
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Uludağ, Ömer; Putta, Abheeshta; Paasivaara, Maria; Matthes, Florian

Evolution of the Agile Scaling Frameworks

Published in:

Agile Processes in Software Engineering and Extreme Programming - 22nd International Conference on Agile Software Development, XP 2021, Proceedings

DOI:

[10.1007/978-3-030-78098-2_8](https://doi.org/10.1007/978-3-030-78098-2_8)

Published: 01/01/2021

Document Version

Publisher's PDF, also known as Version of record

Published under the following license:

CC BY

Please cite the original version:

Uludağ, Ö., Putta, A., Paasivaara, M., & Matthes, F. (2021). Evolution of the Agile Scaling Frameworks. In P. Gregory, C. Lassenius, X. Wang, & P. Kruchten (Eds.), *Agile Processes in Software Engineering and Extreme Programming - 22nd International Conference on Agile Software Development, XP 2021, Proceedings* (pp. 123-139). (Lecture Notes in Business Information Processing; Vol. 419 LNBIP). Springer.
https://doi.org/10.1007/978-3-030-78098-2_8

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.



Evolution of the Agile Scaling Frameworks

Ömer Uludağ¹(✉), Abheeshta Putta², Maria Paasivaara^{2,3},
and Florian Matthes¹

¹ Technische Universität München, München, Germany

{oemer.uludag,matthes}@tum.de

² Aalto University, Espoo, Finland

{abheeshta.putta,maria.paasivaara}@aalto.fi

³ LUT University, Lappeenranta, Finland

maria.paasivaara@lut.fi

Abstract. Over the past decade, agile methods have become the favored choice for projects undertaken in rapidly changing environments. The success of agile methods in small, co-located projects has inspired companies to apply them in larger projects. Agile scaling frameworks, such as Large Scale Scrum and Scaled Agile Framework, have been invented by practitioners to scale agile to large projects and organizations. Given the importance of agile scaling frameworks, research on those frameworks is still limited. This paper presents our findings from an empirical survey answered by the methodologists of 15 agile scaling frameworks. We explored (i) framework evolution, (ii) main reasons behind their creation, (iii) benefits, and (iv) challenges of adopting these frameworks. The most common reasons behind creating the frameworks were improving the organization's agility and collaboration between agile teams. The most commonly claimed benefits included enabling frequent deliveries and enhancing employee satisfaction, motivation, and engagement. The most mentioned challenges were using frameworks as cooking recipes instead of focusing on changing people's culture and mindset.

Keywords: Agile scaling frameworks · Large-scale agile · Survey

1 Introduction

Ever since the creation of the Agile Manifesto in 2001, practitioners and academics have devoted a great deal of attention to agile software development methods [1]. Initially, they were designed for small, co-located, and self-organizing teams that develop software in close collaboration with business customers using short iterations [2]. Hence, agile methods have been primarily applied to projects within the so-called '*agile sweet spot*', i.e., small and co-located teams of less than 50 persons with easy access to the user and business

experts and that develop non-life-critical software [3]. Given the successful adoption of agile methods in small organizations and projects, also many large software organizations have begun to adopt these methods [4]. However, the adoption of agile methods outside the agile sweet spot poses significant challenges to organizations, such as coordination challenges in multi-team environments [5]. To resolve issues associated with the adoption of agile methods in large-scale organizations and projects, several agile scaling frameworks, such as Large Scale Scrum (LeSS)¹ and Scaled Agile Framework (SAFe)², have been created both by some custodians of existing agile methods and by others who have worked with companies to scale agile methods to their settings [4, 6, 7]. As large organizations face growing pressures and expectations to become more agile, and the agile scaling frameworks claim to provide off-the-shelf solutions to scaling, their adoption has rapidly increased in industry, as confirmed by the yearly non-scientific survey on the state of agile development conducted by VersionOne [6–8].

