

Aug 9th, 12:00 AM

Practices for Large-Scale Agile Transformations: A Systematic Literature Review

Lennard Trippensee
University of Applied Sciences Wedel, l-trippensee@hotmail.de

Gerrit Remané
University of Applied Sciences Wedel, gre@fh-wedel.de

Follow this and additional works at: <https://aisel.aisnet.org/amcis2021>

Recommended Citation

Trippensee, Lennard and Remané, Gerrit, "Practices for Large-Scale Agile Transformations: A Systematic Literature Review" (2021). *AMCIS 2021 Proceedings*. 5.
https://aisel.aisnet.org/amcis2021/it_projmgmt/it_projmgmt/5

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2021 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Practices for Large-Scale Agile Transformations: A Systematic Literature Review

Completed Research

Lennard Trippensee

University of Applied Sciences Wedel
imca103936@fh-wedel.de

Gerrit Remané

University of Applied Sciences Wedel
gerrit.remane@fh-wedel.de

Abstract

Agile methodologies have long been successfully applied to smaller software development projects with limited team sizes. In recent years, some firms have started to transfer agile methodologies to the whole organization. However, large-scale agile transformations are challenging and require a comprehensive view of cultural, operational, and strategic aspects. This paper systematically reviews available large-scale agile practices from the literature and provides a holistic overview of different large-scale agile practices. In total, four practice categories, 11 practices, and 19 example concepts were found. The categories are organized in a framework and serve as a possible entry point for researchers and managers to approach the topic of large-scale agile development more efficiently and economically.

Keywords

Agile at scale, large-scale agile software development

Introduction

Since the agile manifesto release in 2001 (Beck et al. 2001), agile software development has become progressively popular. It can increase time efficiency and software quality (Livermore 2008) and has been used successfully on a smaller scale. Due to several advantages such as higher customer satisfaction, increased employee motivation, better productivity, organizations such as Amazon, Salesforce, Yahoo have started to scale agile methodologies beyond individual teams (Dikert et al. 2016; Dyba and Dingsoyr 2009). Large-scale agile typically applies to organizations where 50 or more people in at least six teams work according to agile principles (Dikert et al. 2016).

However, the transformations to large-scale agile development are complicated as they tackle operational, strategic, and cultural aspects (Karvonen et al. 2018). A literature review conducted by Dikert et al. (2016) showed that large-scale agile methods are insufficiently addressed in the literature. Many organizations have problems tailoring the agile methods to their needs. Even though scaling frameworks like Scaled Agile Framework ("SAFe") offer an "all-in-one" solution, most organizations use a mix of different methods and practices (Laanti and Kettunen 2019), thereby creating a need for a complete overview of large-scale practices that are not confined by a single framework.

This review tries to fill this gap by providing a holistic overview of large-scale transformation practices. Furthermore, this review aims to give managers a brief outline of the topic and a structure to identify gaps in their transformation approaches.

The review includes four parts. After the introduction, the term "large-scale agile" is defined. The second part describes the research methodology in more detail. Subsequently, the qualitative results are presented. Finally, the last part discusses the results and gives an outlook on future research.

Research method

We conducted a systematic literature review on "agile at scale transformation practices". The search for possible sources was not confined to geographical region, research methodology, or one journal set (Webster and Watson 2002).

To reduce the likelihood of bias as suggested by Kitchenham and Charters (2007), we defined three inclusion criteria:

- study contents should be about large-scale agile development
- study contents should be about agile transformations in an organization
- study contents should sufficiently describe at least one large-scale agile practice

Afterward, a preliminary analysis of the topic determines each criterion's optimal search strings. Therefore, the database "Google Scholar" was searched with the search terms "agile at scale" and "large-scale agile", and the results to identify potential search strings were identified. For the criterion "large-scale agile", the search terms "large-scale" and "agile at scale" were found. To fulfill the criterion "organizational transformation", we selected the search terms "transformation", "implementation", "adoption", "introduction", and "rollout". The criterion "large-scale agile practices" was not included in the preliminary analysis; instead, it is achieved by the filtering steps below. The final search string is composed as follows: ("large-scale agile" OR "agile at scale") AND (transformation OR adoption OR introduction OR rollout OR roll-out).

After the preliminary search term analysis, we searched the following databases using the defined search strings: ACM Digital library, ScienceDirect, IEEE Explore, EBSCO Host, and AISeL. Two filters refined the searches afterward. Initially, we filtered the source types. Thereby newspaper articles were not included in the search results. For the next step, we limited the language by only including English publications in the results. 541 eligible papers were found and used for further analysis.

