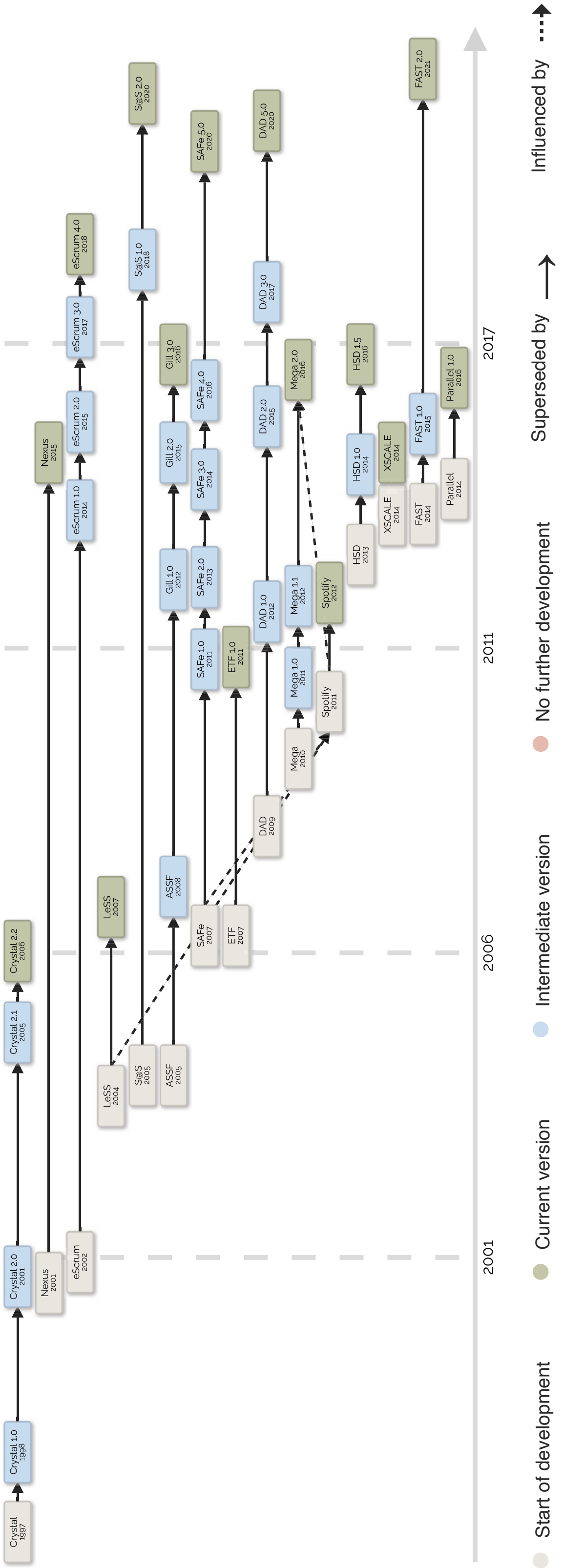

Figure 4.1. Evolution of agile scaling frameworks

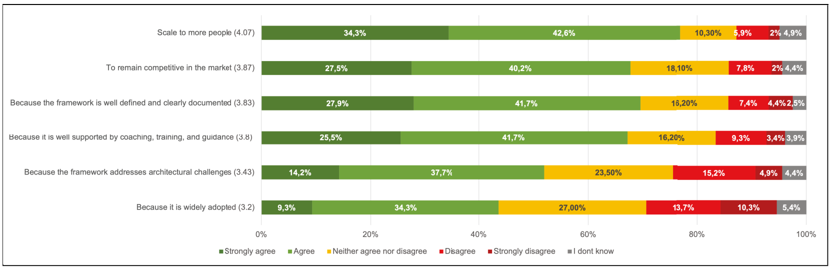

Figure 4.2. Adoption reasons arranged by mean values

Table 4.1. Write-in Adoption Reasons

Framework Specific Reasons	**Framework(s)**	**n**
Popular and proven successful	SAFe	5
Provides flexibility	SAFe,LeSS,S@S	5
Helps in de-scaling the organization	LeSS	4
Provides a light-weight process model	LeSS,Nexus	3
Provides a common language/structure	SAFe	2
Maps well for traditional organizations, e.g., provides an easier transition for managers	SAFe	2
Helps in mixing practices from other methods, e.g., Scrum/DevOps	SAFe,SoS	2
Has a focus on lean portfolio management and system thinking	SAFe	1
Well documented approaches	Nexus	1
Good materials and people with experience to guide	SAFe	1
Organizational/Business Reasons	**Framework(s)**	**n**
Helps in dealing with the organization's pain points/needs/current challenges	SAFe,LeSS,Nexus,SoS,DAD,Internal methods	8
To inculcate an agile mindset	SAFe,Spotify	5
To address growth and complexity	SAFe, Spotify, DAD, Internal methods	4
To have product focus	LeSS,CAF (Continuous Agile Framework)	3
To improve or provide customer value	SAFe, LeSS, CAF	3
To scale agile to more teams or other units	SAFe,LeSS,Spotify	3
To improve employee satisfaction/morale/empowerment	SAFe,LeSS,S@S	3
To improve productivity	SAFe	2
To address the needs of regulatory environments/compliance	SAFe	2
Adopted to support client transition	LeSS	2
Improve agile process and business agility	DAD	2
Increase profits/market share	LeSS,Nexus	2
To be more effective in managing work and teams	SoS	2
To have cross-functional/feature teams	CAF	1
To improve alignment	SAFe	1
Improve performance of the organization	SAFe	1
Foster innovation	LeSS	1
To have secured delivery	SAFe	1
To support new digital ways of working	SAFe	1
To become more flexible	LeSS	1
To generate value faster	SAFe	1
To increase attractiveness as an employer	SAFe	1
Other reasons	**Framework**	**n**
Consultants recommended the framework	Spotify	1
Association with founders of the framework	Nexus	1

Table 4.2. Post-hoc Analysis for Adoption Reasons

Reason for adoption	**Mean Ranks**	**Statistically Significant Pairs**

Because it is widely adopted	**SAFe=88.81**
	LeSS = 47.38
	Spotify=71.10
	S@S= 62.40
	Internal= 32.70
		Internal-Spotify (p=0.039)
		Internal-SAFe (p=0)
		LeSS-SAFe (p=0)

Because the framework is well defined and clearly documented	**SAFe = 90.71**
	LeSS = 71.93
	Spotify = 26.00
	S@S = 79.45
	Internal = 22.30
		Internal-LeSS (p=0.001)
		Internal-S@S (p=0.003)
		Internal-SAFe (p=0)
		Spotify-LeSS (p=0.003)
		Spotify-S@S (p=0.005)
		Spotify-SAFe (p=0)
		LeSS-SAFe (p=0.039)

Because it is well supported by
coaching, training, and guidance
	SAFe = 86.46
	LeSS = 67.77
	Spotify = 38.60
	S@S = 106.39
	Internal = 30.70
		Internal-LeSS (p=0.017)
		Internal-SAFe (p=0)
		Internal-S@S (p=0)
		Spotify-LeSS (p=0.06)
		Spotify-SAFe (0.001)
		Spotify-S@S (p=0)
		LeSS-SAFe (p=0.038)
		LeSS-S@S (p=0.016)

Table 4.3. Mean values for Adoption Reasons

Reason	SAFe	LeSS	Internal	S@S	Spotify
Widely adopted	3.69	2.5	2	2.9	3.1
Well defined and documented	4.26	3.8	2.27	3.9	2.3
Addresses architectural challenges	3.52	3.3	3.27	3.9	3
Well supported	4.12	3.7	2.55	4.6	2.7
To remain competitive	3.91	3.9	3.45	4.4	3.4
To scale to more people	4.12	4	4.18	4.5	3.7

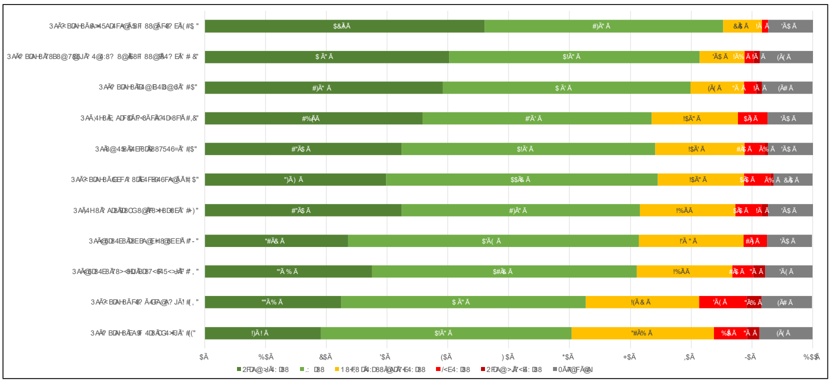

Figure 4.3. Expected benefits arranged by mean values

Table 4.4. Write-in Expected Benefits

Expected benefits	Framework(s)	n
To improve employee engagement/satisfaction	SAFe,LeSS	3
To address most common scaling challenges	LeSS	1
To be customer-centric and product focused	LeSS	1
To foster innovation	SAFe	1
To improve productivity	SAFe	1
To manage multiple requirements	SAFe	1
On-time delivery	CAF	1
To have scalable products	SAFe	1
To have technical excellence	LeSS	1
To dissolve silos	Internal	1
Better understanding of Cost of Delay, queuing theory	SAFe	1
To reduce cost	LeSS	1

4.4 Benefits and challenges of adopting agile scaling frameworks

The benefits in this research question are divided into two parts: expected benefits and claimed benefits. The first part is answered in publication II. The results are based on the survey, where respondents used different scaling frameworks. The results from the survey are presented in Figure 4.3. The expected benefits with top mean values are: to improve collaboration between teams, improve dependency management between teams, and to improve transparency.

To catch expected benefits not covered by our list, we had an open question, which identified twelve additional items, presented in Table 4.11. These largely matched with the reasons discussed in the earlier RQ.

We also wanted to find if there were any difference in reasons between different frameworks. We choose the five most common frameworks based on our survey. We calculated the mean values, Table 4.5. Later, we performed Kruskal-wallis H test and found statistically significant differences for the expected benefits: (i) *to increase delivery predictability* (p= 0.038), (ii) *to increase responsiveness* (p= 0.032), (iii) *to improve software quality*

Table 4.5. Mean values for expected benefits per framework

Expected benefit	SAFe	LeSS	Internal	S@S	Spotify
To improve dependency management between teams	4.09	3.36	3.92	3.9	4.1
To improve collaboration between teams	4.14	4.36	4.08	3.7	4
To improve customer satisfaction	3.82	4.04	3.83	3.7	3.8
To improve team autonomy	3.34	4.18	3.25	3.7	4.2
To have more frequent deliveries	3.89	4.14	3	3.5	3.5
To have shorter time to market	3.9	4.25	3.5	3.6	3.5
To enable faster feedback	3.82	4.21	3.08	3.6	3.3
To increase responsiveness	3.68	4.21	3.17	3.6	3.5
To increase delivery predictability	3.76	3.86	3.83	3.6	2.3
To improve software quality	3.45	4	3.17	2.4	2.6
To improve transparency	3.84	4.46	3.33	3.7	3.5

(p= 0.033), (iv) *to improve team autonomy* (p= 0.006), and (v) *to enable faster feedback* (p= 0.027). Next we conducted a post-hoc analysis to find the individual differences between frameworks, see Table 4.6.

Based on the above statistical tests, we understood that improving team autonomy was the most significant expected benefit among our Spotify respondents when compared to SAFe and internally created methods. Also, enabling faster feedback, increasing responsiveness, and improving software quality were significant expected benefits for LeSS respondents over the respondents of internally created methods and Spotify.

The second part of the benefits is answered in publication I. We identified claimed benefits of adopting scaling frameworks based on our survey. These benefits were grouped into two categories, namely: business/product and organization/culture. The most commonly mentioned benefits were: *enabling frequent product deliveries*, *enhancing employee satisfaction / motivation / engagement*, *improving software quality*, *providing customer / business value*, *improving the collaboration of agile teams working on same product*, *improving the coordination of agile teams working on same product*, *improving the synchronization of agile teams working on same product*.