Not only is there a growing interest in adopting agile scaling frameworks from an industrial perspective [8], but there is also a growing academic interest to study the adoption of these frameworks [6]. A systematic mapping study by Uludağ et al. [6] uncovered the topic of agile scaling frameworks as a major research stream in the field of large-scale agile development, with a total of 16% of all published studies related to large-scale agile development. The existing literature on the scaling frameworks mainly investigates how individual frameworks are adopted based on case studies (cf. [9]) followed by a comparison of the frameworks based on their underlying characteristics based on literature reviews (cf. [10]). However, the existing literature on agile scaling frameworks disregards the following topics: (i) providing a comprehensive overview of agile scaling frameworks and their evolution, (ii) studying the reasons behind creating these frameworks, and (iii) investigating the benefits and (iv) challenges of adopting these frameworks. To address this research gap, we conducted a survey with the creators/methodologists of known agile scaling frameworks and aim to answer the following research questions (RQs):

- *RQ1: How did the agile scaling frameworks evolve over the years?*
- *RQ2: What are key reasons behind creating of agile scaling frameworks?*
- *RQ3: What are the claimed benefits of adopting agile scaling frameworks?*
- *RQ4: What are the claimed challenges of adopting of agile scaling frameworks?*

The remainder of this paper is structured as follows. In Sect. 2, we provide an overview of related work. In Sect. 3, we portray the research design of our paper. Section 4 presents the result of our survey. In Sect. 5, we discuss our main findings and limitations and conclude our study with a summary of our results and remarks on future research.

¹ <https://less.works/>, last accessed on: 03-10-2021.

² <https://www.scaledagileframework.com/>, last accessed on: 03-10-2021.

2 Background and Related Work

The successful adoption of agile methods in small teams ignited a new passion among firms to start using agile methods in large projects, even beyond software development, across the enterprise [11]. This phenomenon is often referred as ‘large-scale agile development’ [12]. In line with Dikert et al. [5], we understand the term ‘large-scale agile development’, as the application of agile methods in large multi-team settings consisting of 50 persons or more, or at least six teams.

Over the past two decades, software engineers and researchers have devoted a great deal of attention to agile software development [13]. Within few years, various agile methods appeared on the landscape, such as Extreme Programming and Scrum, to name a few [1]. Figure 1 presents the various agile methods, their interrelationships, and their evolutionary paths [13].

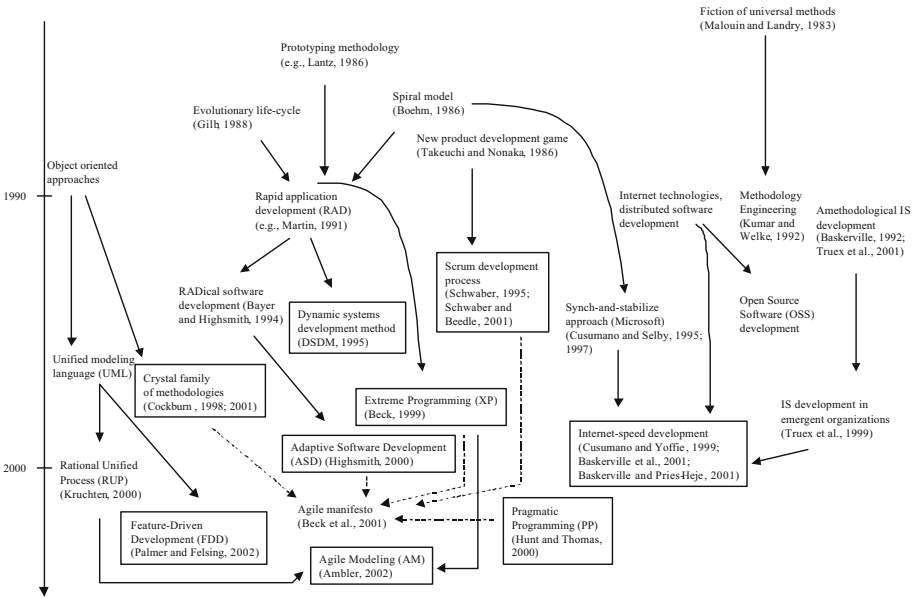


Fig. 1. Evolutionary map of agile methods [13]