To select relevant literature, we applied a four-step filtering process. First, we removed the duplicates. Secondly, we reviewed the titles of the publications for relevance. Papers with a title not clearly related to agile at scale were removed. Ambiguous cases were kept and filtered in the following steps. Thirdly, we read the abstracts to verify whether the three inclusion criteria listed earlier were met. Finally, we checked the remaining 132 publications for compliance with the criteria by analyzing the entire study. In particular, papers were rejected that provided a reference to practices in their abstract but did not provide a sufficient description in the full text. A sufficient description is given if the practices are a central part of the analysis and presented comprehensively. The detailed steps are shown in table 1.

Filter process step	Number of results
Database results	541
1 Remove duplicates	428
2 Relevance by title	276
3 Relevance by abstract	132
4 Relevance by full text	19

Table 1. The filtering steps and their numbers of results

We categorized the selected literature into four different practice categories (see Appendix 1). The categories agile program and agile enterprise practices were adopted from Laanti (2013). Since the paper refers to "large-scale agile" practices (see introduction) and much literature already exists for team-level practices, we did not examine team-level practices further (e.g., Brhel et al. 2015; Eloranta et al. 2016; Hossain et al. 2009). Nevertheless, it should be noted that team-level practices play an essential role in the agile transformation and must be incorporated into a transformation approach. Furthermore, we identified transformation practices with the two widely used transformation approaches bottom-up or top-down and agile scaling frameworks as a separate category because they mostly consist of combining different practices.

Results

The following sections briefly describe the agile practice framework (see figure 1), including its four categories and twelve practices. Every practice serves a different goal and attempts to solve a specific problem. Considering that there are many ways to put the practices in concrete actions, the following categories describe typical example concepts to explain how the practices could be implemented. Finally, table 2 provides a condensed overview.

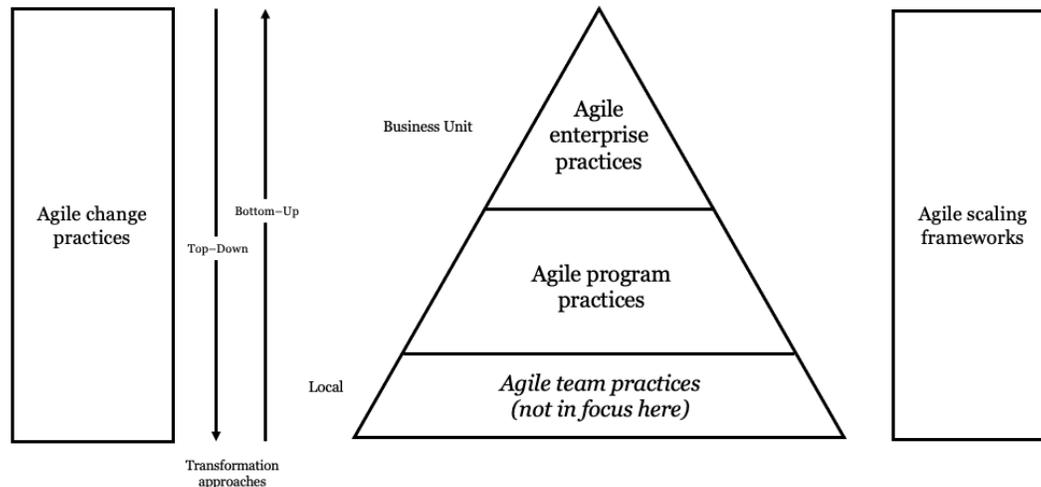


Figure 1. The different agile practices framework (adapted from Laanti 2013)

Agile change practices

In this category, we grouped practices that focus on the agile transformation change itself. The group includes practices to perform and support the transformation process to adopt agile at scale organization-wide. It should be noted that these practices are not exclusive to agile transformations. However, an "agile change approach" with iterative feedback loops and continuous learning and improvement can help manage an agile large-scale transformation (Taribuka and Amrit 2020). The category is intended to provide an overview of the change practices analyzed in the literature to support practitioners' agile transformation process.

The literature reveals two generally used transformation approaches. The bottom-up approach usually starts the transformation at the team level and proceeds hierarchically upwards. The other identified approach, the top-down approach, begins at the enterprise/business unit level and conducts the change hierarchically downwards (Dikert et al. 2016; Schuch et al. 2020). The majority of the analyzed transformations use the bottom-up approach.