We identified 22 claimed challenges of adopting scaling frameworks based on our survey in publication I. These challenges were grouped into three categories: implementation, organization/culture, and scope. The most commonly mentioned challenges were: *using frameworks as cooking recipes* and *using frameworks without understanding for what reasons they should be applied*.

Table 4.6. Post-hoc Analysis for Expected Benefits

Expected benefits	Mean Ranks	Statistically Significant Pairs
To improve team autonomy	SAFe = 72.52	Internal-LeSS (p=0.017)
	LeSS = 99.23	Internal-Spotify (p=0.033)
	Spotify=102.85	**SAFe-LeSS (p=0.005)**
	S@S = 98.15	SAFe-Spotify (p=0.038)
	Internal = 61.77	
To enable faster feedback	SAFe = 80.87	Internal-SAFe (p=0.022)
	LeSS = 94.21	**Internal-LeSS (p=0.004)**
	Spotify = 59.25	Spotify-LeSS (p=0.030)
	S@S = 86.30	
	Internal = 49.05	
To increase responsiveness	SAFe = 77.86	Internal-S@S (p=0.042)
	LeSS = 96.71	**Internal-LeSS (p=0.003)**
	Spotify = 67.65	SAFe-LeSS (p=0.041)
	S@S = 89.95	
	Internal = 51.68	
To increase delivery predictability	SAFe = 83.45	Spotify-LeSS (p=0.011)
	LeSS = 79.11	Spotify-Internal (p=0.024)
	Spotify = 38.25	**Spotify-SAFe (p=0.002)**
	S@S = 88.50	Spotify-S@S (p=0.010)
	Internal = 81.14	
To improve software quality	SAFe = 81.56	**Spotify-LeSS (p=0.011)**
	LeSS = 96.39	S@S-LeSS (p=0.029)
	Spotify = 55.05	Internal-LeSS (p=0.040)
	S@S = 60.90	
	Internal = 64.18	

4.5 Satisfaction of practitioners after adopting agile scaling frameworks

This question is answered in publication II. We measured satisfaction level based on four different statements in our survey, Figure 4.4: (i) meeting the expectations of the organization, (ii) willingness to move back to old ways of working, (iii) willingness to shift to another framework, and (iv) willingness to recommend the framework to other similar organizations. We have reverse coded the statements (ii) and (iii).

By analyzing the results for these four statements, we can conclude that majority of our respondents are satisfied with using their chosen scaling frameworks in their respective organizations.

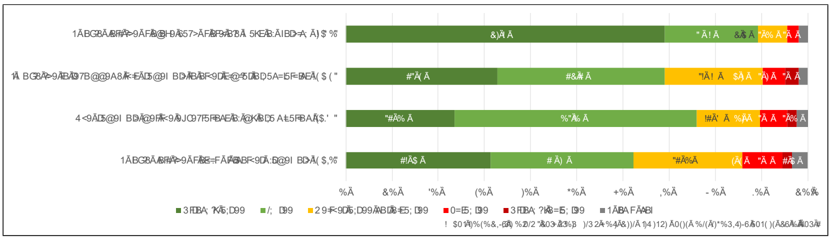

Figure 4.4. Framework satisfaction arranged by means

Later, we started by calculating mean values for all the four statements, see Table 4.7. We observed mean values differed between different frameworks for statements that measured satisfaction level. Next, we conducted the *Kruskal-Wallis H test* and identified statistically significant differences for the following expectations: (i) *the framework met the expectations of my organization* (p= 0.032), (ii) *I would like to shift to another framework* (p= 0.020), and (iii) *I would like to recommend this framework to other similar organizations* (p= 0.000). Next, we did a post-hoc analysis to find out statistically significant differences between the individual frameworks. The results of the post-hoc analysis for these three statements are presented in Table 4.8.

Based on the above analysis, it is understood that S@S had higher satisfaction when compared to other frameworks, such as SAFe and internally created methods. Also, SAFe respondents have more likeliness to shift towards other frameworks when compared to LeSS and S@S respondents.

Table 4.7. Satisfaction level means per framework

Satisfaction Level of the Practitioners	SAFe	LeSS	Internal	S@S	Spotify
The framework met the expectations of my organization	3.81	4.12	3.83	4.6	4
I would not* like to move back to the old ways of working	4.57	4.67	4.6	4.7	3.4
I would not* like to shift to another framework	3.63	4.17	3.4	4.5	3.5
I would like to recommend this framework to other similar organizations	3.95	4.21	3.36	4.5	2.6

*) For readability reasons, both statements have been reverse coded and negatively worded by a "not".

Table 4.8. Post-hoc Analysis for Framework Satisfaction

Satisfaction aspect	**Mean Ranks**	**Statistically Significant Pairs**

4=The framework met the expectations of my organization	SAFe =73.69
	LeSS = 86.37
	Spotify = 76.94
	S@S = 113.15
	Internal = 65.95
		SAFe-S@S (p=0.004)
		Internal-S@S (p=0.008)

5=I would like to shift to another framework[1]	SAFe = 73.09
	LeSS = 98.04
	Spotify = 76.70
	S@S = 108.75
	Internal = 73.77
		S@S-SAFe (p=0.015)
		LeSS-SAFe (p=0.008)

5=I would like to recommend this framework to other organizations	SAFe = 80.51
	LeSS = 91.61
	Spotify = 32.35
	S@S = 103.50
	Internal = 50.90
		Spotify-SAFe (p= 0.001)
		Spotify-LeSS (p=0)
		Spotify-S@S (p=0)
		Internal-SAFe (p=0.039)
		Internal-LeSS (p=0.011)
		Internal-S@S (p=0.007)

4.6 Reasons for adopting SAFe

The reasons for SAFe adoption have been presented in publications V and II. Publication V is a case study that explicitly answers the reasons for SAFe adoption, and publication II is a survey that has investigated reasons for the adoption of other scaling frameworks along with SAFe.

The reasons for SAFe adoption based on our case study can be broadly divided into three categories: (i) to solve the exiting challenges in the organization, (iii) to improve the current state of the organization, (iii) merits of the framework. The reasons from the case study are summarized in Table 4.9. The reasons are listed from the most number of mentions to the least.

The most common reasons found from our case study are: to have faster time to market, to improve collaboration, and well described and comprehensive framework. Some unique reasons, such as financial affordability, to achieve prioritization, and to become a market influencer were also found from our case.

Table 4.9. Reasons for Transformation from Publication V

Reasons	Description	No. of interviewees
Business Reasons		
To have Faster time to market	To overcome long delivery cycles and to enable faster deliveries	9
To become a market influencer	To get on the edge	4
To be flexible to the changing needs	To react faster to changes	2
Financial affordability	Has lots of money to afford the transformation	1
Organizational Reasons		
To improve collaboration	To get rid of the silos, improve collaboration between business and IT	5
To improve productivity	Struggles with low productivity, goals of 2020 strategy	4
To improve efficiency	Goals to improve efficiency by 40 percent	4
To become more customer-driven	To become closer to customer, deliver value to customer	4

To improve transparency	Struggles with transparency, create transparency	3
Improve quality	Goals to improve quality	3
To have continuous delivery pipeline	To enable continuous delivery	3
To improve predictability	Having predictable deliveries	2
To achieve prioritization	Requirements prioritization, framework has some good aspects on this topic	2
To enable cross functional teams	To get rid of silos to work together	2
Financial affordability	financially stable to afford the trainings and certifications	1
Merits of the framework		
Well-described and Comprehensive Framework	Presentable website, coaching support and well-documented	5
Popular framework for scaling	Popularity across financial organizations	3
To get buy-in	Easy to get management buy-in	2
No Exact Reason		
No exact reasons	No clue about the reasons	5

We found 20 reasons for SAFe adoption (from the open-ended question) from our survey; see Table 4.1. The 20 reasons were categorized into framework-specific and organizational/business reasons. We found similar reasons from our survey and case study.

4.7 Activities involved in doing a SAFe transformation

Publications IV and V report the activities identified from a case organization while doing the SAFe transformation. Publication IV has more in-depth information on how ARTs were formed, whereas publication V has an overview of all the transformation activities.

The list of activities in chronological order from the case study are presented in Table 4.10. The timeline of transformation is presented in Figure 4.5. The SAFe transformation, in our case, was top-down and step-wise.

The change started by communicating the reasons, which was followed by forming a pilot ART. The pilot ART was formed with the teams that already had experience with agile ways of working.

Also, these teams were working with front-end development, which was seemed to be easy to show results and business value that could help get in buy-in for forming other ARTs. The success of the pilot ART led to the formation of three new ARTs. The second ART was formed by conducting a design workshop with the key stakeholders. The coaches facilitated the workshops. The last two ARTs were formed in the Lego workshops were different colour Legos were used for different roles and every manager had a certain number of Lego blocks representing a role.

The organization conducted various SAFe training's, e.g., Leading SAFe, SAFe SPCs prescribed by SAFe. Later on, Scrum tours were given to the ARTs before the formal SAFe trainings. As SAFe training's were very specific towards the implementation of framework and practices and did not focus much on understanding the principles and practices of agile and lean.

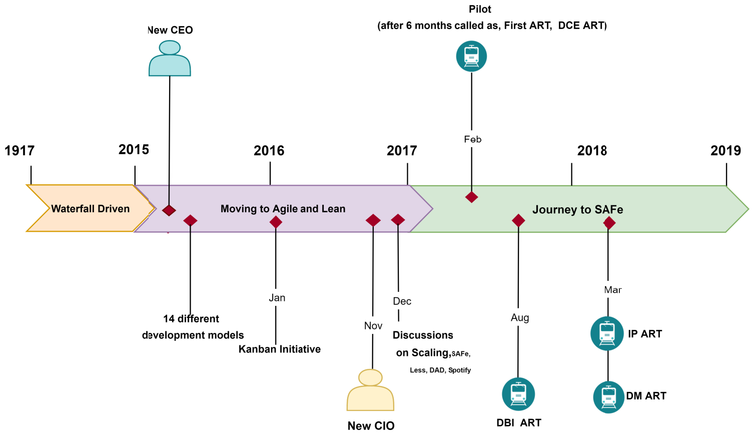

Figure 4.5. Timeline of transformation journey

Table 4.10. Transformation Process from Publication V

Transformation Process	**Description**
Idea of SAFe	Discussions on several frameworks, new CIO
Creating awareness	Steering groups, discussions with directors
Communicating about SAFe	Message of transition in a news letter, whole organization was informed
Piloting	Started with small and teams with agile experience
Internal and External Coaches	Combination of external and internal coaches, handpicked consultants
Education	More than 300 people were trained in different training's, leading SAFe, SAFe for training's, SAFe practitioner, SAFe SPC's, and role training's
Transformation board and Roadmap	Consists of strategies
Prioritization councils and software centers	Four prioritization councils and two software centers were formed
First PI planning	Flying people from Poland, coaches helping the teams, widening the competencies
Creating CoP culture	Encouraging people to join CoPs and giving away power of decision making
Discussion on projects	How to put projects into SAFe world, how to budget and control money

Transformation Process	Description
Transformation teams	To help in forming and designing the ARTs
Forming of other ARTs	Formed three new ARTs
Scrum tours	Five different modules were designed for introducing agile and lean

4.8 Benefits of adopting SAFe

The research question has three parts, (i) expected benefits, (ii) claimed benefits, and (iii) realized benefits of SAFe adoption.

We found six expected benefits of SAFe from the open-ended question in publication II, see Table 4.11. The most common expected among our SAFe respondents was to improve employee engagement.

Table 4.11. Write-in Expected Benefits

Expected benefits
To improve employee engagement/satisfaction
To foster innovation
To improve productivity
To manage multiple requirements
To have scalable products
Better understanding of Cost of Delay, queuing theory

The second part of the research question, i.e., claimed benefits, are presented in Table 4.12. These are presented as a part of publication I, which were reported by SAFe methodologists (from Scaled Agile who developed SAFe) via the survey questionnaire.

