Comparing Scaling Agile Frameworks Based on Underlying Practices

Sven Theobald[1(✉)], Anna Schmitt[1], and Philipp Diebold[2]

[1] Fraunhofer IESE, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
{sven.theobald, Anna.schmitt}@iese.fraunhofer.de
[2] Bagilstein GmbH, Mainz, Germany
philipp.diebold@bagilstein.de

Abstract. *Context*: Agile software development is widely-used by small teams and has benefits like increased transparency or faster feedback. However, companies want to benefit from Agile also in the development of big products, where multiple teams are involved. Many Scaling Agile Frameworks exist, but only few can be found in industry, especially SAFe, LeSS, and Nexus. *Objective*: The aim of this work is to identify commonalities of existing Scaling Agile Frameworks concerning their practices. *Method*: We extracted and consolidated the practices of twelve frameworks and compared the frameworks based on their practices using a visualization. *Results*: Frameworks prescribe scaling practices as well as practices on team level. There are practices common to most frameworks like the scaled Scrum events, e.g., a scaled planning meeting or retrospective. *Conclusion*: Practitioners are enabled to make informed decisions when choosing or tailoring their individual Scaling Agile Framework.

Keywords: Agile development · Scaling agile · Scaling frameworks · Scaling practices · Framework comparison · Subway Map

1 Introduction

The rising popularity of agile software development is based on many benefits like managing changing priorities, increasing time to market or team moral [1]. Agile is composed of values, principles, and methods. Scrum [2] is the most used agile method across all organization types and sizes [1]. All these methods base on different Agile Practices, like Daily Stand-Up or Sprint [2]. However, Scrum and all other Agile Methods are not sufficient to achieve the desired benefits of all kinds of organizations regarding agile development. Especially for big projects or organizations, Agile Methods are not sufficient, since they were designed for small teams only. However, organizations with big teams also want to develop Agile. Therefore, several so-called Scaling Agile Frameworks increasingly came up in the last six years. The most famous ones according to [1] are the Scaled Agile Framework (SAFe) [3] and Scrum-of-Scrums [4]. For those commonly used frameworks, some experience reports and studies exist, especially for SAFe [4–6]. However, also less known ones like FAST Agile Scaled Technology (FAST) [7] or Recipes for Agile Governance (RAGE) [8]

R. Hoda (Ed.): XP 2019 Workshops, LNBIP 364, pp. 88–96, 2019.
https://doi.org/10.1007/978-3-030-30126-2_11

exist. Scaling frameworks are based on practices on the technological and managerial level. These practices form the foundation for the implementation of all frameworks. Only [9] conducted a comparison on practice level so far and identified eight common scaling practices by comparing LeSS and SAFe. If Scaling Agile Frameworks were compared directly in related work, the comparison was along characteristics of the frameworks [10–12]. [13] compared eight Scaling Agile Frameworks on how IT governance is covered. In this work, we aim to identify the commonalities of Scaling Agile Frameworks concerning their defined practices. We used the twelve Scaling Agile Frameworks from [12] and updated the visualization [12]. To be able to conduct a comparison on practice level, we first extracted and consolidated all practices from these twelve Scaling Agile Frameworks.

2 Overview Over Practices

We went through the descriptions of each practice given by the frameworks. Based on these descriptions, we divided the practices into three groups: (1) practices that are only used on team level (cf. Table 1 that only displays the Scrum practices), (2) practices that are only used to scale agile (c.f. Table 2), and (3) practices that can be used for both – scaling agile and on team level (c.f. Table 3). Based on this classification, we created three different tables that provide an overview of the categories, subcategories, and related practices. Scaling Agile frameworks do not only define scaling practices, but also demand practices on team level. These coordination mechanisms for each team help to better align multiple teams. Table 1 only shows the Scrum practices, since they also appear in the Subway Map. Scrum is the most commonly used method [1]. It describes the management practices without prescribing technical practices [14]. Most scaling frameworks base on Scrum on team level.