We identified two concrete change practices. The first one is the establishment of a cross-functional change team. The team's goal is to spread the agile mindset, analyze processes, and manage the transformation (Fry and Greene 2007). The change team usually consists of in-house staff and often involves the agile teams' leaders (Fry and Greene 2007; Garzón et al. 2018).

The second practice is the use of agile methodologies to perform the transformation. It is motivated by the given uncertainty and the high risk of large changes. Two agile methodologies were used in particular: Scrum- and Lean-change. For the scrum methodology, the change team divides the change process into sprints. With each sprint, the change progresses iteratively. The sprint's output primarily consists of established, smaller scaling practices within a relatively small group of teams (Fry and Greene 2007; Garzón et al. 2018). The lean-change method works similarly to the iterative process. Based on the "lean startup" concept by (Ries 2011), the methodology uses "Minimum Viable Changes" to validate the transformation assumptions. E.g., "four motivated, agile adopters can be found to bring the agile transformation into the company". In a build, measure, learn cycle, the riskiest assumptions are first checked for validity. E.g.,

conducting agile workshops and testing whether enough enthusiasts can be found. Afterward, the assumption is checked, and new assumptions are made based on feedback (Hui 2013). Agile changes may increase the chances of success and reduce transformation duration (Hui 2013; Karvonen et al. 2018).

Agile program practices

This category summarizes all practices which are supposed to ensure agility within a program. It contains practices that solve typical cross-team problems. The primary goal is flexible overarching team coordination aimed towards aligning all teams to create value. The category differentiates between five practices: (1) inter-team coordination practices, (2) Knowledge sharing practices, (3) Technical integration practice, (4) Continuous improvement practices, and (5) Architecture practices. Inter-team coordination practices aim at coordinating teams and aligning them towards a common goal. We found three different concepts among the described practices. The first concept is the establishment of demos. Demos are rounds of presentations in which, for example, a team presents a newly created feature. The presentation of current topics helps teams to understand what the other teams are currently working on and breaks the "silo-thinking" (Kalenda et al. 2018; Paasivaara et al. 2018; Rolland 2016). The second concept is the establishment of a "heartbeat". A heartbeat is a synchronized iteration to which all teams adhere. The synchronization is supposed to help align teams to perform better consistent release cycles (Fry and Greene 2007; Putta et al. 2018; Uludağ, Kleeaus, et al. 2019). The last concept is the introduction of coordination meetings. Seven publications mentioned at least one form of coordination meeting, usually involving all relevant stakeholders from the various teams. Those meetings aim to identify challenging issues in the next iteration or align the teams to the strategy. (Fuchs and Hess 2018; Rolland 2016)

Knowledge sharing practices aim to spread knowledge and best practices within the company and break down knowledge silos. All examined publications that described knowledge sharing practices used one concept: Communities of Practices (CoP). CoPs are groupings of experts with similar areas of expertise. The experts set up groups voluntarily to exchange information on the topic regularly. (Fuchs and Hess 2018; Uludağ, Kleeaus, et al. 2019) This exchange aims to avoid the repetition of errors and distribute knowledge more effectively within the company. CoPs have a wide range of benefits, including helping to drive strategy, starting new lines of business, providing an arena for problem-solving, transferring best practices, developing professional skills, and increasing the retention of talent. (Wenger et al. 2002)

Technical integration practices are practices to support and enable technical integration and agile development. The analyzed literature introduced two concepts. The first concept describes technology's use to enable continuous integration of software components (Fuchs and Hess 2018; Rolland 2016; Smart 2018). The choice of technology is very case-specific, e.g., Jenkins. The second concept is the use of software to support the management of agile methodologies, e.g., Jira (Paasivaara et al. 2018). The software is usually designed to provide digital support to the teams to provide transparency and simplify coordination also across different teams.

The area "Continuous improvement" contains practices to improve scaling practices continuously. The literature mentioned a familiar concept. So-called "scaled retrospectives" can be carried out, in which, similar to the one-team variant, retrospectives are performed with all teams. Representatives of the individual teams meet regularly to talk about the methodology and give feedback for improvement. The responsible persons take up the suggestions and improve them (Kalenda et al. 2018). Such feedback loops can take place at all levels and serve to improve the processes continuously.