Table 4.12. Claimed benefits of adopting SAFe from publication I

Claimed benefits of SAFe
Business / Product
Enabling frequent product deliveries
Improving software quality
Enabling faster time-to-market
Organization / Cultural
Enhancing employee satisfaction/motivation/engagement

The third part of the research question is the realized benefits of SAFe adoption, which is answered in the following publications: I, III, and IV.

Publication III is based on MLR. A total of 23 benefits were identified from the existing primary studies, which were categorized into five themes: organizational, business, transparency, alignment, and built-in quality. The most popular benefits from our MLR were: transparency, alignment, productivity, predictability, and time to market.

Publications V is based on a case study. The realized benefits from the case organization are presented in Table 4.13. The list is ordered based on the most number of mentions to the least. The most mentioned benefits were improved collaboration, transparency and visibility, faster time to market, and clear prioritization.

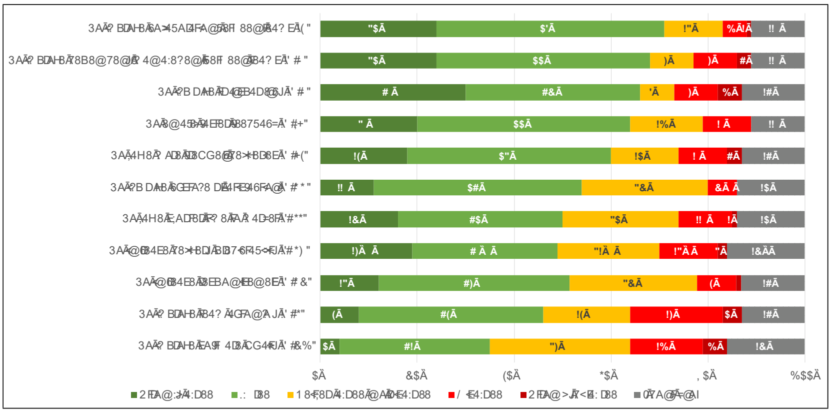

Figure 4.6. Agreement of the Realized Benefits of SAFe, arranged according to the Mean Values from publication IV

Publication VI is based on a survey. Responses to the nine benefits from 100 respondents using SAFe as their primary framework are presented in Figure 4.7. The highest mean value was found for improved collaboration between teams, followed by improved dependency management, and improved transparency. The least mean value was found for software quality.

We found that the benefits identified from the three research methods in the Publications I, II, III, V and VI were similar to each other. We also found that expected benefits, claimed benefits, and realized benefits were similar to each other.

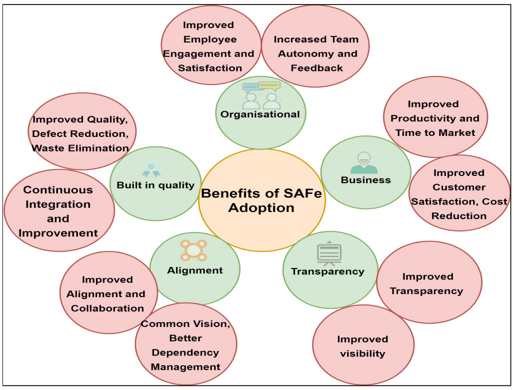
Improved Employee Engagement and Satisfaction
Increased Team Autonomy and Feedback
Organisational
Improved Quality, Defect Reduction, Waste Elimination
Improved Productivity and Time to Market
Built in quality
Benefits of SAFe Adoption
Business
Improved Customer Satisfaction, Cost Reduction
Continuous Integration and Improvement
Alignment
Transparency
Improved Transparency
Improved Alignment and Collaboration
Common Vision, Better Dependency Management
Improved visibility

Figure 4.7. Realized benefits of SAFe from Publication I

Table 4.13. Realized benefits of SAFe adoption from publication V

Benefits of SAFe adoption	Description	No. of interviewees
Business benefits		
Faster time to market	Reduction in the number of handovers, cross functional teams, having right priorities have helped in having faster deliveries	6
Clear prioritization	Sitting together and prioritizing, deciding what is most important for organization, prioritization councils	6
Increase in predictability	Delivered 70 percent of what was promised	3
Organizational benefits		
Improved collaboration	PI planning bought people together, people from different levels sit together	12
Transparency and visibility	Everybody knows what everybody is doing, prioritization became transparent	7
Delivering more value	Focus on delivering right things helped in delivering more value	5
Improved team autonomy	People worked by themselves, self governed and independent decision making	4
Increased employee satisfaction	Employee satisfaction surveys, score: 9 out of 10	4
Scope of innovation	enabled room for innovation	3
Improved communication	Frequent gatherings, introverts started to present their work	3

Improved productivity	Scaled the output, deliver more output, increase in the velocity	2
Better dependency management	Identifying dependencies between teams, shorter path to dependencies	2
Waste removal	Focused on what is important, decrease in misuse and stalling of resources	2

Table 4.14. Claimed challenges of adopting SAFe

Claimed challenges of SAFe
Implementation related challenges
Implementation overhead
Organization / Culture
Using frameworks as cooking recipes
Lack of management buy-in
Moving away from agile

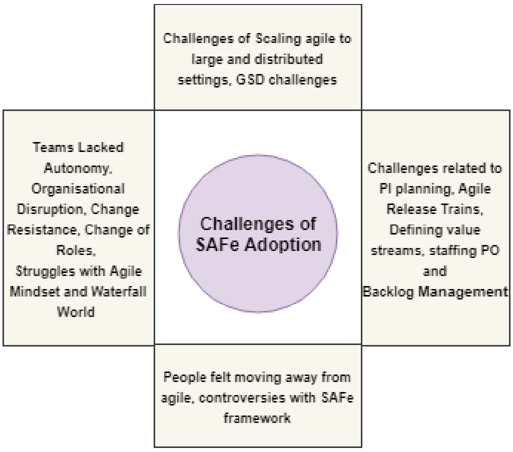
Challenges of Scaling agile to large and distributed settings, GSD challenges
Teams Lacked Autonomy, Organisational Disruption, Change Resistance, Change of Roles, Struggles with Agile Mindset and Waterfall World
Challenges of SAFe Adoption
Challenges related to PI planning, Agile Release Trains, Defining value streams, staffing PO and Backlog Management
People felt moving away from agile, controversies with SAFe framework

Figure 4.8. Challenges of SAFe from Publication I

4.9 Challenges of adopting SAFe

The research question has two parts: (i) claimed challenges and (ii) realized or witnessed challenges of SAFe adoption.

The first part of the research question is answered in publication I. The list of claimed challenges is presented in Table 4.14. A total of five claimed challenges were reported by the methodologists of SAFe, which were divided into two main categories: implementation-related challenges and organizational/cultural challenges.

The second part of the research question is answered publications III, IV, V, and VI. Fifteen challenges related to SAFe adoption were identified from the literature (from the publication III), see Figure 4.8. The challenges were divided into different categories: organizational/business, roles, practices and scaling and distribution. The most mentioned challenges from our MLR were: change resistance, challenges with the first program increment planning, and moving away from agile.

We conducted a case study in publications IV and V. We identified 16 challenges, which were divided into four different categories: organizational, human factor related, challenges with practices and ceremonies, and challenges from SAFe framework. The 16 challenges are presented in Figure 4.9. The most critical challenges were: struggles with projects, struggles by project managers, and forming of ARTs not aligned to value streams (marked in red in Figure 4.9), and pictorially presented in 4.10.

Figure 4.11, presents the results of 16 challenges from 100 respondents using SAFe from publication VI. The highest mean value was found for organizational politics, followed by difficulties in establishing agile mindset

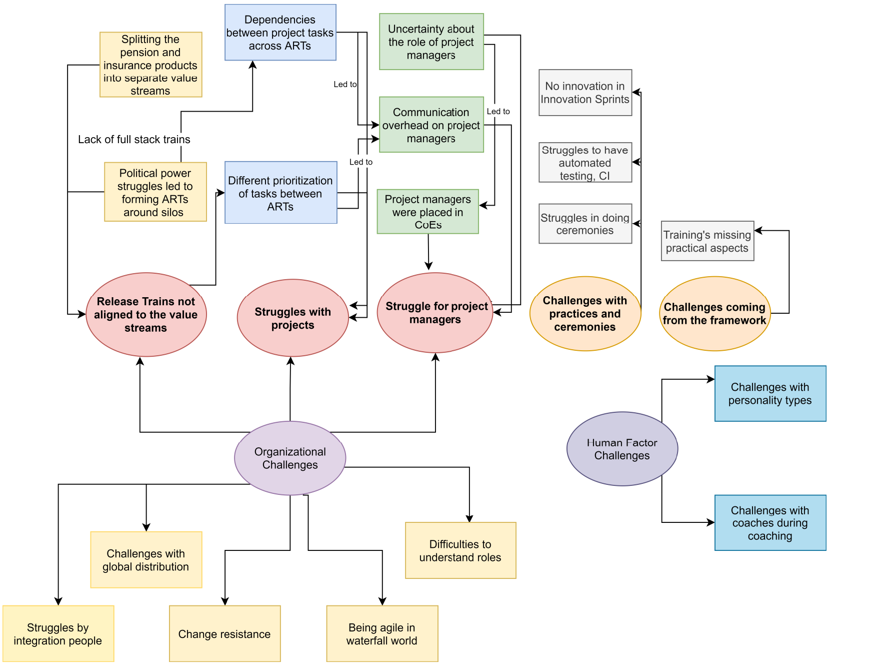
Splitting the pension and insurance products into separate value streams
Dependencies between project tasks across ARTs
Uncertainty about the role of project managers
No innovation in Innovation Sprints
Led to
Communication overhead on project managers
Led to
Lack of full stack trains
Struggles to have automated testing, CI
Led to
Political power struggles led to forming ARTs around silos
Different prioritization of tasks between ARTs
Project managers were placed in CoEs
Struggles in doing ceremonies
Training's missing practical aspects
Release Trains not aligned to the value streams
Struggles with projects
Struggle for project managers
Challenges with practices and ceremonies
Challenges coming from the framework
Challenges with personality types
Human Factor Challenges
Organizational Challenges
Challenges with coaches during coaching
Difficulties to understand roles
Challenges with global distribution
Struggles by integration people
Change resistance
Being agile in waterfall world

Figure 4.9. Challenges of SAFe from Publication V

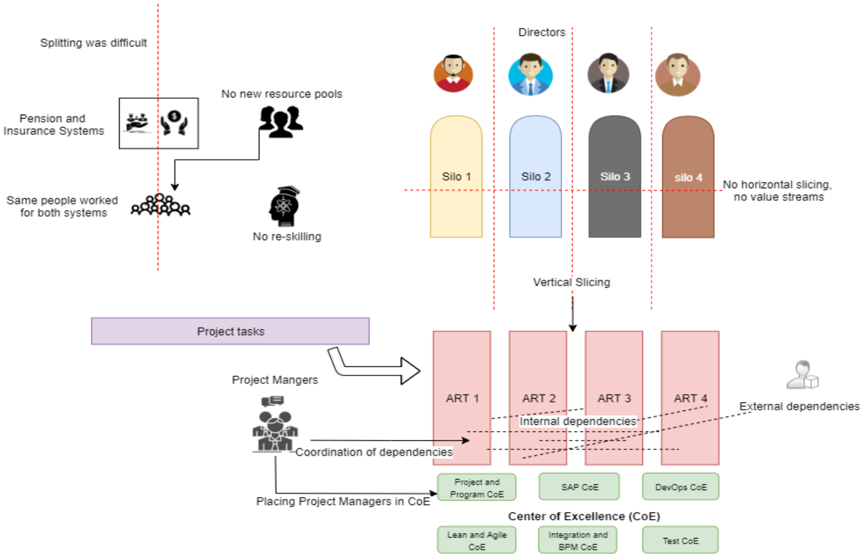

Figure 4.10. Most Critical Challenges from Publication V

and change resistance. The least mean value was found for the framework does not help in resolving dependencies between development teams.