Table 1. Scrum practices used on team level

Categories	*Subcategories*	*Practices*
Meeting types	Daily Stand-Up	Daily Scrum, Daily Stand-Up, Weekly Scrum, Stand-Up Meeting, Daily Coordination Meeting
Planning Meeting	Sprint Planning	Iteration Planning, Sprint Planning Part 1, Sprint Planning and Investigation, Phase Planning, Sprint Planning, Planning Session, FAST Meeting - Part 2: Marketplace in Open Space style, Kick-Off
Backlog Preparation	Product Backlog	Backlog, Product Backlog, Tribe Product Backlog, Team Backlog
	Sprint Backlog	Sprint Backlog, Iteration Backlog
	Backlog Refinement	Backlog Grooming, Product Backlog Refinement, Backlog Decomposition, Backlog Prioritization, PBI Inspection (in Sprint), Look-ahead Planning

(*continued*)

Table 1. (*continued*)

Categories	*Subcategories*	*Practices*
Iterative Procedure	*Sprint*	Sprint, Synchronous Sprints, Iteration
Lessons Learned	*Retrospective*	Retrospective, Sprint Retrospective, Iteration Retrospective, Team Retrospective
Review/Demo	*Review/Demo*	Sprint Review, Sprint Review Record, Iteration Review, Production Readiness Review, Light-Weight Milestone Review, FAST-Meeting - Part 1: Review (show and tell), Project Review
Progress Activities	*Definition of Done*	Definition of Done

On a scaled level (cf. Table 2), many practices on team level are adapted on a scaled level. Team level practices like the Scrum events were adapted for a scaled environment, e.g. by changing the participants of the events. Many frameworks also demand team level mechanisms, such as a Kanban board, Burn Charts or Release Planning activities, to be used in scaled projects. In addition, dedicated scaling practices like the Architecture Release Train from SAFe help to align the work of teams.

Table 2. Scaling practices

Categories	*Subcategories*	*Practices*
Meeting Types	*Scrum-of-Scrums*	Scrum-of-Scrums-Meeting, Scrum-of-Scrums, Nexus Daily Scrum, Cross-Team Coordination, Inter-Team Coordination Meeting
	Product Owner Sync	Product Owner Sync
Planning Meeting	*Scaled Planning*	Program Increment Planning, Sprint Planning Part 2, Nexus Sprint Planning, Portfolio Planning Meeting, Multisite Sprint Planning Part 1
	Scaled (Sprint) Goal	FAST Meeting - Part III: Announcements and Alignment of Vision, Nexus Sprint Goal, Program Increment Objective, Terms of Reference, Agile Charter
Backlog Preparation	*Scaled Backlog*	Program Backlog, Sync Backlog, Portfolio Backlog, Nexus Sprint Backlog
	Scaled Backlog Refinement	Joint Light Product Backlog Refinement, Multisite Product Backlog Refinement, Portfolio Grooming Meeting

(*continued*)

Table 2. (*continued*)

Categories	*Subcategories*	*Practices*
Manage Impediments	*Scaling Impediments Management*	Impediments (Backlog)
Delivery	*Agile Release Train*	Agile Release Train, Release Train
Architecture	*Architectural Runway*	Architectural Runway
Open Source Data	*Collective Ownership*	Collective Ownership
Release Activities	*Release Planning*	Release Planning, Release Management, Release Planning Meeting
	Release Handoffs	Release Handoffs
	Release Review	Release Review
Lessons Learned	*Scaled Retrospective*	Joint Retrospective, Nexus Sprint Retrospective, Inspect & Adapt Workshop
Review/Demo	*Review/Demo*	Quality Assessment
	Scaled Review	Overall Sprint Review, Multisite Sprint Review, Staging Readiness Review, Nexus Sprint Review, System Demo
Progress Activities	*Portfolio/Program Kanban Board*	Portfolio Kanban, Program Kanban
Others		*Initiative Assessment, Flex-Teaming, Beta Codex, Automated Metrics*

