



Comparison of Agile Scaling Frameworks

Robin J. Yeman
Colorado State University
robinyeman@gmail.com

Yashwant K. Malaiya
Colorado State University
prof.malaiya@gmail.com

ABSTRACT

21st century software development approaches, such as Agile has benefitted small initiatives with a single team building software in their ability to respond to change, reduce product delivery schedules, reduce product cost, increase product quality, and increase employee morale. The industry has begun to question what the benefits and challenges could be if we use those same practices at scale for large Initiatives with multiple teams. There have been multiple studies on the benefits of Agile at Scale including quality, productivity, and shorter time to market. Over the last two decades dozens of Scaling frameworks have been created. However, given the large number of frameworks available, how do companies choose the right scaling framework? In this paper we review what the difference between the frameworks are by comparing their core principles. This paper began by analyzing results Digital.ai's 16th Annual state of Agile Survey which identifies the 10 most utilized frameworks according to their respondents. The paper presents a detailed analysis of the principles to assess the degree of differences among the frameworks and determine based on this which framework is the best for large scale organizations to choose.

CCS CONCEPTS •

KEYWORDS

Agile, Scale, Framework

ACM Reference Format:

Robin J. Yeman and Yashwant K. Malaiya. 2023. Comparison of Agile Scaling Frameworks. In *2023 The 6th International Conference on Information Science and Systems (ICISS 2023)*, August 11–13, 2023, Edinburgh, United Kingdom. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3625156.3625164>

1 INTRODUCTION / BACKGROUND

There is a need to improve ability to respond to change, reduce time to market, reduce costs, and increase quality of large-scale teams. Agile methods have been proven to support these needs on a small scale. The most common small team approaches are Scrum and Kanban. Scrum is basically an agile, lightweight framework that provides steps to manage and control the software and product development process [1]. Kanban as described by David Anderson, limits work-in-progress according to team capacity, which balances demand against the throughput of team delivered work [2]. The industry has is interested in implementing Agile practices at scale

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICISS 2023, August 11–13, 2023, Edinburgh, United Kingdom

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0820-6/23/08...\$15.00
<https://doi.org/10.1145/3625156.3625164>

for large Initiatives with multiple teams. However it was found that Scrum and all other Agile Methods are not sufficient to achieve the desired benefits for big projects or organizations therefore we need to apply a scaling framework [3]. One question comes up frequently when we look to scale, which framework provides the best approach? This to be a difficult question because context matters. It is critical to understand what your objectives are, what type of work you are supporting, and what the skills are on your teams. Here we use the principles to compare the frameworks and to identify differences and commonalities. The Agile approach was formulated by a group of software professionals with a goal to improve software delivery challenges such as changing requirements, poor quality, and long lead times. The Agile Manifesto [4] written in 2001 promoted the idea that they should value: (i) individuals and interactions over processes and tools; (ii) working software over comprehensive documentation; (iii) customer collaboration over contract negotiation; and (iv) responding to change over following a plan. In comparison to the waterfall lifecycle Agile uses short iterative cycles with frequent customer involvement to incrementally deliver products resulting in increased adaptability, shorter schedules, reduced cost, increased transparency, and higher employee morale. The initial literature focused on small, cross-functional software teams who were collocated. Williams and Cockburn [5] stated that agile methods “best suit collocated teams of about 50 people or fewer who have easy access to user and business experts and are developing projects that are not life-critical”. However, based on the successes Agile has grown in popularity for not only small co-located software teams but large-scale geographically dispersed product development. For this review large-scale is defined as teams larger than 50 People.

2 PROBLEM STATEMENT

Agile has been traditionally used for small, collocated teams of less than 9 people. Agile has shown multiple [9] benefits in small scale software development through increased adaptability, increased transparency, shorter lead times, reduced cost of ownership, increased quality, and improved employee morale. There is a need to bring those benefits to larger scale projects with multiple distributed teams. Over the last 2 decades multiple scaling frameworks have evolved. The most stated reasons behind the creation of scaling frameworks were due to complexity, customer, market, and organization [7]. According to a recent survey documented by A. Putta [6], the two most mentioned reasons for adopting agile scaling frameworks are to scale to more people and to remain competitive in the market. There have been multiple studies on the challenges of scaling but a recent study highlighted that new technologies such as skype and Jira have reduced the issues [8]. Given the multiple options, we need to understand how significant are the differences agile scaling frameworks? What do the frameworks have in common? How do organizations choose the best framework for them.