We found that the challenges identified from the three research methods in the publications I, III, IV, V, and VI were similar to each other. We also found that claimed challenges were similar to that of realized challenges.

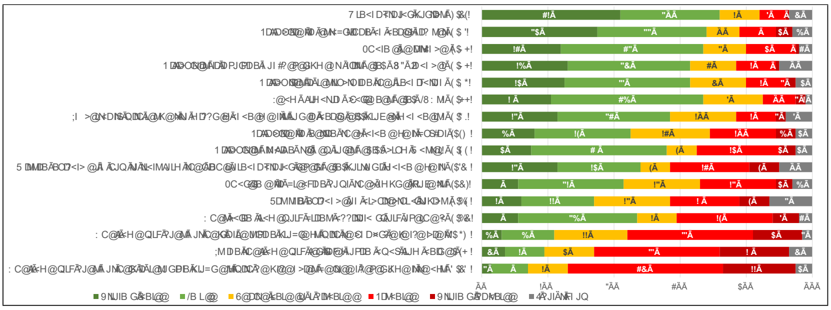

Figure 4.11. Agreement of the Challenges of SAFe, arranged according to the Mean Values

5. Discussion

In this chapter, we will discuss the results presented in Chapter 4. Later, we present the limitations and how we mitigated them in the thesis.

5.1 Goal 1: Understanding the evolution, usage and adoption of scaling frameworks

In this section we discuss the answers related to our goal 1, which is sub-divided into five research questions.

5.1.1 Reasons behind creation of the scaling frameworks

In publication I, we found 12 reasons behind the creation of 15 agile scaling frameworks. The reasons from our survey fall into either the category of improving the organization's current state of dealing with the organization's critical challenges. Both look similar to reasons that trigger an organizational change. Several reasons, e.g., *improving the collaboration and coordination agile teams working on same product* and *dealing with changing environments* were found in previous studies on large-scale agile development [Paasivaara, 2017, Gustavsson, 2019c]. These reasons behind creating scaling frameworks look similar to challenges in large-scale [Dikert et al., 2016], which means scaling frameworks were intended to solve the existing challenges in large-scale organizations. They are also identical to reasons for adopting agile in general [One,]. However, to our knowledge, two reasons found in our survey related to *descaling large product organizations into smaller independent entities* and *improving customer involvement* were not reported by the existing literature on agile methods.

5.1.2 Evolution of scaling frameworks

The scaling framework evolution map found based on our survey in publication I helps different practitioners to visualize the evolution of scaling frameworks. This work is similar to the evolutionary map of agile methods by Abrahamsson et al. [Abrahamsson et al., 2003].

Agile scaling frameworks have been continuously emerging and evolving over the years, Fig. 4.1. They will keep on evolving as the methodologists of agile scaling frameworks seem to be committed to improving them in the future. This continuous evolution resembles the principles of agile, e.g., continuous reflection, and continuous improvement and also welcoming change.

We identified many similarities between the agile movement and the scaling frameworks movement. For instance, both agile methods and agile scaling frameworks movement emerged from parallel innovation. Both these movements were led by consultants and practitioners who supported organizations in scaling the agile methodologies.

Even-though the evolution map visualizes many agile scaling frameworks, users have concentrated on a few frameworks, particularly on *SAFe* and *SoS* [VersionOne, 2020].

A similar observation can be made for agile methods, as Scrum and Extreme Programming were most commonly used out of all the methods presented in agile evolutionary map [VersionOne, 2020, Dikert et al., 2016].

5.1.3 Reasons for adopting scaling frameworks

Based on our results, we understood that organizations adopted scaling frameworks for either framework-specific or organizational/business-related reasons or both. Further, we can classify them into (i) improving the organization's current state and (ii) to deal with the current challenges. We also observed that reasons for adopting scaling frameworks in general were similar to reasons behind creating them found in RQ1.

Several reasons found in publication II were also reported by other studies based on SAFe and LeSS. For instance, scaling to more people in [McMunn and Manketo, 2017, Curtis Michelson, 2019] and remain competitive in [Badanahatti and Pillutla, 2020]. Also, *well-defined and clearly documented*, and *well supported by training and guidance* were reported as significant reasons for adopting SAFe [Schongot and Man, 2018, Inc, a, Inc, 2016c, Inc, 2017a] and LeSS [Uludağ et al., 2019].

Publication II has identified several new reasons such as *inculcating an agile mindset, addressing the needs of regulatory environments, fostering innovation, increasing attractiveness as an employer*, and *having cross-functional/feature teams*. While these reasons were not found in previous literature on scaling frameworks, they might reflect general goals of agile adoption that are, typically, maybe not be specific to a particular framework.

More research comparing the reasons between frameworks, selecting a framework, and investigating framework-specific reasons would help practitioners choose between frameworks.

5.1.4 Benefits and challenges of adopting scaling frameworks

We identified both claimed and expected benefits of adopting agile scaling frameworks. The claimed benefits and expected benefits were similar to the reasons for adopting the scaling frameworks and agile in general [VersionOne, 2020]. In total, we identified 30 claimed benefits from our survey in the publication I.

However, the most common benefit of agile, namely improved productivity [VersionOne, 2020], was not mentioned by any methodologists. We also identified benefits related to *reducing headcount* and *fostering servant leadership*, which were not found in the previous literature on large-scale agile development. More research on benefits is needed to establish scientific evidence of using these frameworks in the industry. It is also crucial to understand which practices have contributed to these benefits. We believe that claimed benefits by the methodologists can be realized when organizations understand the underlying agile principles, values, and mindset.

We identified 22 claimed challenges from 15 scaling frameworks. To our knowledge, none of the framework's official websites has given information related to the difficulties encountered while adopting these frameworks. The most common challenges identified in our study, i.e., *using frameworks as cooking recipes* and *using frameworks without understanding for what reasons they should be applied*, were not reported by previously published empirical studies.

Several claimed challenges look similar to agile transformation challenges in general. Hence, using an agile scaling framework is not a silver bullet for scaling agile in large organizations but a starting point for an agile transformation. Several methodologists mentioned that leaders and change agents should focus on changing people's culture and mindset rather than using frameworks only as cooking recipes. Therefore, practitioners should focus on inculcating agile values, principles and mindset. More studies related to how to focus on inculcating agile mindset and values are encouraged in large-scale agile.

5.1.5 Satisfaction of practitioners after adopting agile scaling frameworks

Based on our results, we found that majority of our respondents were satisfied with the framework they adopted. We did not find any research-based studies that measured satisfaction explicitly. However, we found from our MLR (publication III) and case study (Publication V) that SAFe users improved their employee satisfaction. Another study reported Spotify users improved employee satisfaction [Patil et al., 2016]. We did not get information from other frameworks, e.g., LeSS or S@S, to compare our results due to a lack of research on agile scaling frameworks.

The inferential statistics showed that our respondents' satisfaction differed based on the choice of the frameworks; deeper analysis on why they differ between frameworks is an exciting area for future research.

Based on the Kruskal-Walis test and Post-hoc analysis (refer Table 4.8), we can conclude that respondents using Scrum derived frameworks, e.g., S@S, are more satisfied when compared to SAFe. We could also infer that SAFe is more likely to be adopted for its popularity and claimed benefits rather than actual benefits of its adoption [Putta et al., 2018, Uludağ et al., 2021]. Thus, there is more likeliness among SAFe respondents to shift to another framework.

The software practitioners should always look for empirical evidence of the framework usage, e.g., benefits, rather than just getting drawn by their marketing and popularity. Lack of in-depth research on reasons and how to select a framework is a hindrance for practitioners. We encourage researchers to conduct more research on these topics in future.

5.2 Goal 2: Understand the usage and adoption of SAFe

In this section we discuss the answers related to our goal 1, which is sub-divided into four research questions.

5.2.1 Reasons for adopting SAFe

We identified 16 reasons from our case study and 20 reasons from our survey via the open-ended question for adopting SAFe. We identified similarities between reasons identified from our survey and case study. Typically, SAFe adoption reasons could be divided into three categories: improve the current state or deal with the current challenges or framework-specific. These are similar to the reasons that trigger an organizational change. However, the reasons, financial affordability, and buy-in from management from the case study were not found in our survey. Other reasons such as has a focus lean portfolio management and system thinking, and to support digital ways of working, to have secured delivery were only found in the survey and not the case study. However, this reasons were not reported in the previous literature in agile and also in large scale agile studies.

We identified several framework-specific reasons e.g., comprehensive framework, well-defined, focus on lean portfolio management, which were not identified in the current scientific literature. Another unique reason, the financial affordability of the organization, was mentioned in our case study. It is already evident that SAFe transformations need a lot of capital, e.g., for coaching, hiring external consultants, It may be easier for many financial organizations, like our case, which is financially stable, to afford this framework. According to our MLR (Publication III), this could be one of the reasons why the financial domain had the highest number of SAFe transformations.

The popularity of SAFe has been the most prevalent reason for SAFe adoption. SAFe in many organizations is adopted just because other similar organizations have embraced it. One of the reasons for SAFe popularity could be due to its marketing, e.g., publishing case studies inclined towards reporting benefits, which was reflected in our MLR (Publication III). Such case studies create a positive affect towards the benefits, which in due course increases the adoption rate.

It would be ideal for practitioners to understand what the framework offers rather than just going with the popularity. This generates a need for more research on framework-specific reasons to help organizations map their needs with what the framework provides. It would also be interesting to compare the framework-specific reasons between the different frameworks.

5.2.2 Activities involved in doing a SAFe transformation

We identified several activities during the transformation to SAFe from our case study. Typically, the majority of the activities identified from our case study had a mapping with SAFe implementation roadmap.

The SAFe transformation was driven by top-level leaders and was proceeded step-wise. This type of transformation is also prevalent in large-scale agile transformations [Dikert et al., 2016] and also across other organizations taking SAFe [Paasivaara, 2017, Conboy and Carroll, 2019a]. The foremost activity of the transformation was communicating reasons and creating awareness to the people across the organization regarding SAFe. However, several people from the team level did not know the reasons for the transformation, which confirms that the transition was purely a top-down approach with the directors and leaders as driving forces. Communicating the reasons helps in preparing for the upcoming cultural shock. Lack of proper communication could also be a reason for strong resistance in our case. The SAFe roadmap also recommends communicating change and vision during transformation [Inc, b]. Similar information on communicating change was mentioned in [Paasivaara, 2017].

Forming the transformation teams to support the transformation process was an important step. In like manner, transformation teams were formed in other organizations taking SAFe as well [post, 2017, Inc, 2016a, Scaled Agile Inc., 2017, Inc, 2016b, Limited, 2018, McMunn and Manketo, 2017]. Several names have been used to refer to these teams in the previous studies. However, LACE is the official name prescribed by the SAFe roadmap [Scaled Agile Inc., c].

Another significant activity of the transformation was forming a pilot and later other ARTs. SAFe roadmap of implementation also suggests starting with a single ART. However, the word pilot is not mentioned by the roadmap [Scaled Agile Inc., c]. Other cases from literature started a single ART and then expanded to more number of ARTs similar to our case [Paasivaara, 2017, Pries-Heje and Krohn, 2017, Yael Man, 2017, McMunn and Manketo, 2017]. Our case study also provided detailed information on ART formation, e.g., conducting Lego workshops and design workshops, which is not provided in the current literature. The SAFe website also lacks detailed description on how to form these ARTs. More research on this topic is needed to understand the formation ARTs to assist the practitioners.

Our case invested a lot in educating people on SAFe with different SAFe prescribed trainings, e.g., Leading SAFe, SAFe for teams, and implementing SAFe. Similar trainings were found from previously published studies [post, 2017, Inc, 2017b, Inc., , Inc, 2016d, Inc, a]. Apart from the usual SAFe training's, our case had Scrum tours for having basic knowledge on agile, leadership, prioritization as typical SAFe trainings lacked these basics. Several other cases also had such kind of basic agile trainings [Inc, 2017a, Scaled Agile Inc., 2016, Inc, 2017b, software, 2015, Inc, 2017c, Scaled Agile Inc., 2018], which emphasizes the need for incorporating more agile and lean training, especially for organizations that are new to agile.