With Table 3, we show that there are also practices that are demanded on team level, but are also demanded under scaling conditions. This does not necessarily mean that the same framework demands a practice in both environments; it could also be that one framework uses the practice on team level, whereas another framework uses the practice as a scaling mechanism. General concepts like Time Boxing, Estimation or Open Source can be used by a single team as well as by multiple teams. User Stories help to describe the functionality of a product, independent of how many teams are responsible for this product. Communities of Practice are independent from projects. There are also practices that gain importance in a scaled environment, like Architecture or Release Activities. A focus on such topics is essential due to the increased coordination effort of multiple teams and the complexity of larger products. Likewise, Strategic Activities that can also already be applied on team level, support alignment of teams and reduce risk related to larger complex products.

Table 3. Practices for both scaled and team level

Categories	*Subcategories*	*Practices*
Meeting Types	*Timeboxing*	Timeboxing
Planning Meeting	*Prioritization*	Prioritization Meeting, MoSCoW, Prioritized Requirements List
Backlog Preparation	*Transition Backlog*	Evaluation Backlog, Transition Backlog, Practice Backlog
	Release Map	Release Map
Manage Impediments	*Manage Impediments*	Impediment Removal, Impediment Backlog, Continuous Impediment Removal
Requirements Documentation		User Stories, Portfolio Epic, Epic, Story Document, Requirement Document
Community of Practice		Community of Practice
Iterative Procedure	*Increment*	Increment of Change, Integrated Increment, Evolutionary Development, Pre-and Post-Program Increment
Architecture	*Architectural Envisioning*	Architectural Envisioning
Open Source Data	*Internal Open Source*	Internal Open Source
Release Activities	*Delivery/Release Plan*	Delivery/Release Plan
Strategic Activities		Decision making Framework, Lean-Agile Budgeting, Value Stream, Roadmap, Strategic Themes, Business Case, Decision Matrix, Funding Decision, Project Map
Estimation		Estimation, Forecasting
Others		*Benefits Assessment*

3 Comparison of Frameworks

We extended our "Subway Map" inspired visualization (similar to [15]) from [12] to show (1) which framework contains which practices as well as (2) which common practices are shared by multiple frameworks (cf. Figure 1). In the Subway Map (cf. Figure 1), each line represents a Scaling Agile Framework. The single subway stations illustrate the single practices that appear in those Scaling Agile Frameworks. We wanted the comparison to be easy to understand and visible at a glance. For the sake of simplicity, some subway stations represent only categories instead of single practices. The big stations symbolize practices that are used by many frameworks, e.g. Daily Stand-Up or Product Backlog.

The Subway Map shows that some frameworks share common Scaling Practices like the scaled form of the Scrum practices, namely: Scaled (Sprint) Goal, Scaled Retrospective, Scaled Planning, Scaled Review, Scrum of Scrums, and Scaled Backlog. Whereas, some more individual practices only occur in few frameworks, such as,

Release Review, Program-/Portfolio Kanban Board, Agile Release Train, Beta Codex, and Architectural Runway. On a closer inspection, it can be seen that most of the widespread practices are based on Scrum. This can be explained by the fact that Scrum contains management practices that mainly serve to organize the process around the software development in a lightweight manner.