Table 1: Top ten most used Agile Scaling Frameworks

Framework	Methodologist	Date	% Used
Lean Management	James Womack, Daniel Jones	1990	8%
Enterprise Scrum	Mike Beedle	1997	6%
Large-Scale Scrum (LeSS)	Craig Larman, Bas Vodde	2005	6%
Scrum@Scale / Scrum of Scrums	Jeff Sutherland, Ken Schwaber	2006	28%
Agile Portfolio Management	Jason Krebs	2008	7%
Scaled Agile Framework (SAFe)	Dean Leffingwell	2010	53%
Disciplined Agile (DA)	Scott Ambler, Mark Lines	2011	3%
Spotify	Henrik Kniberg, Anders Ivarsson	2012	7%
Recipes for Agile Governance (RAGE)	Kevin Thompson	2013	1%
Nexus	Ken Schwaber	2015	3%
Other	Various	N/A	8%

3 RESEARCH QUESTIONS AND CONTRIBUTIONS OF THE STUDY

3.1 Research Questions

RQ 1. What principles are different across Agile Scaling Frameworks?

RQ 2. What principles are common across Agile Scaling Frameworks?

RQ 3. Which scaling framework should an organization choose?

3.2 Contributions of this study

The contributions of this study include not only a comparison of the scaling frameworks that exist but also provides insights into the principles that support the frameworks. The benefit of examining the core principles is we can determine if the scaling frameworks are different in their core intent or if they are simply different brands recommending the same core intent. If they are at their core the same, then we can have high confidence in their recommendations given that a core set of technologists have reached similar conclusions.

4 RESEARCH METHOD

The approach for this paper is a mixed methods approach beginning digital.ai’s 16th annual state of Agile report (ASAR), ten most utilized scaling frameworks [9] given in Table 1 in in chronological order, based on their annual survey which received hundreds of thousands of respondents. Some respondents are apparently using more than one framework.

The data was collected from each of the framework websites as well as a literature search. We analyzed the list for differences and commonalities. In some cases where the there was a principle worded differently with the same intent, they were combined into a single line item. Table 2, Aggregated Principles in Scaling Frameworks has a list of all the principles explicitly discussed in the framework literature.

5 SCALING FRAMEWORKS

The Scaling Frameworks were developed to support projects with multiple distributed teams. Key challenges seen on large efforts include resistance to change, cross-team coordination, organizational

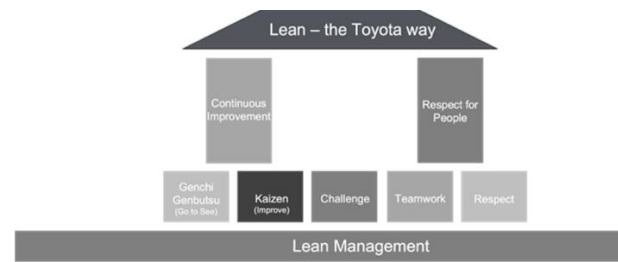


Figure 1: Lean Management

boundaries, requirements engineering, quality assurance, and integration outside of development [10]. In the companies that apply agile scaling frameworks there are two critical transformational patterns required which are they need to be tailored for local context and frameworks require adapting process, tools, and roles in parallel [11]. Martin Kalenda’s paper [12] reinforces the fact that scaling agile within an organization does not need to be prescriptive, the process should be tailored while keeping the core values and principles of agile methodologies intact.

5.1 Lean Management

Lean Management, illustrated in, Figure 1, was introduced in 1990, By James Womack and Daniel Jones. The goal of this framework is to deliver maximum value in the shortest Leadtime. There is a heavy focus on removing waste from the system. While this was one of the frameworks included in the survey, traditionally Lean is thought of as a body of knowledge as opposed to a scaling framework.