Architecture practices are used to coordinate the (software) architecture. We were able to identify two concepts. First, "Undone departments" introduces a group supporting development teams to achieve a full iteration/Sprint. The group helps with all team-wide concerns from architectural design to quality assurance (QA) activities (Kalenda et al. 2018). Second is the practice of introducing an architecture committee, a group of people who meets regularly and makes relevant architectural decisions, much like a specialized community of practice (Uludağ, Nägele, et al. 2019; Uludağ, Proper, et al. 2019). The difference between the two practices is that undone departments are more flexible, while architecture committees hold regular meetings to discuss architectural topics.

Agile enterprise practices

The following section groups the practices that improve the company/enterprise-agility. This section covers two practices.

The first practice is to build an agile mindset within the organization. The practice can be implemented by convening annual “Future Vision”- meetings for the entire organization. Agile coaches, the management team, and developers discuss and define the core agile values in these meetings. To share the values with the whole organization, another specific concept can be to set up “value workshops” where employees create a shared vision through multiple workshops (Paasivaara et al. 2018).

The second practice is the use of leadership to support agility. It is essential to involve middle management and assign new roles (Smart 2018). Furthermore, the management should build new leadership competencies and act as an ambassador of the agile mindset. A leader with a “Transformational Leadership”-skillset encourages the team to work towards a common goal through vision, communication, intellectual stimulation, personal recognition, and the evident caring about the followers' personal needs (Rafferty and Griffin 2004). Transformative leaders enable the necessary agile practices and impact the team's IT performance (Puppet Labs 2017).

Agile Scaling frameworks

Agile scaling frameworks are frameworks to support the scaling of agile software development in an organization. There are nine different large-scale agile frameworks listed in the Agile Scaling Knowledgebase Decision Matrix (Dolman and Spearman 2017). According to the 14th annual state of agile report, the two most popular frameworks are the Scaled Agile Framework (“SAFe”) and Scrum of Scrums (Digital.ai 2020). Within the analyzed literature, six publications referred to frameworks. 66% of those mentioned the framework SAFe, and 33% the framework Large Scale Scrum (“LeSS”). The following sections describe both frameworks and tailoring processes to customize agile practices.

Craig Larman and Bas Vodde released the LeSS framework in 2008 (Larman and Vodde 2008). The idea behind the Large Scale Scrum (“LeSS”) framework is to take the basic one-team Scrum and scaled it to fit in a larger environment. There are two versions of LeSS. The “normal” LeSS is applicable for up to eight Scrum teams. The other version is the “LeSS Huge”, which can be applied to a few thousand people on one product (Paasivaara and Lassenius 2016). It preserves many practices already known from the one-team Scrum:

- single product backlog
- the same definition of “done” for all teams
- synchronized sprints leading to a potentially shippable product after each sprint
- single product owner

Scrum meetings are modified for multiple-team development by including members from other or all teams. A fundamental principle behind the “LeSS Huge Frameworks” is to divide the product into requirement areas, each having an Area Product Owner with 4-10 feature teams and prioritized area backlogs. Every area performs its sprint planning, meetings, sprint reviews, and retrospectives (Uludağ, Kleehaus, et al. 2019).

Dean Leffingwell designed the Scaled Agile Framework (SAFe) to scale agile software development to large enterprises. The framework features four levels: team-, program-, portfolio-, and the optional value-stream-level (Alqudah and Razali 2016; Vaidya 2014). It utilizes Scrum with XP engineering practices at the team level, but applying Kanban is also possible. The framework introduces agile release trains (ARTs) at the program level, which is the analogy to Sprints at the team level, operating at a slower time frame (Paasivaara 2017). ART's follow the HIP (Hardening, Innovation, Planning) iterations to develop the Potential Shippable Increments/Program Increments planned during the release planning days. The program level also includes additional roles, e.g., system architect, product manager, system team, release train engineer, and release management team (Alqudah and Razali 2016; Vaidya 2014). The portfolio level allows agile alignment of strategy and execution. The alignment is done as epics that define large development initiatives. The value stream level supports the development of large and complex solutions, which require multiple, synchronized ARTs (Paasivaara 2017).

Even if the frameworks offer a theoretically working concept to scale agile practices, according to a study by (Laanti and Kettunen 2019), only 5% of respondents stated that they are using SAFe as the only method. Seven of the examined publications did not adopt a uniform methodology but instead tailored their own. We found two general practices for tailoring. The first practice is "cherry-picking", where the most suitable practices are picked from various methodologies (Garzón et al. 2018), e.g., Lean, Scrum, Extreme Programming. Using cherry-picking as the only practice can be difficult because no "overreaching" structure is in place to connect the practices (Paasivaara et al. 2018), e.g., one standardized way to communicate regularly. The second practice is adding, removing, or modifying a given method or framework to fit the requirements. Three of the examined studies used this practice to customize their large-scale practices or frameworks (Fry and Greene 2007; Fuchs and Hess 2018; Uludağ, Kleehaus, et al. 2019).