Table 4. Practices and their occurrence over frameworks

#	Practice	#	Practice	#	Practice
11	Sprint Planning	4	Increment	1	Manage Impediments
11	Sprint	4	Scaled Review	1	Scaling Impediments Management
10	Retrospective	4	Strategic Activities	1	Architectural Runway
10	Review/"Demo"	4	Estimation	1	Architecture Envisioning
9	Daily Stand-Up	3	Agile Release Train	1	Internal Open Source
8	Product Backlog	3	Release Planning	1	Delivery/Release Plan
7	Definition of Done	3	Scaled Retrospective	1	Release Handoffs
6	Scrum of Scrums	2	Prioritization	1	Product Deploy Validation
6	Sprint Backlog	2	Transition Backlog	1	Release Review
6	Backlog Refinement	2	Scaled Backlog Refinement	1	Beta Codex
5	Scaled Planning	2	Collective Ownership	1	Facilitated Workshop
5	Scaled (Sprint) Goal	2	Portfolio/Program Kanban Board	1	Flex-Teaming
5	Requirements Documentation	1	Product Owner Sync	1	Initiative Assessment
4	Scaled Backlog	1	Timeboxing	1	Benefits Assessment
4	Community of Practice	1	Release Map		

Technical practices like Pair Programming are rather seldom part of scaling frameworks, since they often do not scale beyond software development on team level. Furthermore, it can be seen that all Scaling Agile Frameworks include scaling practices, but also non-scaling practices, namely practices on team level. Table 4 lists the practices across the frameworks ordered by occurrence. With the help of Table 4 and our visualization, it also can be seen that the Scrum practices, which are only used on team level, are still applied by almost every framework. This obvious commonality across the frameworks was the reason to include the Scrum practices in the visualization, though they are team level practices. Sprints and sprint planning are the practices recommended by almost all frameworks.

4 Implications

If practitioners have to decide on a suitable Scaling Agile Framework, they first need to know what frameworks exist. With the list of frameworks that are considered in our comparison, practitioners understand that there are more possibilities than the well-known frameworks that are typically presented by consultants. To identify the most suitable framework, we already presented a comparison of those frameworks in previous work [12] that compares criteria like the purpose, advantages or context of those frameworks. With an initial selection of a suitable framework, it can then be extended or adapted to the specific needs, leading to an individual approach.

When designing a scaling approach, practitioners might want to check for coverage of the suggested categories of practices (cf. Tables 1, 2 and 3). The practices from those categories might be important to consider since the authors of those scaling frameworks considered them. The Subway Map provides an overview over the existing Scaling Agile Frameworks, and shows the corresponding scaling practices. It can be seen that some frameworks are rather similar, sharing many practices, while others are rather individual and provide many practices that are not covered in other frameworks. We recommend considering the most commonly used practices (cf. Table 4) first, in order to implement the best practices of multiple frameworks. In addition, the individual practices can be evaluated to complement the base framework. Our comparison of frameworks shows that many frameworks are based on Scrum on team level. Practitioners that want to scale up their agile development should first consider their implementation of team level practices that support scaled agile development.