Our case did not entirely rely on external partners. Instead, they had some handpicked consultants and external partners, which helped them have some very experienced coaches to guide the journey. Such mixture also helped to have different perspectives of the transformation. Whereas, several cases [Pries-Heje and Krohn, 2017, Paasivaara, 2017, McMunn and Manketo, 2017] from the literature were primarily dependent only on these scaled agile partners. Future SAFe adopters also could try to have a combination of both handpicked consultants and external partners.

5.2.3 Benefits of SAFe adoption

In our research we identified expected, claimed, and realized benefits of SAFe adoption. By an large all these categories of the benefits were similar to each other. Also, expected benefits and claimed benefits had similarities with reasons of adopting scaling frameworks found in our Publication II and also to the reasons for agile adoption in [One,].

The biggest benefits of SAFe reported based on our MLR, case study, and survey (three research methods) are improved transparency, collaboration, and faster time-to-market. SAFe practices such as PI plannings and system demos, have helped to improve transparency and collaboration. Many organizations using traditional ways of working before SAFe observed a shift from long delivery cycles to short ones (in Publication III). However, organizations using agile before adopting SAFe felt like they are moving away from agile due to fixed 3-month increments [Paasivaara, 2017, Turetken et al., 2017].

We identified some unique benefits of SAFe related to cost reduction and clear prioritization which were not mentioned in existing literature on scaling frameworks and agile in large. However, these benefits were mentioned in agile related studies. Even-though many of our benefits matched with agile in general, we had additional information on what practices and principles enabled to get these results.

SAFe website has given some metrics to measure the quantitative benefits, however, to our knowledge we did not find any metrics for measuring the qualitative benefits. We could not get detailed information on metrics either from our MLR (Publication III) nor our case study, as our case study (Publication V) had mostly reported qualitative benefits. Our MLR also lacked information on metrics due to poor information on this topic from the case studies published on the SAFe webpage. We need to collect numeric data (e.g. number of deliveries per year) on benefits in order to establish an evidence of benefits of using SAFe in future studies.

5.2.4 Challenges of SAFe adoption

In this thesis we have identified challenges of adopting SAFe by conducting an MLR, case study, and a survey. Our both primary studies (case study and survey) showed a balance in reporting the benefits and challenges. However, the MLR, which was mostly based on case studies from SAFe website was more inclined towards the benefits. Only few studies reported the challenges on SAFe.

The most significant challenges are related to *organizational politics, change resistance and inculcating agile mindset, and identifying value streams and formation of ARTs*. The first two challenges are also common among other scaling frameworks and agile in large scale as well [Dikert et al., 2016]. The last challenge is specific to SAFe.

It is evident that SAFe or agile transformation is more than just implementing the practices. Practitioners should focus on inculcating agile mindset and principles. Communicating the importance of transformation could help in overcoming the resistance from different levels in the organization [Paasivaara, 2017].

Apart from aforesaid organizational challenges, we identified shortcomings from SAFe itself. For instance, SAFe training's missed the practical aspects of its implementation and also SAFe roadmap lacks in-depth information on value streams and ART formation. Other scientific studies on SAFe also pointed out shorting comings from the framework, e.g., Jos Trienekens et al. in [Trienekens et al., 2018], mentioned that SAFe does not have an explicit focus regarding customer involvement and suggested incorporating a customer involvement level in the program, value stream as well as in team level. Sreenivasan et al. [Sreenivasan and Kothandaraman, 2019], reported SAFe lacks standardization for estimation.

The framework developers can get an insight into these challenges for improving the framework, e.g., adding some training's on agile and lean, especially for traditional organizations without prior agile experience.

5.3 Implications for Research

This study has provides the following implications for researchers.

- We provided an overview of the evolution of agile scaling frameworks and why they were created directly from the people who made them. Researchers are now provided with a comparison of reasons between popular agile scaling frameworks that included internally developed methods as well.

- We identified several framework-specific reasons and some generic reasons that are common among agile literature. Understanding such specific reasons can serve as a starting point to create more research on selecting an appropriate framework.

- We were also able to measure respondents' satisfaction using different frameworks, which was not investigated in the current literature. We were able to identify differences in satisfaction between different frameworks. More research on measuring satisfaction could help select the framework based on the satisfaction level of the respondents.

- Our research also gave in-depth information related to the SAFe transformation process, not found in the previous literature. Now researchers are provided more in-depth information on value streams and ARTs. Researchers could compare this information to other SAFe organizations and also with other scaling frameworks.

- We found several benefits for adopting SAFe from the case study, literature review, and survey. In the case study, some metrics were reported for benefits such as employee satisfaction and productivity. The primary studies identified in our MLR also did not mention much about the metrics, reinforcing the need for more research into measuring the benefits to establish scientific evidence.

- We identified several challenges of adopting scaling frameworks. We identified some specific challenges of adopting SAFe, not found in the literature. Some challenges remain common between scaling frameworks, agile in large scale. For example, change resistance and inculcating an agile mindset have been prevalent challenges since the beginning of agile methods. There is a need for more solution-centric research in large-scale agile, e.g., on how to inculcate agile mindset and overcome change resistance during the transformation.

5.4 Implications for Practice

The study provides the following implications for practitioners.

- Many practitioners adopted scaling frameworks due to their popularity, especially SAFe. Our research would assist the practitioners in reflecting on: why do they need a particular framework, did they reach a tipping point, where the existing methods cannot be used to develop a product, and if they are motivated for a change for a better future state of the organization.

- Common challenges of agile in large existed even after adopting the frameworks. Scaling frameworks are not a silver bullet to overcome the scaling related challenges during transformation. Therefore, analyzing the current challenges and goals of the organization and mapping with characteristics of the scaling frameworks is needed before selecting a framework.

- A significant challenge during the agile transformation is inculcating an agile mindset. The practices cannot be implemented if the underlying principles are not understood. Organizations should give focus on principles, mindset and culture. As training offered by the framework, e.g., SAFe, LeSS are framework-specific and do not focus on agile mindset and culture. Many practitioners believe that training people in framework-specific training could lead to a successful implementation of the frameworks. Training's give a glimpse of framework practices and implementation. The people inside the organization need to apply them to their context by understanding the underlying principles. An organization must cultivate the agile mindset, especially in traditional organizations, with less or no agile experience.

In the second part of this thesis, we focused only on SAFe. Therefore, we have some specific implications for the SAFe practitioners. However, this can be applicable for practitioners using other frameworks if they have similar challenges.

- Several cases had many struggles in getting the right value streams due to organizational politics and difficulties in splitting the tightly coupled systems. Practitioners should carefully evaluate these challenges and mitigate them from the very beginning of the transformation. If these are not addressed initially, there might be many problems with managing dependencies and coordination overhead.

- Many large scale organizations have projects. Working with projects along with ARTs creates dependencies and clashes with the delivery cycles. Organizations should decide on the projects, i.e., whether to continue them or not in the future? If they want to continue, then proper rules for project tasks should be established in advance. For the best setup, it is always better to get rid of projects.

- There is no role for project managers in the SAFe or agile setup. However, many cases tend to have them for coordination between ARTs for managing the dependencies. In our case, these managers were placed in CoEs. Many project managers reflected uncertainties about their roles. Most organizations cannot fire the managers as they are talented and have a long association with them (e.g., 10-20 years). Many software industries, especially, in Europe have job security and role security; due to such legal norms, firms cannot take any drastic decisions regarding changing roles or firing them. Practitioners should carefully evaluate the suitable roles for project managers and enable a safe transition into the roles.

- SAFe provides a well-defined structure, roles, guidance to scale agile. It might be easy for organizations to adopt it, as they directly map the existing roles to the new ones defined in SAFe, which is also a common reason for adopting SAFe. For instance, project managers take up RTEs, without having complete knowledge about the new role. It might be simple to adopt SAFe, but difficult to inculcate the agile mindset and culture. Therefore, organizations should focus on principles rather than adhering to the framework practices and customize them based on the context.

5.5 Threats to validity

This section will summarize the different threats we encountered during our research journey and how we mitigated them. Descriptions of the threats of individual studies are presented in the respective publications. This thesis employed three different research methods: literature review, a case study, and two surveys. Each method had its own threats to validity, and they are described based on the terminology defined in experimental software engineering and [Wohlin et al., 2012] and literature reviews [Zhou et al., 2016]. First, we would describe the threats specific to MLR, then describe the common threats among all the research methods.

Selection Bias. This occurs during the selection of primary studies based on interpretation of inclusion criteria. We mitigated this by involving all authors in designing the criteria. Two authors filtered the abstracts and titles of peer-reviewed articles independently. Regarding grey literature, we took all the case studies published on the SAFe official website, which completely mitigated the threat of selection bias across the selected grey literature. However, there might be additional relevant grey literature from other sources that we did not include.

Subjective Bias. This threat occurs during the coding of qualitative data. Coding was meticulously performed by one of the authors, and there was a series of discussions during the coding process among all three authors regarding the naming of the axial codes and categorization of the open codes into axial codes. The process is traceable.

Publication Bias. Including grey literature could be seen as a serious limitation of this study. Grey literature articles usually describe positive results of a particular case [Zhou et al., 2016]. This was also evident from this study, as the majority of these cases gave attention to the benefits of the framework. This threat was partially mitigated by comparing the findings to peer-reviewed studies identified from the database search and also to existing literature on large-scale agile [Version One, b, Laanti and Kettunen, 2019, Dikert et al., 2016]. This validated some of the main categories evolved during the coding process. Scaled Agile team specialists have reviewed all case studies reported by the organizations. The Scaled Agile team might have influenced the organizations to present only the positive elements of the SAFe adoption process to market the framework and get buy-in from large-scale organizations.

Internal Validity. The threat is concerned with different factors that could effect the relation between the research process and obtained results, e.g., selection of interviewees/respondents bias. We made of list of potential interviewees during the PI planning session at the our case organization, and this list was checked by the one of the core member from LACE team. Then some other additional interviewees were suggested. This ensured were collected data from the right people.

We mitigated the respondent bias by collecting data from reliable sources, i.e., largely our responses (more than 90%) came from people we met during the conferences and agile Meetups and via personal contacts, and we knew that they were using scaling frameworks, which helped us to avoid unreliable responses

External Validity. This threat is concerned with ability to generalize the results to other contexts. The results of our case study can applied to an organization having similar contextual information, e.g., pension and insurance systems, having tightly coupled products. We found many similarities between our cases and previously published cases, with respect to reasons, transformation process, benefits and challenges.

Our survey data results on demographics was similar to other surveys in the field, giving us some degree of confidence that our sample at least does not completely misrepresent the population. Our results on challenges, benefits, reasons, satisfaction resembled with the previous studies on agile scaling frameworks.

Construct Validity. This threat is applicable to both survey and case study, which deals with measuring the right concepts that are intended to be measured. We formulated the survey statements based on earlier findings in the domain of agile and large-scale agile. We had to limit the questionnaire length, and we could not include all the possible reasons, challenges, that we identified from the literature. We compensated for this through open questions, which we think helped get the most probable reasons, benefits, challenges for agile scaling framework adoption.

The interview questionnaire was designed iteratively by collecting feedback and improving the questions to make them clear and understandable to the interviewees. Interviewees belonged to several roles that helped triangulate and make sure the questions were interpreted in the proper manner.

Reliability. This threat is concerned with the extent of repetition of the study. The case study data was collected from multiple roles, different levels of the organization, ensuring the correctness of the data. Our survey data also was collected from different geographical locations, different roles, using different frameworks. We believe that same survey conducted in future might result in similar results.

6. Conclusion and Future Work

In this chapter we present the conclusions and propose future work.