Practice category	Practices	Example concepts
Agile change practices	Leverage agile methods for change process	Lean-change
		Scrum-change
	Build up a cross-functional change team	<i>(no concept given)</i>
Agile program practices	Implement inter-team coordination practices	Organize demos
		Synchronize the sprints (Create a "heartbeat")
		Organize coordination meetings (planning, requirements)
	Implement knowledge sharing practices	Communities of practice
	Implement technical integration practices	Improve integration by leveraging technology
		Integrate software to support the management of agile practices
Implement continuous improvement practices	Scaled retrospective	
Implement architecture practices	Architecture communities	
	Undone department	
Agile enterprise practices	Build an organization-wide agile mindset	Value workshops
		Organize future vision meetings
	Leverage leadership to support agility	Establish new competencies in the management
		Give middle management a roll
Agile Scaling frameworks	Tailor practices	Cherry-pick the practices
		Add, remove, or modify a given method or framework
	Use agile scaling frameworks	Large Scale Scrum ("LeSS")
		Scaled Agile Framework ("SAFe")

Table 2. Practices found in the literature

Discussion

Since agile methods were initially designed for smaller teams (Boehm and Turner 2005), most of the literature deals with agility in general (e.g., Diegmann et al. 2018). However, with the growing trend to apply agile methods on a larger scale, the topic of large-scale agility is becoming more critical than ever (Uludağ et al. 2020). Thus, this paper makes several contributions to existing research. First, this paper complements the various sources on agile team practices that can be used to implement agility on a program or enterprise level. Second, the paper arranges existing large-scale agile practices in a helpful structure that

emphasizes the different focus of the various practices. Third, the framework provides a solid foundation for further investigating large-scale agile practices in the future, independent of commercial frameworks.

Practitioners can use this paper to further structure the large-scale agile transformation within their organization. The developed framework can be helpful to classify transformation-relevant processes and to compare known scaling frameworks for applicability. Furthermore, the concrete concepts provide an overview of ready-to-use tools to help to implement large-scale agile in an organization.

Finally, this paper provides a foundation for further research on large-scale agile practices. Overall, there is little literature on large-scale agile transformation practices. Many publications only name the practice but do not describe them in detail. There is a need for more case studies that describe concrete practices. Furthermore, different terms were often used to describe similar practices. Uniform terminology for practices could make research on the topic more accessible. Even though the concepts “lean-change” and “scrum-change” are well-known, introducing agile at scale in an organization still lacks more detailed research. The practices on program agility are more matured but leave a gap regarding the alignment with the organizational strategy. Further, enterprise practices and tailoring practices lack the level of detail and overall diversity. With agile scaling frameworks, practices are visible and well known. However, it would be valuable to compare the underlying practices in more detail and map their strength and weaknesses regarding the different practice categories. The following questions about large-scale practices appear particularly promising for future research:

- How should the selection process of agile frameworks look like?
- How do you implement a hybrid-agile-change (agile and non-agile techniques)?
- What practices are used to align the products with the organizational strategy?
- What scaling practices are used to enable agility on the enterprise level?
- What practices are used to tailor agile scaling frameworks?

Our study is not free of limitations. First, the literature review might not be exhaustive due to search string, limited databases included, and new papers that might have been published after this research. We conducted our analysis based on five different databases. Therefore, literature that cannot be found in the databases was not included in the review. In order to meet all inclusion criteria, much literature was filtered out manually. The selected papers are therefore subject to a selection bias. By filtering the title first, relevant papers with ambiguous titles could be excluded from the analysis. Second, we argue that the classification of practices provides a helpful overview of these practices for further research and their application in business practice. However, this classification is certainly not perfect and very like subject to our individual experiences and bias (Nickerson et al. 2013). Third, we summarized several practices as duplicates, even though they were differently named. We could have avoided this by skipping this step, but we were convinced that the benefit of not having to review very similar or identical practices multiple times outweighs this disadvantage.