Lean Management was first introduced in a book called “The machine that changed the world” [13] by James Womack and Daniel Jones. This was the first book to describe the Toyota Lean Production System [14]. Lean has evolved over many years and continues to deliver exceptional value across a wide variety of domains. Key concepts of Lean Management include Value, Value Stream, Flow, Pull, and Perfection. According to ASAR, Lean Management is utilized by 8% of the respondents.

Table 2: Aggregated principles across the scaling frameworks

Principle	Lean Management	Enterprise Scrum	LeSS	Scrum@Scale	APM	SAFe	DA	Spotify	Nexus	RAGE	% of frameworks
Take an Economic View / (whole Product Focus) / Optimize at the whole			X		X	X					30%
Decentralize Decision Making / (Decentralized Design)		X	X	X		X		X	X	X	70%
Organize around value / (organize around the work) / organize around products and services		X	X	X	X	X		X	X	X	80%
Built-in Quality	X					X	X				30%
Minimum Viable Product / Simplest Product	X					X					20%
Transparency		X	X	X		X	X	X	X		70%
Be Pragmatic				X			X	X			30%
Embrace Change	X	X	X	X	X	X	X	X	X	X	100%
Repeatable results over repeatable process	X					X	X				30%
Encourage Leadership at all Levels			X			X					20%
Add value to the organization by developing your people and partnerships								X			10%
% of principles used	36%	36%	55%	45%	27%	82%	45%	55%	36%	27%	

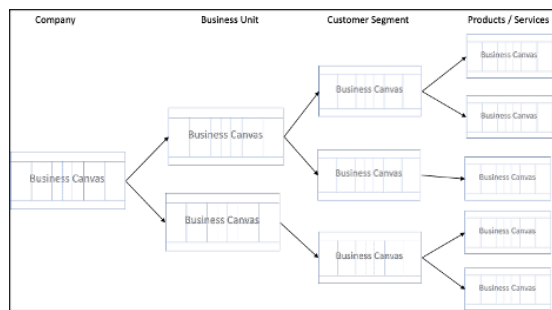


Figure 2: Enterprise Scrum

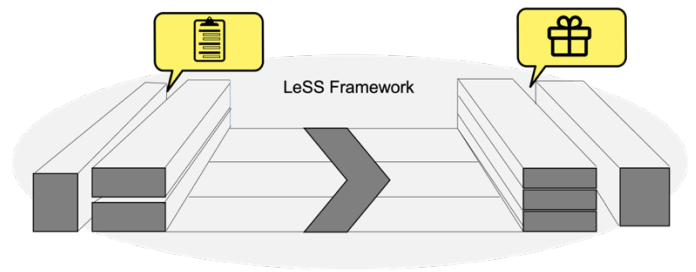


Figure 3: LeSS Framework

5.2 Enterprise Scrum

Enterprise Scrum, illustrated in Figure 2, was introduced in 1997 prior to the Agile Manifesto according ZenTao Blog [15]. Beedle went on to co-author Agile Manifesto. Enterprise Scrums mission is to regenerate the enterprise, including all the enterprise’s business units, business models, processes, products, and services [15]. The approach is a business canvas-based approach. Beedle published, Enterprise Scrum, An Adaptive Method for Project Success in 2013 [16].

The framework pulls from multiple bodies of knowledge including Program Management, Scrum and Kanban. Key concepts of Enterprise Scrum include Visualize work through canvases, and alignment. According to ASAR, Enterprise Scrum is utilized by 6% of the respondents. Enterprise Scrum uses a tiered set of canvases linking company, business unit, customer segment, and products and services to align the work being done across the enterprise.

5.3 Large Scale Scrum (LeSS)

Large-Scale Scrum, illustrated below in Figure 3 was released in 2005 by Craig Larman and Bas Vodde, the authors of [14] Agile and Iterative Development, a manager’s guide. The goal of the LeSS framework is to support up to 8 teams. This framework was originally conceived to solve the specific problem of large projects in multiple locations (Offshoring). One key difference with LeSS over the other frameworks is the recommendation of one backlog and one product owner for the entire implementation. LeSS has an additional option which is referred to as Less-Huge which support greater than 8 teams. I have had the opportunity to implement LeSS Huge on multiple programs and found and found effective. Craig Larman continues to evolve this framework with the last revision published in 2021.