6.1 Conclusions

This thesis is an exploratory research aimed at investigating the usage and adoption of scaling frameworks. We used mixed-methods: survey, case study, and an MLR and to investigate this topic. The first part of the thesis aimed to explore several frameworks and second part was focused only on SAFe. We would briefly answer the research questions in the following.

6.1.1 What are reasons behind designing agile scaling frameworks?

The common reasons for designing frameworks include improving the agility/adaptability of the organization, improving the collaboration of agile teams working on the same product, improving the coordination of agile teams working, and improving the synchronization of agile teams working on the same product. We found two new reasons which were not reported by the existing literature on agile development: *descaling large product organizations into smaller independent entities* and *improving customer involvement*.

6.1.2 How did the agile scaling frameworks evolve over the years?

A significant number of scaling frameworks came to light between 2011 and 2018. The majority of the frameworks evolved by releases multiple versions after their first release.

6.1.3 What are reasons for adopting agile scaling frameworks?

The most common reasons for adopting scaling frameworks are: to scale more people, remain competitive, and deal with the organisation's pain points.

The popularity of the framework, well-defined and documented framework, support and guidance from coaches were significant reasons (based on the inferential statistics) for adopting SAFe when compared to LeSS, Spotify, and internally created methods.

6.1.4 What are the claimed benefits and challenges of adopting scaling frameworks?

We identified 30 different claimed benefits of adopting scaling frameworks, which are categorized into business, product, organizational, and cultural aspects. The methodologists also reported two new benefits which were not described in the previous literature: *reducing headcount* and *fostering servant leadership*.

The methodologists recognized 22 challenges in the adoption of the frameworks, of which two were newly discovered in our study, i.e., *using frameworks as cooking recipes* and *using frameworks without understanding for what reasons they should be applied*.

6.1.5 How satisfied are the practitioners after using agile scaling frameworks?

Based on our survey data, the majority of the practitioners are satisfied with the current framework selection. Based on inferential statistics, we observed that respondents using S@S had the highest satisfaction compared to LeSS, DAD, Spotify, and internally created methods. A statistical significance was also found between S@S, LeSS, and internally created methods. Respondents using SAFe were more likely to shift towards other frameworks when compared to respondents of other frameworks.

6.1.6 What are reasons for adopting SAFe?

The reasons for adopting SAFe could be divided into three categories: organizational, business and framework-specific. The most common reasons for SAFe adoption were faster market time, improved collaboration, and well-defined and comprehensive framework. Financial affordability and framework-specific reasons were the unique reasons from our research, which was not found in the previous literature.

6.1.7 What activities are involved in doing a SAFe transformation?

Top-down approach of SAFe transformation was seen in our case study. The transformation proceeded step-wise. The most significant activities identified during the SAFe transformation were piloting and ART formation, educating people in SAFe, and Scrum tours.

The activities identified from our study are very well-mapped with the steps from the SAFe roadmap. The roadmap has very little information on forming of ARTs. In our study, we found in-depth information related to ARTs and value streams.

6.1.8 What are benefits of adopting SAFe?

The most commonly mentioned benefits of SAFe were related to improved transparency, improved transparency, collaboration, and faster time-to-market. We identified some unique benefits of SAFe related to cost reduction and clear prioritization, which were not mentioned in the existing literature on scaling frameworks and agile in large.

6.1.9 What are challenges of adopting SAFe?

The biggest challenges of SAFe adoption are: identifying value streams and formation of ARTs, organizational politics, change resistance and inculcating an agile mindset. More research on how to mitigate these challenges will help create more solution-centric research as the research on a large scale is dominated by problem-centric research.

6.2 Future Work

We encourage researchers to conduct extensive research on gaps that are identified from our studies. They are as follows:

Primary studies. The number of studies on SAFe, Spotify and LeSS has increased in recent years. However, there is a need for more in-depth empirical investigations on other agile scaling frameworks, especially S@S, DAD, and Nexus.

Benefits. The present literature does not give any substantial evidence on the benefits of scaling frameworks usage and adoption. Also, which practices helped get reported benefits and how organizations measured them could be investigated.

Challenges and mitigation. It would be interesting to find out how to mitigate the challenges organizations encounter while adopting scaling frameworks. Additionally, a distinction between the challenges from certain practices and approaches of scaling frameworks and inherent organizational challenges would be an exciting area to research.

Comparison between scaling frameworks. The reasons, benefits, transformation process and challenges could be compared among the frameworks such as Large Scale Scrum (LeSS), Disciplined Agile Delivery (DAD), and Nexus. The suitability of frameworks to various organisational contexts could also be determined.

Further studies on the impact of contextual factors such as agile maturity, complexity, or multi-product development on the suitability of different frameworks could provide organizations with a decision framework to aid in framework selection.

Bibliography

[Abbas et al., 2008] Abbas, N., Gravell, A. M., and Wills, G. B. (2008). Historical roots of agile methods: Where did “agile thinking” come from? In *International conference on agile processes and extreme programming in software engineering*, pages 94–103. Springer.

[Abrahamsson et al., 2003] Abrahamsson, P., Warsta, J., Siponen, M. T., and Ronkainen, J. (2003). New directions on agile methods: A comparative analysis. In *Proceedings of the 25th International Conference on Software Engineering*, pages 244–254. IEEE.

[Alliance,] Alliance, A. Disciplined Agile Delivery. https://www.agilealliance.org/wp-content/uploads/2016/01/Disciplined-Agile-Delivery-RAW14261USEN.pdf. [Online; accessed 06-APR-2021].

[Alqudah and Razali, 2016] Alqudah, M. and Razali, R. (2016). A review of scaling agile methods in large software development. *International Journal on Advanced Science, Engineering and Information Technology*, 6(6):828–837.

[Alsaqaf et al., 2020] Alsaqaf, W., Daneva, M., and Wieringa, R. J. (2020). Analysing large-scale scrum practices with respect to quality requirements challenges. In *ICSOFT*, pages 92–103.

[Ambler, 2007] Ambler, S. W. (2007). Agile software development at scale. In *IFIP Central and East European Conference on Software Engineering Techniques*, pages 1–12. Springer.

[Ampatzoglou et al., 2015] Ampatzoglou, A., Ampatzoglou, A., Chatzigeorgiou, A., and Avgeriou, P. (2015). The financial aspect of managing technical debt: A systematic literature review. *Information and Software Technology*, 64:52–73.

[Badanahatti and Pillutla, 2020] Badanahatti, A. and Pillutla, S. (2020). Interleaving software craftsmanship practices in medical device agile development. In *Proceedings of the 13th Innovations in Software Engineering Conference on Formerly known as India Software Engineering Conference*, pages 1–5.

[Balaji and Murugaiyan, 2012] Balaji, S. and Murugaiyan, M. S. (2012). Waterfall vs. v-model vs. agile: A comparative study on sdlc. *International Journal of Information Technology and Business Management*, 2(1):26–30.

[Bass, 2019] Bass, J. M. (2019). Future trends in agile at scale: A summary of the 7 th international workshop on large-scale agile development. In *International Conference on Agile Software Development*, pages 75–80. Springer.

[Bass and Salameh, 2020] Bass, J. M. and Salameh, A. (2020). Agile at scale: a summary of the 8th international workshop on large-scale agile development. In *Agile Processes in Software Engineering and Extreme Programming–Workshops*, page 68.

[Beck et al., 2001] Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al. (2001). Manifesto for agile software development.

[Beecham et al., 2021] Beecham, S., Clear, T., Lal, R., and Noll, J. (2021). Do scaling agile frameworks address global software development risks? an empirical study. *Journal of Systems and Software*, 171:110823.

[Berger and Beynon-Davies, 2009] Berger, H. and Beynon-Davies, P. (2009). The utility of rapid application development in large-scale, complex projects. *Information Systems Journal*, 19(6):549–570.

[Bjarnason et al., 2011] Bjarnason, E., Wnuk, K., and Regnell, B. (2011). A case study on benefits and side-effects of agile practices in large-scale requirements engineering. In *proceedings of the 1st workshop on agile requirements engineering*, pages 1–5.

[Boehm and Turner, 2005] Boehm, B. and Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE software*, 22(5):30–39.

[Boehm, 1988] Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5):61–72.

[Boehm et al., 2004] Boehm, B. W., Boehm, B., and Turner, R. (2004). *Balancing agility and discipline: A guide for the perplexed*. Addison-Wesley Professional.

[Booch, 1995] Booch, G. (1995). *Object solutions: managing the object-oriented project*. Addison Wesley Longman Publishing Co., Inc.

[Carlile, 2002] Carlile, P. R. (2002). A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization science*, 13(4):442–455.

[Carroll et al.,] Carroll, N., Bjørnson, F. O., Dingsøyr, T., Rolland, K.-H., and Conboy, K. Operationalizing agile methods: Examining coherence in large-scale agile transformations. In *Proceedings of the 21th International Conference on Agile Software Development: Agile Processes in Software Engineering and Extreme Programming – Workshops*, pages 75–83. Springer, Cham.

[Christopher and De Vries, 2020] Christopher, L. A. and De Vries, M. (2020). Selecting a scaled agile approach for a fin-tech company. *South African Journal of Industrial Engineering*, 31(3):196–208.

[Coad et al., 1999] Coad, P., Luca, J. d., and Lefebvre, E. (1999). *Java modeling color with UML: Enterprise components and process with Cdrom*. Prentice Hall PTR.

[Cohen et al., 2004] Cohen, D., Lindvall, M., and Costa, P. (2004). An introduction to agile methods. *Adv. Comput.*, 62(03):1–66.

[Conboy and Carroll, 2019a] Conboy, K. and Carroll, N. (2019a). Implementing large-scale agile frameworks: challenges and recommendations. *IEEE Software*, 36(2):44–50.

[Conboy and Carroll, 2019b] Conboy, K. and Carroll, N. (2019b). Implementing large-scale agile frameworks: Challenges and recommendations. *IEEE Software*, 36(2):44–50.

[Conover, 1998] Conover, W. J. (1998). *Practical nonparametric statistics*, volume 350. John Wiley & Sons.

[Corbin and Strauss, 2008] Corbin, J. M. and Strauss, A. L. (2008). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, Inc., Los Angeles, Calif., 3rd ed edition.

[Cram, 2019] Cram, W. A. (2019). Agile development in practice: Lessons from the trenches. *Information Systems Management*, 36(1):2–14.

[Creswell, 1999] Creswell, J. W. (1999). Mixed-method research: Introduction and application. In *Handbook of educational policy*, pages 455–472. Elsevier.

[Curtis Michelson, 2019] Curtis Michelson, S. A. (2019). Experience reports from agile alliance bias from the bottom: A different way to bootup a safe train. In *International Conference on Agile Software Development*. AgileAlliance.

[Das and Gary, 2021] Das, S. and Gary, K. (2021). Agile transformation at scale: A tertiary study. In *International Conference on Agile Software Development*, pages 3–11. Springer.

[Diebold and Mayer, 2017] Diebold, P. and Mayer, U. (2017). On the usage and benefits of agile methods & practices. In *International Conference on Agile Software Development*, pages 243–250. Springer, Cham.

[Diebold et al., 2018] Diebold, P., Schmitt, A., and Theobald, S. (2018). Scaling agile: how to select the most appropriate framework. In *Proceedings of the 19th International Conference on Agile Software Development: Companion*, pages 1–4.

[Dikert et al., 2016] Dikert, K., Paasivaara, M., and Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, 119:87–108.

[Dingsøyr et al., 2014a] Dingsøyr, T., Fægri, T. E., and Itkonen, J. (2014a). What is large in large-scale? a taxonomy of scale for agile software development. In *International Conference on Product-Focused Software Process Improvement*, pages 273–276. Springer.

[Dingsøyr et al., 2014b] Dingsøyr, T., Fægri, T. E., and Itkonen, J. (2014b). What Is Large in Large-Scale? A Taxonomy of Scale for Agile Software Development. In *Proceedings of the 15th International Conference on Product-Focused Software Process Improvement*, Cham. Springer.