Agile methods adhering to varying degrees to the tenets of the Agile Manifesto³ share some common characteristics, e.g., iterative and incremental development and focus on small releases [1]. The ideal context of applying agile methods in software projects lies within the so-called ‘agile sweet spot’, i.e., small and co-located teams of less than 50 persons with easy access to the user and business experts and that develop non-life-critical software [3]. However, applying agile methods both for larger projects or in larger companies [5], i.e., scaling agile

³ <http://agilemanifesto.org/>, last accessed on: 03-10-2021.

methods, involves two significant challenges. First, the scaling of agile methods entails additional scaling and complexity factors that summon ‘*bitter spot*’ conditions for agile methods, such as a large number of teams, geographical distribution, entrenched culture, or formal governance structures [14]. Second, present agile methods do not provide sufficient guidance on dealing with these scaling and complexity factors [15]. Thus, custodians of existing agile methods and consultants that have worked with companies in scaling agile to their settings have proposed several agile scaling frameworks over the last years to address the limitations of the agile methods in large organizations and projects [6, 7, 10]. These frameworks incorporate predefined workflow patterns to deal with issues related to large number of teams, inter-team coordination, and customer involvement [10, 16].

Due to the importance of this topic to companies, researchers have started to study the frameworks’ adoption [6]. Based on a structured literature review, Uludağ et al. [17] identified 20 different agile scaling frameworks presented in Table 1⁴. Secondary studies on the scaling frameworks compare some of them based on different criteria. For instance, Alqudah and Razali [10] juxtapose *DAD*, *LeSS*, *Nexus*, *RAGE*, *SAFe*, and *Spotify* based on, e.g., team size, available training and certificates, and the underlying agile methods and practices. Diebold et al. [18] provide a map visualizing underlying agile practices of different frameworks, such as *DAD*, *LeSS*, and *Nexus*, to support organizations in the selection of appropriate frameworks. Based on 13 agile transformation cases, Conboy and Carroll [16] provide nine challenges and a set of recommendations associated with agile scaling frameworks, such as *LeSS*, *Nexus*, *S@S*, and *Spotify*.

Although agile scaling frameworks have received some attention from academics [6], to the best of our knowledge, there is no other work that provides an overview of agile scaling frameworks, their evolution, and reasons, as well as the benefits and challenges of these frameworks.

3 Research Methodology

Survey Design. To answer the research questions, we created a survey following the guidelines suggested by Linåker et al. [19]. We opted to conduct a survey as it often aims to provide a state-of-the-art overview on particular methods [20], such as agile scaling frameworks. As a large part of our survey consists of closed-ended questions to quantitatively analyze the agile scaling frameworks, we used a survey as it is a suitable means to provide a quantitative description of the data [20]. The questionnaire consisted of four sections with a total of 22 questions⁵. The first section included questions on the framework background, e.g., reasons

⁴ We extended the table by Uludağ et al. [17] by adding a column to show the scaling levels of the frameworks and expanded the list of the frameworks by two additional frameworks: *HSD* and *Parallel* as their methodologists approached us during two agile conferences (see Sect. 3). We set the names of agile scaling frameworks whose methodologists participated in our survey in bold.

⁵ Questionnaire link: <https://bit.ly/2ZPI69S>.

Table 1. Overview of agile scaling frameworks based on [17]