LeSS is derived from three bodies of knowledge including Scrum, Lean, and Systems Thinking. Key concepts of LeSS include transparency, aligned focus, wholistic system optimization, empirical process control, and queuing theory. The difference between LeSS

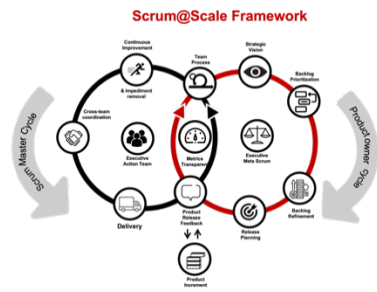


Figure 4: Scrum@Scale Framework

and LeSS Huge is the realization that we need more than one backlog and product owner due to cognitive load. With the addition of Product Owners additional meetings to support coordination have been added. According to ASAR, LeSS is utilized by 6% of the respondents.

5.4 Scrum@Scale

The Scrum@Scale framework illustrated in Figure 4, was introduced in 2006 by Jeff Sutherland and Ken Schwaber, the authors of the Scrum Guide. This framework was introduced to support the goal of coordinating multiple business units. Most of the content is focused on the organizational structure and business rhythms. Scrum at Scale has two cycles: the Scrum Master cycle and the Product Owner cycle creating an intersection of the teams creating a common goal.

Dr. Sutherland continues to evolve this framework with the latest revision published in February of 2022. Ken Schwaber went on to build Nexus, a competing framework that is reviewed in a later section. Scrum@Scale derives from three bodies of knowledge which include Scrum, Lean, and Agile Operating System. Key concepts of Scrum@Scale include transparency, minimal governance, structured formal events, and a flat organizational structure. According to ASAR, Scrum@Scale is utilized by 28% of the respondents in Digital.ai’s survey, which has tripled in use since 2021.

5.5 Agile Portfolio Management (APM)

Agile Portfolio Management (APM), illustrated in Figure 5, was first introduced in 2008 by Jason Krebs in his book Agile Portfolio Management [17]. The goal was to bridge the gap between project teams and executives. This framework does need to be combined with a framework for the team level execution.

APM leverages three bodies of knowledge including Program Management (PMBok), Unified Process, and Scrum. Key concepts are focused on setting strategic direction and managing portfolios. There is not direction regarding execution at the team level as the focus is at the executive level. According to ASAR, APM is utilized by 7% of the respondents.

5.6 Scaled Agile Framework (SAFe)

The Scaled Agile Framework (SAFe), illustrated Figure 6, was introduced in 2010 by Dean Leffingwell, the author of Agile Software Requirements [18]. SAFe is began with a focus on three tiers (Portfolio, Program, and Team). The goal of SAFe was helping organizations design software and systems to make the world a better place.

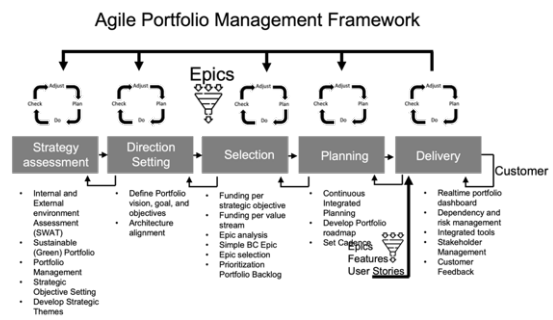


Figure 5: Agile Portfolio Management Framework (APM)

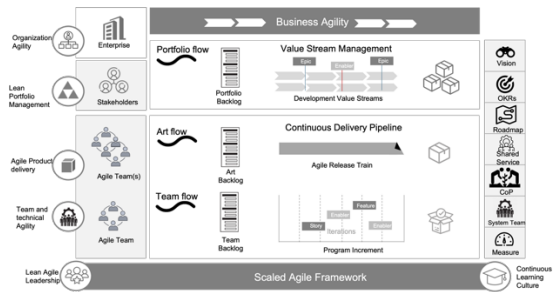


Figure 6: Scaled Agile Framework 6.0

The framework continuously evolves with the latest major revision delivered in 2023. The framework pulls from multiple bodies of knowledge including Scrum, Kanban, Lean, Systems Thinking, Design Thinking, and DevOps. Key concepts of SAFe include transparency, built-in quality, alignment, wholistic optimization, empirical process control, queuing theory, and continuous learning culture. The most common cited benefits from companies who have implemented SAFe are transparency, alignment, quality, productivity, collaboration, and time to market [19]. According to ASAR, Scaled Agile Framework (SAFe) is utilized by 53% of the respondents making it the most selected Scaling Framework available on the market. Since last years “State of Agile” Survey they have increased by more than 15%.