[Dingsøyr et al., 2018] Dingsøyr, T., Moe, N. B., and Ohlsson, H. H. (2018). Towards an understanding of scaling frameworks and business agility: A summary of the 6th international workshop at xp2018. *arXiv preprint arXiv:1812.10280*.

[Dingsøyr et al., 2012] Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development.

[Dybå and Dingsøyr, 2008] Dybå, T. and Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, 50(9-10):833–859.

[Edison et al., 2021] Edison, H., Wang, X., and Conboy, K. (2021). Comparing methods for large-scale agile software development: A systematic literature review. *IEEE Transactions on Software Engineering*.

[Fowler, 1995] Fowler, F. J. (1995). *Improving survey questions: Design and evaluation*, volume 38. Sage.

[Framework,] Framework, N. Nexus Framework. http://growing-agility.com/nexus/. [Online; accessed 06-APR-2021].

[Fuchs and Hess, 2018a] Fuchs, C. and Hess, T. (2018a). Becoming agile in the digital transformation: The process of a large-scale agile transformation. In *Proceedings of the 39th International Conference on Information Systems (ICIS)*.

[Fuchs and Hess, 2018b] Fuchs, C. and Hess, T. (2018b). Becoming agile in the digital transformation: the process of a large-scale agile transformation.

[Garousi et al., 2016] Garousi, V., Felderer, M., and Mäntylä, M. V. (2016). The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. In *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, page 26. ACM.

[Garousi et al., 2019a] Garousi, V., Felderer, M., and Mäntylä, M. V. (2019a). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology*, 106:101–121.

[Garousi et al., 2019b] Garousi, V., Felderer, M., Mäntylä, M. V., and Rainer, A. (2019b). Benefitting from the grey literature in software engineering research. *arXiv preprint arXiv:1911.12038*.

[Garousi and Mäntylä, 2016] Garousi, V. and Mäntylä, M. V. (2016). When and what to automate in software testing? a multi-vocal literature review. *Information and Software Technology*, 76:92–117.

[Gerster et al., 2020] Gerster, D., Dremel, C., Brenner, W., and Kelker, P. (2020). How enterprises adopt agile forms of organizational design: a multiple-case study. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, 51(1):84–103.

[Gustavsson, 2018] Gustavsson, T. (2018). Impacts on team performance in large-scale agile software development. In *2018 Joint of the 17th Business Informatics Research Short Papers, Workshops and Doctoral Consortium, BIR-WS 2018, 24 September 2018 through 26 September 2018*, volume 2218, pages 421–431. CEUR-WS.

[Gustavsson, 2019a] Gustavsson, T. (2019a). Changes over time in a planned inter-team coordination routine. In *International Conference on Agile Software Development*, pages 105–111. Springer.

[Gustavsson, 2019b] Gustavsson, T. (2019b). Dynamics of inter-team coordination routines in large-scale agile software development.

[Gustavsson, 2019c] Gustavsson, T. (2019c). Dynamics of inter-team coordination routines in large-scale agile software development. In *Proceedings of the 27th European Conference on Information Systems (ECIS)*.

[Gustavsson, 2019d] Gustavsson, T. (2019d). Voices from the teams-impacts on autonomy in large-scale agile software development settings. In *International Conference on Agile Software Development*, pages 29–36. Springer.

[Gustavsson, 2020] Gustavsson, T. (2020). Visualizing inter-team coordination. In *Proceedings of the Evaluation and Assessment in Software Engineering*, pages 306–311.

[Gustavsson and Bergkvist, 2019] Gustavsson, T. and Bergkvist, L. (2019). Perceived impacts of using the scaled agile framework for large-scale agile software development.

[Heikkilä et al., 2015] Heikkilä, V. T., Paasivaara, M., Rautiainen, K., Lassenius, C., Toivola, T., and Järvinen, J. (2015). Operational release planning in large-scale scrum with multiple stakeholders–a longitudinal case study at f-secure corporation. *Information and Software Technology*, 57:116–140.

[Hirsch, 2005] Hirsch, M. (2005). Moving from a plan driven culture to agile development. In *International Conference on Software Engineering*, volume 27, page 38.

[In,] In, L. Linked-In Professional Network. https://www.linkedin.com/.

[Inc.,] Inc., S. A. CSG Case Study. http://www.scaledagileframework.com/csg-case-study-2/. [Online; accessed 09-June-2018].

[Inc, a] Inc, S. A. Intel Case study. http://www.scaledagileframework.com/case-study-intel/. [Online; accessed 06-APR-2020].

[Inc, b] Inc, S. A. Large Solution SAFe Configuration. https://www.scaledagileframework.com/large-solution-safe/ . [Online; accessed 06-APR-2020].

[Inc.,] Inc., S. A. The Essential SAFe Configuration. https://www.scaledagileframework.com/essential-safe/ . [Online; accessed 06-APR-2020].

[Inc, a] Inc, S. A. The Portfolio SAFe configuration. https://www.scaledagileframework.com/portfolio-safe/ . [Online; accessed 06-APR-2020].

[Inc, b] Inc, S. A. Tipping to SAFe. https://www.scaledagileframework.com/reaching-the-tipping-point/ .

[Inc.,] Inc., S. A. What is New in SAFe 5.0. https://www.scaledagileframework.com/whats-new-in-safe-5-0/.

[Inc, 2016a] Inc, S. A. (2016a). Amdocs Case Study. http://www.scaledagileframework.com/amdocs-case-study. [Online; accessed 06-APR-2020].

[Inc, 2016b] Inc, S. A. (2016b). AstraZeneca Case Study. http://www.scaledagileframework.com/astra-zeneca-case-study/. [Online; accessed 09-June-2018].

[Inc, 2016c] Inc, S. A. (2016c). HPE Case Study. http://www.scaledagileframework.com/hpe-case-study/. [Online; accessed 06-APR-2020].

[Inc, 2016d] Inc, S. A. (2016d). Westpac Case Study. http://www.scaledagileframework.com/westpac-case-study/. [Online; accessed 09-June-2018].

[Inc, 2017a] Inc, S. A. (2017a). Capital One Case Study. http://www.scaledagileframework.com/capital-one-case-study/. [Online; accessed 06-APR-2020].

[Inc, 2017b] Inc, S. A. (2017b). Standard Bank Case Study. http://www.scaledagileframework.com/standard-bank-case-study/. [Online; accessed 09-June-2018].

[Inc, 2017c] Inc, S. A. (2017c). Vantiv Case Study. http://www.scaledagileframework.com/vantiv-case-study. [Online; accessed 09-June-2018].

[Irons, 2006] Irons, A. (2006). Agile methods silver bullet or red herring. *Northumbria University. UK. An online document that can be found at http: / /www. newcastle. bcs. org / BCS% 20Agile% 20methods. ppt (Last accessed: July 2006)[The year 2006 mentioned in this reference corresponds to the year of last access of this online document. The year of publication of this document is unknown].*

[Kalenda et al., 2018] Kalenda, M., Hyna, P., and Rossi, B. (2018). Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software: Evolution and Process*, 30(10):e1954.

[Keele et al., 2007] Keele, S. et al. (2007). Guidelines for performing systematic literature reviews in software engineering. In *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. sn.

[Klünder et al., 2018] Klünder, J., Hohl, P., and Schneider, K. (2018). Becoming agile while preserving software product lines: An agile transformation model for large companies. In *Proceedings of the 2018 International Conference on Software and System Process*, pages 1–10. ACM.

[Kniberg and Ivarsson, 2012] Kniberg, H. and Ivarsson, A. (2012). Scaling agile@ spotify. *online], UCVOF, ucvox. files. wordpress. com / 2012 / 11 / 113617905-scaling-Agile-spotify-11. pdf.*

[Kuhrmann et al., 2017] Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trektere, K., McCaffery, F., Linssen, O., Hanser, E., et al. (2017). Hybrid software and system development in practice: waterfall, scrum, and beyond. In *Proceedings of the 2017 International Conference on Software and System Process*, pages 30–39.

[Laanti and Kettunen, 2019] Laanti, M. and Kettunen, P. (2019). Safe adoptions in finland: a survey research. In *Proceedings of the 20th International Conference on Agile Software Development: Agile Processes in Software Engineering and Extreme Programming – Workshops*, pages 81–87. Springer, Cham.

[Laanti et al., 2013] Laanti, M., Similä, J., and Abrahamsson, P. (2013). Definitions of agile software development and agility. In *European Conference on Software Process Improvement*, pages 247–258. Springer.

[Larman and Basili, 2003] Larman, C. and Basili, V. R. (2003). Iterative and incremental developments. a brief history. *Computer*, 36(6):47–56.

[Larman and Vodde, 2010] Larman, C. and Vodde, B. (2010). *Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum*. Pearson Education.

[Larman and Vodde, 2016] Larman, C. and Vodde, B. (2016). *Large-scale scrum: More with LeSS*. Addison-Wesley Professional.

[Leffingwell, 2007] Leffingwell, D. (2007). *Scaling Software Agility: Best Practices for Large Enterprises*. Pearson Education.

[Lilliefors, 1967] Lilliefors, H. W. (1967). On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402.

[Limited, 2018] Limited, E. S. (2018). Edge Verve Case Study. https://www.scaledagileframework.com/wp-content/uploads/delightful-downloads/2018/05/Infosys-EV-Biz-Agility-Casestudy.pdf . [Online; accessed 09-June-2018].

[Linåker et al., 2015] Linåker, J., Sulaman, S. M., Maiani de Mello, R., and Höst, M. (2015). Guidelines for conducting surveys in software engineering.

[Lindvall et al., 2002] Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams, L., and Zelkowitz, M. (2002). Empirical findings in agile methods. In *Conference on extreme programming and agile methods*, pages 197–207. Springer.

[Lindvall et al., 2004] Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J., and Kahkonen, T. (2004). Agile software development in large organizations. *Computer*, 37(12):26–34.

[Mahanti, 2006] Mahanti, A. (2006). Challenges in enterprise adoption of agile methods-a survey. *Journal of Computing and Information technology*, 14(3):197–206.

[McMunn and Manketo, 2017] McMunn, D. and Manketo, P. (2017). Building strong foundations... underwriting fannie mae's agile transformation. In *International Conference on Agile Software Development*. AgileAlliance.

[Misra et al., 2012] Misra, S., Kumar, V., Kumar, U., Fantazy, K., and Akhter, M. (2012). Agile software development practices: evolution, principles, and criticisms. *International Journal of Quality & Reliability Management*.

[Moe and Dingsøyr, 2017] Moe, N. B. and Dingsøyr, T. (2017). Emerging research themes and updated research agenda for large-scale agile development: a summary of the 5th international workshop at xp2017. In *Proceedings of the XP2017 Scientific Workshops*, pages 1–4. ACM.

[Moe et al., 2016] Moe, N. B., Olsson, H. H., and Dingsøyr, T. (2016). Trends in large-scale agile development: A summary of the 4th workshop at xp2016. In *Proceedings of the Scientific Workshop Proceedings of XP2016*, pages 1–4. ACM.

[One,] One, V. 15th State of Agile Survey. https://stateofagile.com/.

[Paasivaara, 2017] Paasivaara, M. (2017). Adopting safe to scale agile in a globally distributed organization. In *Proceedings of the 2017 IEEE 12th International Conference on Global Software Engineering*, pages 36–40. IEEE.

[Paasivaara et al., 2008] Paasivaara, M., Durasiewicz, S., and Lassenius, C. (2008). Using scrum in a globally distributed project: a case study. *Software Process: Improvement and Practice*, 13(6):527–544.

[Paasivaara and Lassenius, 2016] Paasivaara, M. and Lassenius, C. (2016). Scaling scrum in a large globally distributed organization: A case study. In *2016 IEEE 11th International Conference on Global Software Engineering*, pages 74–83. IEEE.

[Patil et al., 2016] Patil, V., Panicker, S., and Kv, M. (2016). Use of agile methodology for mobile applications. *Int. J. Latest Technol. Eng. Manag. Appl. Sci.*, 5(10):73–77.