| Framework | Methodologist | Organization | Publ. date | Category | Scaling level |
|--|---|----------------------------------|------------|-------------------|---------------|
| Dynamic Systems Development Method Agile Project Framework for Scrum (DSDM) | Arie van Bennekum | DSDM Consortium | 1997 | Framework | Portfolio |
| Crystal Family (Crystal) | Alistair Cockburn | Scrum Inc | 1998 | Set of methods | Team |
| Scrum of Scrums (SoS) | Jeff Sutherland; Ken Schwaber | LeSS Company B.V | 2001 | Mechanism | Program |
| Large Scale Scrum (LeSS) | Craig Larman; Bas Vodde | Adapt Inn | 2007 | Framework | Enterprise |
| Gill Framework (Gill) | Asif Qumer; Brian Henderson-Sellers | agile42 | 2008 | Framework | Enterprise |
| Enterprise Transition Framework (ETF) | — | Universo Online S.A | 2011 | Framework | Enterprise |
| Mega Framework (Mega) | Rafael Maranzato; Marden Neubert; Paula Heculano | — | 2011 | Framework | Portfolio |
| Scaled Agile Framework (SAFe) | Dean Leffingwell | Scaled Agile Inc | 2011 | Framework | Enterprise |
| Disciplined Agile Delivery (DAD) | Scott Ambler | Disciplined Agile Consortium | 2012 | Framework | Enterprise |
| Enterprise Agile Delivery and Agile Governance Practice (EADAGP) | Erik Marks | AgilePath | 2012 | Set of practices | Enterprise |
| Spotify Model (Spotify) | Henrik Kniberg; Anders Ivarsson; Joakim Sundén | Spotify | 2012 | Model | Enterprise |
| Recipes for Agile Governance in the Enterprise (RAGE) | Kevin Thompson | Cprime | 2013 | Framework | Portfolio |
| Continuous Agile Framework (CAF) | Andy Singleton | Maxos LLC | 2014 | Framework | Program |
| Enterprise Scrum (eScrum) | Mike Beedle ¹ | Enterprise Scrum Inc | 2014 | Framework | Enterprise |
| eXponential Simple Continuous Autonomous Learning Ecosystem (XSCALE) | Peter Merel | Xscale Alliance | 2014 | Set of principles | Enterprise |
| Holistic Software Development (HSD) | Mike MacDonagh; Steve Handy | Holistic Software Consulting Ltd | 2014 | Framework | Enterprise |
| ScALed Agile Lean Development (SALD) | Peter Beck; Markus Gärtner; Christoph Mathis; Stefan Rooock; Andreas Schliep | — | 2014 | Set of principles | Enterprise |
| FAST Agile (FAST) | Ron Quartel | Cron Technologies | 2015 | Set of methods | Program |
| Lean Enterprise Agile Framework (LEAF) | — | LeanPitch Technologies | 2015 | Framework | Enterprise |
| Nexus (Nexus) | Ken Schwaber | Scrum.org | 2015 | Framework | Program |
| Parallel Agile (Parallel) | Doug Rosenberg; Brarry Boehm; Matt Stephens; Charles Suscheck; Shobha Dhalipathi; Bo Wang | Parallel Agile Inc | 2016 | Set of methods | Enterprise |
| Scrum at Scale (S@S) | Jeff Sutherland; Alex Brown | Scrum Inc | 2018 | Framework | Enterprise |

behind the framework creation and the claimed benefits and challenges. The second section presented questions about framework evolution, e.g., the framework version history. In the third section, we aimed to capture the lean and agile foundations behind the framework, e.g., agile practices adopted to develop the framework. In the final section, we collected information on compatibility between the frameworks. The questions were compiled based on previous studies [8, 17] and the Ask Matrix⁶.

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*⁷. We used various approaches to reach out to the inventors or organizations, i.e., methodologists, that created the frameworks shown in Table 1. First, we sent out the questionnaire link to 22 methodologists by email. Second, we contacted some of the methodologists in two of the leading agile conferences: XP 2019⁸ and Agile 2019⁹, and emailed them the survey link. Third, we reached a few methodologists via LinkedIn¹⁰ by sending a personal message with the survey link. We received responses from 15 creators.

Data Analysis. We imported the survey data related to our four research questions to excel sheets. The first two authors analyzed data for all research questions individually by following Corbin and Strauss’s coding guidelines [21]. 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, both authors 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.

4 Results

4.1 RQ1: Evolution of the Agile Scaling Frameworks

Figure 2 shows a time-based overview 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*. Figure 2 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.