5.7 Disciplined Agile (DA)

Disciplined Agile Delivery (DaD), illustrated in Figure 7, was introduced in 2012 by Scott Ambler, the author of Disciplined Agile Delivery [20]. This framework is derived from Unified Process, extreme programming (XP), and Kanban. A key goal of DAD is to cover the entire delivery lifecycle from initial concept to delivery for operations and support [21]. The framework scales both tactically and strategically. The framework evolved to Disciplined Agile in 2015 and was acquired by the Project Management Institute (PMI) in 2019.

This framework continues to evolve with the last published update in 2022. Since the move to PMI there has been a great deal added to this framework around ITSM. The framework pulls from multiple bodies of knowledge including Systems Thinking, Design Thinking, ITSM, DevOps, and the Agile Operating system. Key

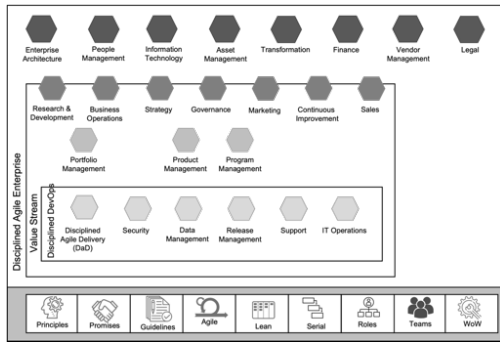


Figure 7: Disciplined Agility

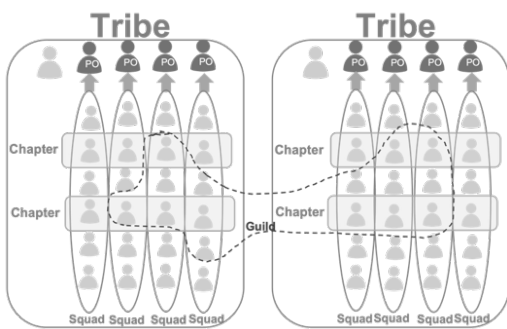


Figure 8: Spotify Model

concepts of DA include Pragmatism, Validated Learning, improve culture by improving the system. According to ASAR, DA is utilized by 3% of the respondents.

5.8 Spotify

Spotify is a leading music company who has been Agile since their inception. The company is in several cities and time zones which required a new approach to coordination. The Spotify scaling model, illustrated in Figure 8, was introduced in 2012 by Henrik Kniberg and Anders Ivarsson who authored a whitepaper *Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds*. The Spotify tribe engineering model consists of 7 organizational elements with the chief architect being the critical element to maintain integrity. The Spotify model is not considered a framework, it is a narrative regarding Spotify’s view on Scaling from a technical and a cultural perspective.

This body of knowledge was developed as an organizational blueprint it has not been updated since 2012. However, because the blueprint is easy to follow, there are many organizations that adapted the blueprint and evolve internally based on their context. Spotify pulls from two bodies of knowledge including Scrum and Kanban. Key concepts in Spotify are autonomy, minimal governance, and a flat organizational structure. This model focuses heavily on the organizational structure but everything else is left up to team preference. According to ASAR, Spotify is utilized by 7% of the respondents, which impressive due to nonexistent marketing.

5.9 Nexus

The Nexus framework, illustrated in Figure 9, was introduced in 2015, by Ken Schwaber. As mentioned previously Ken also author the original Scrum@Scale Framework. Ken stayed close to pure scrum but needed to add coordination and integration mechanisms to support multiple teams. The Nexus integration team is a unique element that Ken added, [16] This team is responsible for ensuring each delivery at the end of a sprint. To this end, they are responsible for solving the technical and non-technical problems that inhibit the Scrum teams from achieving their goals.