Conclusion

The literature review provides a holistic view of large-scale agile practices from 16 studies. It describes in four practice categories, 11 practices, and 19 example concepts a possible entry point for researchers and managers to approach the topic of large-scale agile more easily. Thereby, it hopefully aids in making the tremendous benefits of agile working methods also more easily available for large-scale agile transformations.

Appendix 1: Literature Matrix

Source	Type	Organization	Agile change practices		Agile program practices					Agile enterprise practices		Agile scaling frameworks	
			Build up cross-functional change team	Implement inter-team coordination practices	Implement inter-team coordination practices	Implement knowledge sharing practices	Implement technical integration practices	Implement continuous improvement practices	Implement architecture practices	Build an organization-wide agile mindset	Leverage leadership	Tailor practices	Use agile scaling frameworks
Fry et al. 2013	Experience report	Salesforce	x	x						x	x	x	
Hui 2013	Experience report		x	x	x					x		x	
Rolland 2016	Case Study				x	x	x		x				
Paasivaara and Lassenius 2016	Case Study	Nokia										x	x
Pries-Heje and Krohn 2017	Case Study	SimCorp											x
Paasivaara 2017	Case Study	Comptel											x
Putta et al. 2018	Review												x
Smart 2018	Experience Report	Berkley		x			x	x		x	x		
Garzón et al. 2018	Experience Report	GMV	x	x								x	
Paasivaara et al. 2018	Case Study	Ericsson		x	x	x	x			x		x	
Karvonen et al. 2018			x	x									
Kalenda et al. 2018	Review				x	x		x	x				
Fuchs and Hess 2018	Case Study (multiple)	Insure Tech; EventCom	x	x	x	x	x				x	x	
Putta 2018	Review		x		x				x	x			x
Uludağ, Proper, et al. 2019	Case Study (multiple)								x				
Uludağ, Kleehaus, et al. 2019	Case Study				x	x			x	x	x	x	x
Uludağ, Nägele, et al. 2019									x				
Taribuka and Amrit 2020	Case Study (multiple)		x								x		
Kischelewski and Richter 2020	Review		x	x	x								

REFERENCES

- 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), pp. 828–837. (<https://doi.org/10.18517/ijaseit.6.6.1374>).
- Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, A., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. 2001. "Manifesto for Agile Software Development." (<https://agilemanifesto.org/>, accessed December 8, 2020).
- Boehm, B., and Turner, R. 2005. "Management Challenges to Implementing Agile Processes in Traditional Development Organizations," *IEEE Software* (22:5), pp. 30–39. (<https://doi.org/10.1109/MS.2005.129>).
- Brhel, M., Meth, H., Maedche, A., and Werder, K. 2015. "Exploring Principles of User-Centered Agile