⁶ <http://www.agilescaling.org/ask-matrix.html>, last accessed on: 03-10-2021.

⁷ <https://www.unipark.com/en/>, last accessed on: 03-10-2021.

⁸ <https://www.agilealliance.org/xp2019/>, last accessed on: 03-10-2021.

⁹ <https://www.agilealliance.org/agile2019/>, last accessed on: 03-10-2021.

¹⁰ <https://www.linkedin.com/>, last accessed on: 03-10-2021.

According to our survey data, *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. However, it took the methodologists almost ten years to publish these frameworks, e.g., by publishing their official guides. Although nine frameworks were created before 2010, only three of them went public before 2010. Whereas, between 2011 and 2018, twelve frameworks were published. 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*, and *XSCALE*. *Gill* was initially created as the *ASSF* framework (2005-2008), which then evolved into *Gill* in 2012. The methodologists of *Mega* indicated that *Mega 1.0* was a derivative of *SoS*. They also stated that *Mega 2.0* was influenced by *Spotify* including the idea to extend the adoption of agile practices to other parts of the organization. The methodologists of *Spotify* found inspiration from Craig Larman's and Bass Vodde's two books (cf. [22, 23]), that later became *LeSS*. *Spotify* was also influenced by the Program Increment Planning events of *SAFe* (cf. [24]).

4.2 RQ2: Key Reasons Behind Creating Agile Scaling Frameworks

Table 2 presents 12 reasons behind creating scaling frameworks based on our survey. 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*.

Table 2. Reasons behind the creation of agile scaling frameworks

| Reason category | Reason | Reported in |
|---------------------|---|------------------------------------|
| <i>Complexity</i> | Dealing with increased complexity | ETF, SAFe |
| | Descaling large product organizations in smaller independent entities | eScrum, XSCALE |
| <i>Customer</i> | Delivering higher business value | LeSS |
| | Improving customer involvement | eScrum |
| <i>Market</i> | Improving the agility/adaptability of the organization | DAD, Gill, HSD, SAFe, S@S, Spotify |
| | Dealing with changing environments | LeSS |
| <i>Organization</i> | Improving the collaboration of agile teams working on same product | Nexus, Parallel, SAFe, S@S |
| | Improving the coordination of agile teams working on same product | Crystal, Nexus, S@S |
| | Improving the synchronization of agile teams working on same product | FAST, Nexus, SAFe |
| | Enabling the information/communication flow between agile teams | Crystal, Mega |
| | Scaling agile to more people/teams/higher organizational levels | LeSS, SAFe |
| | Managing dependencies between agile teams | eScrum |

4.3 RQ3: Benefits of Adopting Agile Scaling Frameworks

Table 3 presents 30 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.*

4.4 RQ4: Challenges of Adopting Agile Scaling Frameworks

Table 4 presents 22 challenges of adopting scaling frameworks based on our survey. 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. Claimed challenges of adopting agile scaling frameworks

| Challenge category | Challenge | Reported in |
|-----------------------------|--|-------------------------|
| <i>Implementation</i> | Implementing is difficult due to framework complexity | eScrum, HSD |
| | Missing familiarization with framework | eScrum, Nexus |
| | Implementation overhead | SAFe |
| | Misconception due to unconventional agile practices | FAST |
| | Insufficient guidance | Crystal |
| | Insufficient guidance regarding lean practices | Mega |
| | Using frameworks as cooking recipes | DAD, HSD, SAFe, Spotify |
| | Using frameworks without understanding for what reasons they should be applied | ETF, Nexus, Spotify |
| | Lack of management buy-in | SAFe, S@S |
| | Moving away from agile | Parallel, SAFe |
| <i>Organization/Culture</i> | Moving back from agile to traditional management approaches | ETF, LeSS |
| | Change resistance | LeSS |
| | Implementation is difficult in command and control-style organizations | FAST |
| | Implementation is difficult in traditional organizations | FAST |
| | Involving non-development units is difficult | Gill |
| | Implementation is difficult due to remaining power structures | LeSS |
| | Changing the mindset of the organization is difficult | ETF |
| | Implementation is limited to team level | Mega |
| | Implementation is not suitable for monolithic applications | Mega |
| | Insufficient guidance regarding product backlog management | Mega |
| <i>Scope</i> | Insufficient guidance regarding managers and specialist positions | LeSS |
| | Requiring co-location of agile teams | FAST |

5 Discussion and Conclusions

5.1 Key Findings

RQ1: How did agile scaling frameworks evolve over the years?