[Patton, 2002] Patton, M. Q. (2002). Two decades of developments in qualitative inquiry: A personal, experiential perspective. *Qualitative social work*, 1(3):261–283.

[Petersen and Wohlin, 2011] Petersen, K. and Wohlin, C. (2011). Measuring the flow in lean software development. *Software: Practice and experience*, 41(9):975–996.

[Petri Kettunen and Männistö, 2020] Petri Kettunen, Maarit Laanti, F. F. T. M. and Männistö, T. (2020). Industrial agile transformations lacking business emphasis: Results from a nordic survey study. In *11th International Conference of Software Business*, pages 46–54. Springer, Cham.

[Phalnikar et al., 2009] Phalnikar, R., Deshpande, V., and Joshi, S. (2009). Applying agile principles for distributed software development. In *2009 International Conference on Advanced Computer Control*, pages 535–539. IEEE.

[post, 2017] post, M. A. (2017). Case Study Australian Post. http://www.scaledagileframework.com/wp-content/uploads/delightful-downloads/2018/11/Australia-Post-SAFe-Case-Study-v1.0.pdf. [Online; accessed 09-June-2018].

[Pries-Heje and Krohn, 2017] Pries-Heje, J. and Krohn, M. M. (2017). The safe way to the agile organization. In *Proceedings of the XP2017 Scientific Workshops*, page 18. ACM.

[Putta et al., 2018] Putta, A., Paasivaara, M., and Lassenius, C. (2018). Benefits and challenges of adopting the scaled agile framework (safe): Preliminary results from a multivocal literature review. In *Proceedings of the 19th International Conference on Product-Focused Software Process Improvement*, pages 334–351. Springer, Cham.

[Rafi et al., 2012] Rafi, D. M., Moses, K. R. K., Petersen, K., and Mäntylä, M. V. (2012). Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. In *Proceedings of the 7th International Workshop on Automation of Software Test*, pages 36–42. IEEE Press.

[Reifer et al., 2003] Reifer, D., Maurer, F., and Erdogmus, H. (2003). Scaling agile methods ieee software, july.

[Rejab et al., 2015] Rejab, M. M., Noble, J., and Marshall, S. (2015). Coordinating expertise outside agile teams. In *International Conference on Agile Software Development*, pages 141–153. Springer.

[Rolland, 2016] Rolland, K. H. (2016). Scaling across knowledge boundaries: A case study of a large-scale agile software development project. In *Proceedings of the Scientific Workshop Proceedings of XP2016*, pages 1–5.

[Rolland et al., 2016] Rolland, K. H., Fitzgerald, B., Dingsøyr, T., and Stol, K.-J. (2016). Problematizing agile in the large: alternative assumptions for large-scale agile development. In *Proceedings of the 37th International Conference on Information Systems*, pages 1–21. AIS.

[Royce, 1987] Royce, W. W. (1987). Managing the development of large software systems: concepts and techniques. In *Proceedings of the 9th international conference on Software Engineering*, pages 328–338.

[Salameh and Bass, 2019a] Salameh, A. and Bass, J. (2019a). Spotify tailoring for b2b product development. In *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 61–65. IEEE.

[Salameh and Bass, 2019b] Salameh, A. and Bass, J. M. (2019b). Spotify tailoring for promoting effectiveness in cross-functional autonomous squads. In *International Conference on Agile Software Development*, pages 20–28. Springer.

[Salikhov et al., 2020] Salikhov, D., Succi, G., and Tormasov, A. (2020). An empirical analysis of success factors in the adaption of the scaled agile framework–first outcomes from an empirical study. *arXiv preprint arXiv:2012.11144*.

[Scaled Agile Inc., a] Scaled Agile Inc. Agile Release Trains (ART's). https://www.scaledagileframework.com/agile-release-train.

[Scaled Agile Inc., b] Scaled Agile Inc. Identifying Value Streams and Agile Release Trains. https://www.scaledagileframework.com/identify-value-streams-and-arts/.

[Scaled Agile Inc., c] Scaled Agile Inc. Implementation Roadmap. https://www.scaledagileframework.com/implementation-roadmap/.

[Scaled Agile Inc., d] Scaled Agile Inc. SAFe Case Studies. https://bit.ly/2NGa2J8.

[Scaled Agile Inc., e] Scaled Agile Inc. SAFe Home Page. https://www.scaledagileframework.com/.

[Scaled Agile Inc., 2016] Scaled Agile Inc. (2016). Royal Philips Case Study. http://www.scaledagileframework.com/royal-phillips-case-study/.

[Scaled Agile Inc., 2017] Scaled Agile Inc. (2017). Thales Case Study. http://www.scaledagileframework.com/thales-case-study/.

[Scaled Agile Inc., 2018] Scaled Agile Inc. (2018). KLM Air France Case Study. https://www.scaledagileframework.com/case-study-air-france-klm/ .

[Schongot and Man, 2018] Schongot, J. and Man, Y. (2018). Lean engineering at elbit systems aerospace division - implementation journey. In *Annual Forum Proceedings - AHS International*, volume 2018-May.

[Schuppan and Rußwurm, 2000] Schuppan, V. and Rußwurm, W. (2000). A cmm-based evaluation of the v-model 97. In *European Workshop on Software Process Technology*, pages 69–83. Springer.

[Scrum Inc.,] Scrum Inc. Scrum at Scale. https://www.scrumatscale.com/ .

[Shankarmani et al., 2012] Shankarmani, R., Pawar, R., Mantha, S., and Babu, V. (2012). Agile methodology adoption: benefits and constraints. *International Journal of Computer Applications*, 58(15).

[Smite et al., 2019] Smite, D., Moe, N. B., Levinta, G., and Floryan, M. (2019). Spotify guilds: how to succeed with knowledge sharing in large-scale agile organizations. *IEEE Software*, 36(2):51–57.

[software, 2015] software, R. (2015). Travis Perkins Case Study. http://www.scaledagileframework.com/wp-content/uploads/delightful-downloads/2018/09/Travis-Perkins-Case-Study.pdf. [Online; accessed 09-June-2018].

[Solinski and Petersen, 2016] Solinski, A. and Petersen, K. (2016). Prioritizing agile benefits and limitations in relation to practice usage. *Software quality journal*, 24(2):447–482.

[Sreenivasan and Kothandaraman, 2019] Sreenivasan, S. and Kothandaraman, K. (2019). Improving processes by aligning capability maturity model integration and the scaled agile framework®. *Global Business and Organizational Excellence*, 38(6):42–51.

[Stadler et al., 2019] Stadler, M., Vallon, R., Pazderka, M., and Grechenig, T. (2019). Agile distributed software development in nine central european teams: Challenges, benefits, and recommendations. *International Journal of Computer Science & Information Technology (IJCSIT) Vol*, 11.

[Stebbins, 2001] Stebbins, R. A. (2001). *Exploratory research in the social sciences*, volume 48. Sage.

[Theobald et al., 2019] Theobald, S., Schmitt, A., and Diebold, P. (2019). Comparing scaling agile frameworks based on underlying practices. In *International Conference on Agile Software Development*, pages 88–96. Springer.

[Trienekens et al., 2018] Trienekens, J., Kusters, R., Himawan, H. B., and van Moll, J. (2018). Customer involvement in the scaled agile framework: results from a case study in an industrial company. In *20th International Conference on Enterprise Information Systems*, pages 104–110. SCITEPRESS-Science and Technology Publications, Lda.

[Turetken et al., 2017] Turetken, O., Stojanov, I., and Trienekens, J. J. (2017). Assessing the adoption level of scaled agile development: a maturity model for scaled agile framework. *Journal of Software: Evolution and Process*, 29.

[Turk et al., 2002] Turk, D., France, R., and Rumpe, B. (2002). Limitations of agile software processes. In *Proceedings of the Third International Conference on eXtreme Programming and Agile Processes in Software Engineering*, pages 43–46.

[Uludağ et al., 2018] Uludağ, Ö., Kleehaus, M., Caprano, C., and Matthes, F. (2018). Identifying and structuring challenges in large-scale agile development based on a structured literature review. In *2018 IEEE 22nd International Enterprise Distributed Object Computing Conference*, pages 191–197. IEEE.

[Uludağ et al., 2019] Uludağ, Ö., Kleehaus, M., Dreymann, N., Kabelin, C., and Matthes, F. (2019). Investigating the adoption and application of large-scale scrum at a german automobile manufacturer. In *Proceedings of the 2019 ACM/IEEE 14th International Conference on Global Software Engineering*, pages 22–29. IEEE.

[Uludağ et al., 2017] Uludağ, Ö., Kleehaus, M., Xu, X., and Matthes, F. (2017). Investigating the role of architects in scaling agile frameworks. In *2017 IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC)*, pages 123–132. IEEE.

[Uludag et al., 2020] Uludag, Ö., Philipp, P., Putta, A., Paasivaara, M., Lassenius, C., and Matthes, F. (2020). Revealing the state-of-the-art in large-scale agile development: A systematic mapping study. *arXiv preprint arXiv:2007.05578*.

[Uludağ et al., 2021] Uludağ, Ö., Putta, A., Paasivaara, M., and Matthes, F. (2021). Evolution of the agile scaling frameworks. In *International Conference on Agile Software Development*, pages 123–139. Springer, Cham.

[Version One, a] Version One. 13th State of Agile Survey. https://bit.ly/3sadydS. last accessed on 03-11-2021.

[Version One, b] Version One. State of Agile Survey. https://explore.versionone.com/state-of-agile/ versionone-12th-annual-state-of-agile-report.

[VersionOne, 2020] VersionOne (2020). 14th Annual State of Agile Survey. =https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494. last accessed on: 05-11-2021.

[Vijayasarathy and Turk, 2008] Vijayasarathy, L. and Turk, D. (2008). Agile software development: A survey of early adopters. *Journal of Information Technology Management*, 19(2):1–8.

[Wińska and Dąbrowski, 2020] Wińska, E. and Dąbrowski, W. (2020). Software development artifacts in large agile organizations: a comparison of scaling agile methods. In *Data-Centric Business and Applications*, pages 101–116. Springer.

[Wohlin et al., 2012] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering*. Springer Science & Business Media.

[Yael Man, 2017] Yael Man, I. O. (2017). Developing avionic products using lean-agile at elbit systems. In *International Conference on Agile Software Development*. AgileAlliance.

[Yin, 2009] Yin, R. K. (2009). How to do better case studies. *The SAGE handbook of applied social research methods*, 2:254–282.

[Yin, 2013] Yin, R. K. (2013). *Case study research: Design and methods*. Sage publications.

[Zhou et al., 2016] Zhou, X., Jin, Y., Zhang, H., Li, S., and Huang, X. (2016). A map of threats to validity of systematic literature reviews in software engineering. In *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*, pages 153–160. IEEE.

Aalto-DT 155/2022

Agile methods were originally developed for small and co-located teams. The popularity and success of agile methods in small teams led to growing interest in agile adoption across large organizations as well. The primary goal of the thesis is to investigate the adoption and usage of scaling frameworks in practice. The goal is divided into two parts: a) scaling frameworks usage and adoption and b) SAFe usage and adoption.

Our results for the first part show that majority of the frameworks were designed to improve agility, collaboration, coordination, and synchronization between agile teams. The most common reasons for their adoption were to scale more people and deal with existing challenges and pain points. The benefits of adopting these frameworks were categorized into business, product, organizational, and culture and the challenges were categorized into implementation, organizational, and scope. Our results for the second part show that reasons for SAFe adoption are related to organizational, business, and framework-specific. SAFe transformation activities typically map with the SAFe roadmap activities.

The most common benefits of SAFe adoption are improved transparency, collaboration and faster time to market. The most significant challenges of SAFe adoption are identifying value streams and forming ARTs, change resistance, and inculcating an agile mindset. More in-depth research on scaling frameworks is needed to establish the effectiveness of their usage in practice.

9 789526 409948

ISBN 978-952-64-0994-8 (printed)
ISBN 978-952-64-0995-5 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Computer Science
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
THESES