- Software Development: A Literature Review,” *Information and Software Technology* (61), Elsevier B.V., pp. 163–181. (<https://doi.org/10.1016/j.infsof.2015.01.004>).
- Diegmann, P., Binzer, B., Dreesen, T., and Rosenkranz, C. 2018. “Journey towards Agility: Three Decades of Research on Agile Information Systems Development,” *International Conference on Information Systems 2018, ICIS 2018* (December).
- Digital.ai. 2020. “14th Annual STATE OF AGILE REPORT,” *Annual Report for the STATE OF AGILE* (14:14), pp. 2–19. (<https://explore.digital.ai/state-of-agile/14th-annual-state-of-agile-report>).
- Dikert, K., Paasivaara, M., and Lassenius, C. 2016. “Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review,” *The Journal of Systems & Software* (119), Elsevier Inc., pp. 87–108. (<https://doi.org/10.1016/j.jss.2016.06.013>).
- Dolman, R., and Spearman, S. 2017. “Agile Scaling Knowledge - The Matrix.” (<http://www.agilescaling.org/ask-matrix.html>, accessed December 9, 2020).
- Dyba, T., and Dingsoyr, T. 2009. “What Do We Know about Agile Software Development?,” *IEEE Software* (26:5), pp. 6–9. (<https://doi.org/10.1109/MS.2009.145>).
- Eloranta, V. P., Koskimies, K., and Mikkonen, T. 2016. “Exploring ScrumBut - An Empirical Study of Scrum Anti-Patterns,” *Information and Software Technology* (74:December), pp. 194–203. (<https://doi.org/10.1016/j.infsof.2015.12.003>).
- Fry, C., and Greene, S. 2007. “Large Scale Agile Transformation in an On-Demand World,” in *Proceedings - AGILE 2007*, pp. 136–142. (<https://doi.org/10.1109/AGILE.2007.38>).
- Fuchs, C., and Hess, T. 2018. “Becoming Agile in the Digital Transformation: The Process of a Large-Scale Agile Transformation,” *International Conference on Information Systems 2018, ICIS 2018*.
- Garzón, H., Chaos, J. A., Martín, B., and Garzás, J. 2018. *Large Scale Agile Transformation: A Case from Satellite Monitoring and Control (Short Version)*, (July), Germany, Europe: Unpublished. (<https://doi.org/10.13140/RG.2.2.33026.53446>).
- Hossain, E., Ali Babar, M., and Paik, H. Y. 2009. “Using Scrum in Global Software Development: A Systematic Literature Review,” *Proceedings - 2009 4th IEEE International Conference on Global Software Engineering, ICGSE 2009*, pp. 175–184. (<https://doi.org/10.1109/ICGSE.2009.25>).
- Hui, A. 2013. *Lean Change: Enabling Agile Transformation through Lean Startup, Kotter and Kanban: An Experience Report*, Institute of Electrical and Electronics Engineers (IEEE), October 2, pp. 169–174. (<https://doi.org/10.1109/agile.2013.22>).
- Kalenda, M., Hyna, P., and Rossi, B. 2018. “Scaling Agile in Large Organizations: Practices, Challenges, and Success Factors,” *Journal of Software: Evolution & Process* (30:10), John Wiley & Sons, Inc., N.PAG-N.PAG. (<http://10.0.3.234/smr.1954>).
- Karvonen, T., Sharp, H., and Barroca, L. 2018. “Enterprise Agility: Why Is Transformation so Hard?,” in *Lecture Notes in Business Information Processing* (Vol. 314), Springer Verlag, pp. 131–145. (https://doi.org/10.1007/978-3-319-91602-6_9).
- Kischelewski, B., and Richter, J. 2020. “Implementing Large-Scale Agile - an Analysis of Challenges and Success Factors,” *Proceedings of the 28th European Conference on Information Systems (ECIS), An Online AIS Conference, June 15-17, 2020*. (https://aisel.aisnet.org/ecis2020_rp/176).
- Kitchenham, B. A., and Charters, S. 2007. “Guidelines for Performing Systematic Literature Reviews in Software Engineering,” *Keele University and Durham University Joint Report*.
- Laanti, M. 2013. “Agile Methods in Large-Scale Software Development Organizations: Applicability and Model for Adoption,” Finland, Europe: University of Oulu.
- Laanti, M., and Kettunen, P. 2019. “SAFE Adoptions in Finland: A Survey Research,” in *Lecture Notes in Business Information Processing* (Vol. 364), Springer Verlag, pp. 81–87. (https://doi.org/10.1007/978-3-030-30126-2_10).
- Larman, C., and Vodde, B. 2008. *Scaling Lean & Agile Development Thinking and Organizational Tools for Large-Scale Scrum*, (1. Edition.), Addison-Wesley Professional.
- Livermore, J. A. 2008. “Factors That Significantly Impact the Implementation of an Agile Software Development Methodology,” *Journal of Software* (3:4), pp. 31–36. (<https://doi.org/10.4304/jsw.3.4.31-36>).
- Nickerson, R. C., Varshney, U., and Muntermann, J. 2013. “A Method for Taxonomy Development and Its Application in Information Systems,” *European Journal of Information Systems* (22:3), pp. 336–359.
- Paasivaara, M. 2017. “Adopting SAFE to Scale Agile in a Globally Distributed Organization,” in *Proceedings - 2017 IEEE 12th International Conference on Global Software Engineering, ICGSE 2017*, Institute of Electrical and Electronics Engineers Inc., July 12, pp. 36–40.