By comparing the evolution map of agile scaling frameworks in Fig. 2 with the evolutionary map of agile methods by Abrahamsson et al. [13], we observed two notable parallels. First, similar to the movement of agile methods, the movement to agile scaling frameworks emerged from parallel innovation both by some inventors of existing agile methods and by consultants who supported organizations in scaling the agile methods. Second, likewise to agile methods, agile scaling frameworks have been continuously emerging and evolving after the movement started. This trend will likely continue as the methodologists of agile scaling frameworks seem to be committed to improving them in the future. Although the evolution map visualizes several agile scaling frameworks, users have concentrated on a few frameworks [25], particularly on *SAFe* and *SoS* [8]. The most recent State of Agile survey [8] confirms this by stating that 35% of their respondents adopted *SAFe* and 16% used *SoS*. A similar observation can be made for agile methods, as 58% of the respondents of the State of Agile survey use Scrum, making it the most commonly used agile method [8].

RQ2: What are key reasons behind creating of agile scaling frameworks?

In total, we found 12 reasons behind the creation of 15 agile scaling frameworks. The reasons identified in our survey fall into either the category of improving the current state of the organization or dealing with the organization's prevalent challenges. Both look similar to reasons that trigger an organizational change [26]. 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 [27, 28]. Other reasons related to the scaling of agile methods, such as *dealing with increased complexity and scaling agile to more people*, were also reported in [9, 29–31]. 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 extant literature on agile development. Surprisingly, several popular reasons for agile, e.g., improving productivity, improving visibility, and improving predictability, were not reported as reasons [8]. As the questionnaire's question was about the main reasons of creating a framework, these earlier mentioned reasons can be some of the implicit reasons behind the creation of the 15 agile scaling frameworks.

RQ3: What are the claimed benefits of adopting agile scaling frameworks?

In total, we identified 30 claimed benefits. The majority of these claimed benefits were similar to the benefits of agile adoption in general found from recent studies on agile method, e.g., State of Agile survey [8]. However, the most common benefit of agile, namely improved productivity [8], 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.

RQ4: What are the challenges of adopting agile scaling frameworks?

We identified 22 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. The majority of the challenges found in our study, e.g., *change resistance*, *moving away from agile, implementation is difficult due to remaining power structures*, and *lack of management buy-in*, were already reported in previously published studies on scaling frameworks [16, 32–34] and large-scale agile development [5, 35]. The 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 [33]. 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.

5.2 Limitations

We discuss the limitations of our study through the threats, as suggested by Wohlin et al. [36].

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* [8].

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.

5.3 Conclusions

Large-scale agile development has received significant interest by practitioners and academics over the last years [37]. As organizations are driven by pressures to scale and to react fast, agile scaling frameworks are increasingly prevalent in contemporary software organizations [7, 8], sparking a growing academic interest in studying the adoption of these frameworks [6]. Although there is a body of knowledge on agile scaling frameworks, less research has been conducted to provide an overview of these frameworks and their evolution, study the reasons behind creating these frameworks, and investigate the benefits and challenges of adopting these frameworks. We surveyed the methodologists behind the agile scaling frameworks to address this research gap.

Our study provides an overview of 22 agile scaling frameworks of which 15 were covered by our survey. Our study extends extant literature by providing a map on agile scaling frameworks with their evolutionary paths. Although many methodologists started creating their first frameworks between 2001 and 2011, most guides on these frameworks were published later on. Our findings show a cluster of framework publications between 2011 and 2018, confirming the rising industry interest in scaling the agile methods. We identified 12 reasons behind the creation of the agile scaling frameworks. We revealed 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*. Further, the methodologists claimed 30 different benefits of adopting their frameworks related to 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*.