This framework continues to evolve with the last published update in 2021 The framework extends Scrum to support 3-9 teams. Key concepts of Nexus include Transparency, Team Organization, Work Organization. According to ASAR, Nexus is utilized by 3% of the respondents. Nexus seeks to preserve the original pure scrum to the extent possible with a bottom- up team approach that leverages empirical process control.

5.10 Rage

The Recipes for Agile Governance (Rage) Framework, illustrated in, Figure 10 was introduced by Kevin Thompson from Cprime in 2013.

The framework pulls from two bodies of knowledge including Program Management and Scrum. Key concepts of RAGE include tiered governance, structured roles and Responsibilities, and performance benchmarks. According to ASAR, RAGE is utilized by 3% of the respondents.

6 RESULTS AND DISCUSSION

The key findings of the research question are discussed in sections 6.1 – 6.3.

6.1 RQ1. What principles are different across Agile Scaling Frameworks?

There are minimal differences in principles between the frameworks. The differences that exist are associated with concepts that have not been added to frameworks when they stopped evolving. The principles that are included in the least amount of the frameworks are minimum viable product (MVP), encourage leadership at all levels, and add value to your organization by developing your people. These are relatively new over the last few years and in all cases the frameworks that have not included them have also not been updated recently.

6.2 RQ2. What principles are common across Agile Scaling Frameworks?

While in some cases the principle is worded differently there is extensive commonality in principle intent across the agile scaling frameworks. The principles that are common across most of the frameworks are decentralize decision making, organize around value, transparency, and embracing change with each used by greater than 70% of the frameworks. This finding should provide confidence in their recommendations based on several technologists evolving to the same conclusion. In addition, the degree of

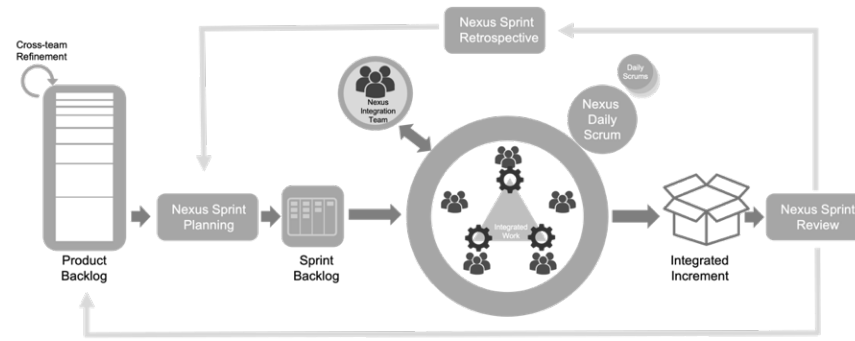


Figure 9: Nexus Scaling Framework

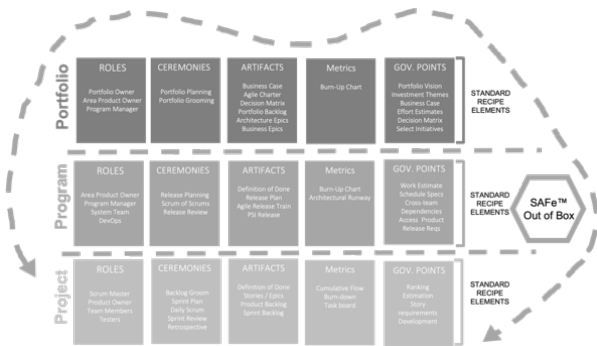


Figure 10: Recipes for Agile Governance (RAGE)

overlap suggests that the framework organizations choose may not impact outcomes as much as previously thought.

6.3 RQ3. Which framework should your organization choose?

Since there are minimal differences in the agile scaling frameworks based on core principles and recommendations the other factors we should consider when selecting a scaling framework should be evolving content, flexibility in application, accessibility of artifacts, and community. During the review frameworks such as Enterprise Scrum, Spotify, and APM have not been evolved content in many years. The world continues to evolve if the framework does not it may cease to be able to address current problems. The frameworks that had the greatest amount of flexibility in how to apply were SAFe, Disciplined Agility, and Scrum@Scale. Each of them had multiple practices to select based upon your organization’s unique goals. The frameworks with the most accessible content based on searching and downloads were SAFe, LeSS, and Scrum@Scale all with robust web content. The Framework with the largest community was SAFe with 53% of the respondents in Digital.ai’s survey claiming use.