- (<https://doi.org/10.1109/ICGSE.2017.15>).
- Paasivaara, M., Behm, B., Lassenius, C., and Hallikainen, M. 2018. "Large-Scale Agile Transformation at Ericsson: A Case Study," *Empirical Software Engineering* (23:5), Springer New York LLC, pp. 2550–2596. (<https://doi.org/10.1007/s10664-017-9555-8>).
- Paasivaara, M., and Lassenius, C. 2016. "Scaling Scrum in a Large Globally Distributed Organization: A Case Study," in *Proceedings - 11th IEEE International Conference on Global Software Engineering, ICGSE 2016*, Institute of Electrical and Electronics Engineers Inc., September 26, pp. 74–83. (<https://doi.org/10.1109/ICGSE.2016.34>).
- Pries-Heje, J., and Krohn, M. M. 2017. "The SAFe Way to the Agile Organization," in *ACM International Conference Proceeding Series* (Vol. Part F1299), Association for Computing Machinery, May 22. (<https://doi.org/10.1145/3120459.3120478>).
- Puppet Labs. 2017. "2017 State of DevOps Report," *PuppetLabs*, pp. 1–55. (<https://puppetlabs.com/solutions/devops/>).
- Putta, A. 2018. "Scaling Agile Software Development to Large and Globally Distributed Large-Scale Organizations," in *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, May 27, pp. 141–144. (<https://doi.org/10.1145/3196369.3196386>).
- Putta, A., Paasivaara, M., and Lassenius, C. 2018. "Adopting Scaled Agile Framework (SAFe): A Multivocal Literature Review," in *ACM International Conference Proceeding Series* (Vol. Part F1477), Association for Computing Machinery. (<https://doi.org/10.1145/3234152.3234164>).
- Rafferty, A. E., and Griffin, M. A. 2004. "Dimensions of Transformational Leadership: Conceptual and Empirical Extensions," *Leadership Quarterly* (15:3), pp. 329–354. (<https://doi.org/10.1016/j.leaqua.2004.02.009>).
- Ries, E. 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Crown Business.
- Rolland, K. H. 2016. "Scaling across Knowledge Boundaries: A Case Study of a Large-Scale Agile Software Development Project," in *ACM International Conference Proceeding Series* (Vol. 24-May-201), Association for Computing Machinery, May 24. (<https://doi.org/10.1145/2962695.2962700>).
- Schuch, F., Gerster, D., Hein, D., and Benlian, A. 2020. "Implementing Scaled-Agile Frameworks at Non-Digital Born Companies - A Multiple Case Study," *Hawaii International Conference on System Sciences 2020 (HICSS-53)*. (https://aisel.aisnet.org/hicss-53/in/digital_transformation/6).
- Smart, J. 2018. "To Transform to Have Agility, Dont Do a Capital A, Capital T Agile Transformation," *IEEE Software* (35:6), pp. 56–60. (<https://doi.org/10.1109/MS.2018.4321245>).
- Taribuka, D., and Amrit, C. 2020. "Agile in the Banking Industry: Exploring Multiple Levels of Agile Transformation Process Facilitators and Challenges From a People Perspective," *Proceedings of the 29th European Conference on Information Systems, ECIS 2020*, p. 16.
- Uludağ, Ö., Kleehaus, M., Dreyman, N., Kabelin, C., and Matthes, F. 2019. "Investigating the Adoption and Application of Large-Scale Scrum at a German Automobile Manufacturer," *Global Software Engineering*, International Conference on Software Engineering, USA: IEEE Press, pp. 22–29. (<https://doi.org/10.1109/ICGSE.2019.00-11>).
- Uludağ, Ö., Nägele, S., and Hauder, M. 2019. "Establishing Architecture Guidelines in Large-Scale Agile Development Through Institutional Pressures: A Single-Case Study," *AMCIS 2019 Proceedings*. (https://aisel.aisnet.org/amcis2019/data_agility/data_agility/3).
- Uludağ, Ö., Philipp, P., Putta, A., Paasivaara, M., Lassenius, C., and Matthes, F. 2020. "Revealing the State-of-the-Art of Large-Scale Agile Development: A Systematic Mapping Study," *Information and Software Technology*. (<https://arxiv.org/ftp/arxiv/papers/2007/2007.05578.pdf>).
- Uludağ, Ö., Proper, H. A., and Matthes, F. 2019. "Investigating the Establishment of Architecture Principles for Supporting Large-Scale Agile Transformations," in *2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC)*, Institute of Electrical and Electronics Engineers Inc., October 1, pp. 41–50. (<https://doi.org/10.1109/EDOC.2019.00015>).
- Vaidya, A. 2014. "Does DAD Know Best, Is It Better to Do LeSS or Just Be SAFe? Adapting Scaling Agile Practices into the Enterprise," *Thirty-Second Annual Pacific Northwest Software Quality Conference*, pp. 1–18.
- Webster, J., and Watson, R. T. 2002. "Analyzing the Past to Prepare for the Future: Writing a Literature Review.," *MIS Quarterly* (26:2), xiii–xxiii. (<https://doi.org/10.1.1.104.6570>).
- Wenger, E., McDermott, R., and Snyder, W. M. 2002. *Cultivating Communities of Practice*, Harvard Business Review Press.