We encourage researchers to investigate further how contextual factors, such as complexity, multi-product development, or agile maturity, impact a scaling framework's selection. We call for cross-case analyses to compare the adoption of agile scaling frameworks based on common comparison characteristics.

References

1. Dingsøy, T., Nerur, S., Balijepally, V., Moe, N.B.: A decade of agile methodologies: Towards explaining agile software development. *J. Syst. Softw.* **85**(6), 1213–1221 (2012). Special Issue: Agile Development

2. Kettunen, P.: Extending software project agility with new product development enterprise agility. *Softw. Process Improv. Practice* **12**(6), 541–548 (2007)
3. Boehm, B.: Get ready for agile methods, with care. *Computer* **35**(1), 64–69 (2002)
4. Dingsøy, T., Moe, N.B., Fægri, T.E., Seim, E.A.: Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empir. Softw. Eng.* **23**(1), 490–520 (2017). <https://doi.org/10.1007/s10664-017-9524-2>
5. Dikert, K., Paasivaara, M., Lassenius, C.: Challenges and success factors for large-scale agile transformations: a systematic literature review. *J. Syst. Softw.* **119**, 87–108 (2016)
6. Uludag, Ö., Philipp, P., Putta, A., Paasivaara, M., Lassenius, C., Matthes, F.: Revealing the state-of-the-art in large-scale agile development: A systematic mapping study. arXiv preprint [arXiv:2007.05578](https://arxiv.org/abs/2007.05578) (2021)
7. Carroll, N., Conboy, K.: Applying normalization process theory to explain large-scale agile transformations. In: *Proceedings of the 14th International Research Workshop on IT Project Management*, January 2019
8. VersionOne: 14th Annual State of Agile Survey (2020). <https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494>. Accessed 03 Oct 2021
9. Pries-Heje, J., Krohn, M.M.: The safe way to the agile organization. In: *Proceedings of the XP2017 Scientific Workshops*, pp. 1–4. ACM, May 2017
10. Alqudah, M., Razali, R.: A review of scaling agile methods in large software development. *Int. J. Adv. Sci. Eng. Inf. Technol.* **6**(6), 828–837 (2016)
11. Paasivaara, M., Behm, B., Lassenius, C., Hallikainen, M.: Large-scale agile transformation at ericsson: a case study. *Empir. Softw. Eng.* **23**(5), 2550–2596 (2018)
12. Dingsøy, T., Fægri, T.E., Itkonen, J.: What is large in large-scale? *A Taxonomy of Scale for Agile Software Development*. In: Jedlitschka, A., Kuvaja, P., Kuhrmann, M., Männistö, T., Münch, J., Raatikainen, M. (eds.) *PROFES 2014*. LNCS, vol. 8892, pp. 273–276. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13835-0_20
13. Abrahamsson, P., Warsta, J., Siponen, M.T., Ronkainen, J.: New directions on agile methods: a comparative analysis. In: *Proceedings of the 25th International Conference on Software Engineering*, pp. 244–254. IEEE, May 2003
14. Ambler, S.W.: agile software development at scale. In: Meyer, B., Nawrocki, J.R., Walter, B. (eds.) *CEE-SET 2007*. LNCS, vol. 5082, pp. 1–12. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85279-7_1
15. Maples, C.: Enterprise agile transformation: the two-year wall. In: *Proceedings of the 2009 Agile Conference*, pp. 90–95. IEEE, August 2009
16. Conboy, K., Carroll, N.: Implementing large-scale agile frameworks: challenges and recommendations. *IEEE Softw.* **36**(2), 44–50 (2019)
17. Uludağ, Ö., Kleehaus, M., Xu, X., Matthes, F.: Investigating the role of architects in scaling agile frameworks. In: *Proceedings of the 21st International Enterprise Distributed Object Computing Conference*, IEEE, pp. 123–132, October 2017
18. Diebold, P., Schmitt, A., Theobald, S.: Scaling agile: how to select the most appropriate framework. In: *Proceedings of the 19th International Conference on Agile Software Development: Companion*, pp. 1–4. ACM, May 2018
19. Linåker, J., Sulaman, S.M., Maiani de Mello, R., Höst, M.: Guidelines for conducting surveys in software engineering (2015)
20. Punter, T., Ciolkowski, M., Freimut, B., John, I.: Conducting on-line surveys in software engineering. In: *International Symposium on Empirical Software Engineering*, pp. 80–88. IEEE (2003)