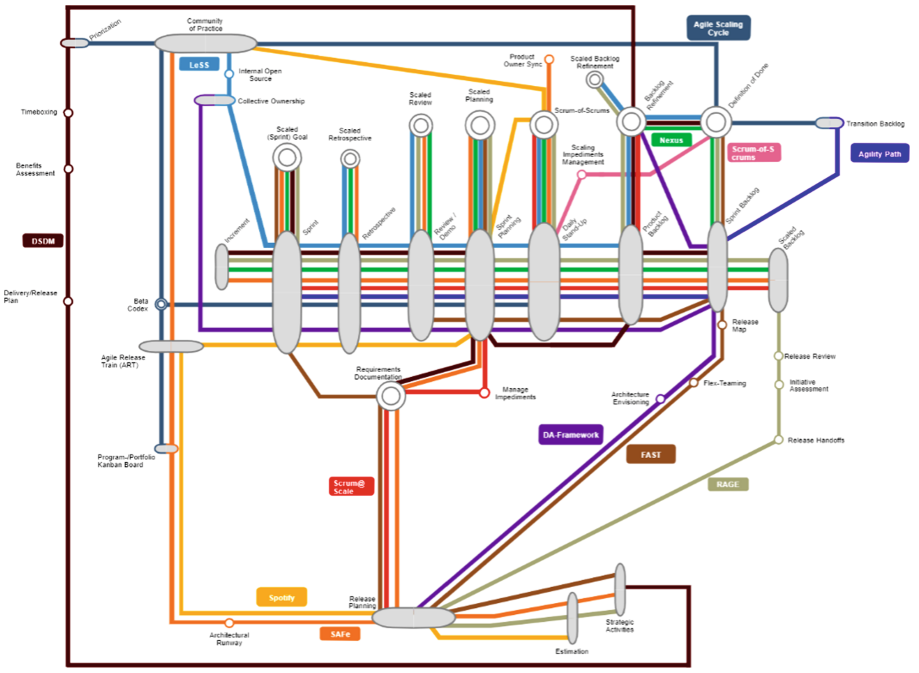

Fig. 1. Subway map visualizing practices of Scaling Agile Frameworks

The categorization of scaling practices and the Subway map need to be validated by the respective framework experts. Due to lack of documentation, there is the risk that wrong categorizations were made or practices from frameworks are missing. Since we did not conduct a systematic literature review, it might be that some frameworks or some of their practices are missing. For the sake of simplicity of the categorization, sometimes practices were clustered without considering the detailed differences. The stations of the Subway map have different abstraction levels, since some stations are based on practices, others on categories.

5 Conclusion

Due to the need to adapt Agile beyond the context Agile methods were initially designed for, many frameworks to scale agile have been developed in recent years. In order to understand similarities between the frameworks, we extracted a list of their underlying practices. A visualization provides a high-level overview over Scaling Agile Frameworks and enables comparison of the frameworks concerning the use of their underlying practices. Additionally, practices common to many frameworks are identified. We discuss how the results help practitioners to build their individual scaling framework. Feedback from framework authors is needed before proceeding with an in-depth analysis and comparison of the similarities and differences of the considered frameworks.

References

1. Version one: 12th annual state of agile TM report (2018). https://www.versionone.com/
2. Sutherland, J., Schwaber, K.: The scrum guide (2016). http://www.scrumguides.org
3. Scaled agile framework (2011). http://www.scaledagileframework.com/. Accessed 20 Sept 2018
4. Sutherland, J.: Scrum-of-scrums (1996). guide.agilealliance.org/guide/scrumofscrums.html
5. Laanti, M., Kettunen, P.: Finnish SAFe adoptions: a survey study. In: LargeScaleAgile@XP 2019, Montreal, Canada, 25 May 2019 (2019)
6. Putta, A., Paasivaara, M., Lassenius, C.: How are agile release trains formed in practice? A case study in a large financial corporation. In: XP 2019, 25 May 2019, Montreal, Canada (2019)
7. Quartel, R.: FAST agile scaled technology (FAST) (2015). http://www.fast-agile.com/method
8. Thompson, K.: Recipes for agile governance (RAGE) (2013). https://www.cprime.com/rage-services/
9. Kalenda, M., Hyna, P., Rossi, B.: Scaling agile in large organizations: practices, challenges, and success factors. J. Softw.: Evol. Proc. **30**(10), e1954 (2018)
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. Ebert, C., Paasivaara, M.: Scaling agile. IEEE Softw. **34**(6), 98–103 (2017)
12. Diebold, P., Schmitt, A., Theobald, S.: Scaling Agile – how to select the most appropriate framework. In: LargeScaleAgile@XP 2018, 21 May 2018, Porto, Portugal (2018)

13. Horlach, B., Böhmann, T., Schirmer, I., Drews, P.: IT governance in scaling agile frameworks. In: Proceedings of the Multikonferenz Wirtschaftsinformatik, Lüneburg (2018)
14. Diebold, P., Zehler, T.: The right degree of agility in rich processes. Managing Software Process Evolution, pp. 15–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31545-4_2
15. Agile Alliance: subway map to agile practices. https://www.agilealliance.org/agile101/subway-map-to-agile-practices/