7 CONCLUSION AND RECOMMENDATIONS

Agile has provided several documented benefits including in increased adaptability, shorter schedules, reduced cost, increased transparency, and higher employee morale for small software teams.

We recognize the need to scale these benefits up to multiple teams distributed geographically. After comparing the agile scaling framework principles, the study has found that all of frameworks are generally similar in principle intent. For organizations who are looking for advice on which framework to select we hypothesize the best framework is one with evolving content, flexibility in application, accessibility of artifacts, and extensive community to collaborate with. It may be that industry has already reached a similar conclusion in that we can see 53% of the respondents in ASAR claim to be using the Scaled Agile Framework (SAFe).

REFERENCES

- [1] Srivastava, Apoorva, Sukriti Bhardwaj, and Shipra Saraswat. "SCRUM model for agile methodology." *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2017.
- [2] David, J, Anderson, "Kanban: Successful Evolutionary Change for Your Technology Business". Sequim, WA: Blue Hole Press, 2010
- [3] Theobald, Sven, Anna Schmitt, and Philipp Diebold. "Comparing scaling agile frameworks based on underlying practices." *International Conference on Agile Software Development*. Springer, Cham, 2019.
- [4] Beck, K., et al., Manifesto for Agile Software Development, Agile Alliance., 2001.
- [5] Williams, L. and A. Cockburn, Agile software development: it’s about feedback and change. *Computer*, 2003. 36(6): p. 39-43.
- [6] Putta, Abheeshta, et al. "Why do organizations adopt agile scaling frameworks? A survey of practitioners." *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 2021.
- [7] Uludağ, Omer, et al. "Evolution of the agile scaling frameworks." *Agile Processes in Software Engineering and Extreme Programming: 22nd International Conference on Agile Software Development, XP 2021, Virtual Event, June 14–18, 2021, Proceedings*. Cham: Springer International Publishing, 2021.
- [8] Girma, Melaku, Nuno M. Garcia, and Mesfin Kifle. "Agile scrum scaling practices for large scale software development." *2019 4th International Conference on Information Systems Engineering (ICISE)*. IEEE, 2019.
- [9] 16th Annual State of Agile Report. 2023, Digital.AI
- [10] Dikert, K., Paasivaara, M., Lassenius, C., 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software* 119, 87–108. <https://doi.org/10.1016/j.jss.2016.06.013>
- [11] Ebert, Christof, and Maria Paasivaara. "Scaling agile." *Ieee Software* 34.6 (2017): 98-103.
- [12] Kalenda, Martin, Petr Hyna, and Bruno Rossi. "Scaling agile in large organizations: Practices, challenges, and success factors." *Journal of Software: Evolution and Process* 30.10 (2018): e1954
- [13] Womak, J.P., D.T. Jones, and D. Roos, *The Machine that Changed the World*. Vol. Simon & Schuster. 2007, London, England.
- [14] Larman, C., *Agile and iterative development: a manager’s guide*. 2004: Addison-Wesley Professional.
- [15] Jing, [A Brief History of Agile] Mike Beedle – Tragic Loss to Scrum Community, in ZenTao Blog, ZenTao, Editor. 2022, ZenTao.
- [16] Beedle, M.A., *Enterprise Scrum: An Adaptive Method for Project Success*. 2013: Addison Wesley Professional.
- [17] Krebs, J., *Agile portfolio management*. 2008: Microsoft Press.

- [18] Leffingwell, D., *Agile software requirements: lean requirements practice for teams, programs, and the enterprise*. 2010: Addison-Wesley Professional.
- [19] Putta, Abheeshta, Maria Paasivaara, and Casper Lassenius. "Adopting scaled agile framework (SAFe) a multivocal literature review." *Proceedings of the 19th International Conference on Agile Software Development: Companion*. 2018.
- [20] Ambler, S.W. and M. Lines, *Disciplined agile delivery: A practitioner's guide to agile software delivery in the enterprise*. 2012: IBM press.
- [21] Takeuchi, H. and I. Nonaka, The new new product development game. *Harvard business review*, 1986. 64(1): p. 137-146.