21. Corbin, J.M., Strauss, A.L.: *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 3rd edn. Sage Publications Inc., Los Angeles, Calif (2008)
22. Larman, C.: *Scaling lean & agile development: thinking and organizational tools for large-scale Scrum*. Pearson Education India (2008)
23. Larman, C., Vodde, B.: *Practices for scaling lean & Agile development: large, multi-site, and offshore product development with large-scale scrum*. Pearson Education (2010)
24. Scaled Agile Inc.: *PI Planning* (2021). <https://www.scaledagileframework.com/pi-planning/>. Accessed 03 Oct 2021
25. Putta, A., Paasivaara, M., Lassenius, C.: Benefits and challenges of adopting the scaled agile framework (SAFe): preliminary results from a multivocal literature review. In: Kuhrmann, M., et al. (eds.) PROFES 2018. LNCS, vol. 11271, pp. 334–351. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03673-7_24
26. Scaled Agile Inc.: *Reasons for SAFe Adoption* (2021). <https://www.scaledagileframework.com/reaching-the-tipping-point/>. Accessed 03 Oct 2021
27. Paasivaara, M.: Adopting safe to scale agile in a globally distributed organization. In: *Proceedings of the 12th International Conference on Global Software Engineering*, pp. 36–40. IEEE, May 2017
28. Gustavsson, T.: Dynamics of inter-team coordination routines in large-scale agile software development. In: *Proceedings of the 27th European Conference on Information Systems*, pp. 1–6, June 2019
29. Heikkilä, V.T., Paasivaara, M., Rautiainen, K., Lassenius, C., Toivola, T., Järvinen, J.: Operational release planning in large-scale scrum with multiple stakeholders—a longitudinal case study at f-secure corporation. *Inf. Softw. Technol.* **57**, 116–140 (2015)
30. McMunn, D., Manketo, P.: Building strong foundations... underwriting fannie mae’s agile transformation. In: *International Conference on Agile Software Development*, Agile Alliance, August 2017
31. Michelson, C., Adolph, S.: Bias from the bottom: A different way to bootup a safe train. In: *International Conference on Agile Software Development*, Agile Alliance (2019)
32. Putta, A., Paasivaara, M., Lassenius, C.: Benefits and Challenges of Adopting the Scaled Agile Framework (SAFe): preliminary results from a multivocal literature review. In: Kuhrmann, M., et al. (eds.) PROFES 2018. LNCS, vol. 11271, pp. 334–351. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03673-7_24
33. Putta, A., Paasivaara, M., Lassenius, C.: How are agile release trains formed in practice? a case study in a large financial corporation. In: Kruchten, P., Fraser, S., Coallier, F. (eds.) XP 2019. LNBIP, vol. 355, pp. 154–170. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19034-7_10
34. Kalenda, M., Hyna, P., Rossi, B.: Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software: Evolution and Process* **30**, (2018)
35. Uludağ, Ö., Kleehaus, M., Dreyman, N., Kabelin, C., Matthes, F.: Investigating the adoption and application of large-scale scrum at a German automobile manufacturer. In: *Proceedings of the 14th International Conference on Global Software Engineering*, pp. 22–29. IEEE, May 2019

36. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer, Cham (2012)
37. Dingsøy, T., Falessi, D., Power, K.: Agile development at scale: the next frontier. *IEEE Softw.* **36**(2), 30–38 (2019)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

