Scaling Agile Company Wide: The Organizational Challenge of Combining Agile Scaling Frameworks and Enterprise Architecture in Service Companies

—Robert M. van Wessel
Department of Technology and Operations Management, Rotterdam School of Management, Erasmus University (RSM), 3062 PA Rotterdam, The Netherlands

ICTU Foundation, 2595 AN The Hague, The Netherlands
(E-mail: rwessel@rsm.nl)

—Philip Kroon
Amsterdam, The Netherlands

—Henk J. de Vries
Department of Technology and Operations Management, Rotterdam School of Management, Erasmus University (RSM), 3062 PA Rotterdam, The Netherlands

Department of Values, Technology and Innovation, Faculty of Technology, Policy and Management, Delft University of Technology, 2628 CD Delft, The Netherlands

(Corresponding author: Robert M. van Wessel.)

The review of this article was arranged by Department Editor Qingyun Zhu.

IEEE DOI 10.1109/EMR.2023.3277128

***Abstract*—*Many organizations have introduced agile ways of working to channel diverse stakeholder requirements in projects interactively and iteratively. Embracing large-scale adoption of agile frameworks for the entire organization seems like a logical next step. However, in practice, seamlessly integrating what has been developed agilely at the project level with processes and applications at the enterprise level is complex. Can the seemingly contradictory approach of enterprise architecture counter this issue? Based on the five case studies dealing with IT-based projects in both the private and public sectors, we demonstrate that the two concepts can be combined, allowing us to reap the benefits of both.*

Key words: Agile methods, agile scaling frameworks (ASFs), collaborations in technology management, enterprise architecture (EA), implementation methods, new service development, organizational change, project management, software process management

I. Introduction

Many organizations have discovered agile methods and are transforming to an agile way of working or have been actively using these methods for several years [1], [2], [3], [4], [5]. We also see a trend of embracing the large-scale adoption of agile frameworks across the organization. However, the large-scale application of Agile is a challenge [6]. Organizations must deal with many internal and external stakeholders with different interests, a diversity of projects, cost overruns, and the complexity of legacy systems with their interdependencies. Due to the popularity of the agile way of working, the enterprise architecture (EA) approach has faded into the background. Architectural design, as a part of an agile approach known as "emerging architecture," is effective at the project level but poses challenges when the developed services must connect seamlessly at the enterprise level. This article aims to answer the following research question: *Can the organization-wide* application of Agile be combined with EA in such a way that the organization benefits from both the flexibility offered by Agile and the consistency and long-term focus offered by EA? We studied five case dealing with software development projects in both the public and private sectors. The results of these case studies were subsequently discussed in a round table session.

II. Background

Software developers have used iterative models since the start of this millennium inspired by the Agile

Manifesto [7]. They have experienced the benefits of iterative development over the linearity of the traditional waterfall model. Scrum is now the most popular method [8], [9]. Furthermore, many agile methods have been introduced that go beyond software development, including manufacturing, project management, and management in general [4], [5].

Because of these successful applications at the project level, large-scale adoption of agile frameworks [agile scaling frameworks (ASFs)] across the organization is on the rise [6]. But this turns out to be challenging. The agile way of working at the project level quickly results in deliverables meeting business requirements. However, results from various agile teams within and across projects show coordination challenges regarding business and technical requirements of the individual deliverables and their integration and testing [10]. The reason is that agile methods pay little or no attention to architectural design for information systems [11], [12]. Therefore, in large-scale agile transformations, the role of architecture is often overlooked, resulting in agile teams that struggle due to the lack of suitable architectures [13].

If an organization has to integrate separate software deliverables, the limitations of agile working methods become apparent. For example, it may turn out that the technology chosen deviates from what is used throughout the organization, that two project teams have implemented standards slightly differently or that different definitions are used for the same object. It is precisely in these kinds of issues that the added value of EA becomes apparent. The primary goal of EA is to guide the coherent design and implementation of information systems by providing oversight and insight, thereby facilitating technology decisions and direction for businesses and IT stakeholders [14], [15], [16].

Often, proponents of an agile approach suffer from rigid thinking. Some agile frameworks even state that you should stay far from EA. Agile principle 11 of the Agile Manifesto [7] "*The best architectures, requirements, and designs emerge from self-organizing teams*" points in that direction. EA proponents can be dogmatic as well. In the past 20 years, enterprise architects have often failed to pay enough attention to the dynamics within organizations and have adopted the idea that one should conform to the ideal and theoretical image laid down in the EA of the organization. They spent much time making architecture visions and target architectures, but when the plans were ready, the situation had already changed. Furthermore, EA thinking tends to be rational, while people in organizations are not always willing or able to make rational choices.

A systematic mapping study into large-scale agile development yielded significant knowledge of ASFs [17]. It also revealed issues about the combination of large-scale agile transformation and EA that relate to:

1) the role and collaboration of enterprise architects with agile teams;
2) coordination methods within and between agile teams;
3) balancing decision making between agile teams and architects;
4) balancing emerging and intentional architectures;
5) driving large-scale agile transformations and managing technical debt.

In summary, the original idea of scrum [7], designed for small teams and with timeboxed iterations called "sprints" typically lasting two weeks [18], evolved into two main directions: First, hybrid methods, which combine traditional and agile development approaches [19], [20], [21], and second, scaling agile methods [6] that facilitate large-scale agile transformation in multiteam settings [22]. Next to these developments, EAs traditional centralized and compliance-checking role [14] shifts toward a supporting role for agile teams and business stakeholders, focusing on translating abstract architecture concepts to concrete measures and implications [13]. However, the organization-wide combination of Agile and EA is poorly understood [23]. There are few empirical insights in the academic literature about the role of EA in agile contexts [24], [25], including the role of the enterprise architect in such settings [2], [6], [26].

III. Research Method

We explore combinations of EA and Agile and use insights from various sources. Our basis is a longitudinal case study on the impact of implementing ASFs enterprise wide [27]. We studied the role, governance, and processes of EA in three case studies conducted at large service organizations in the private sector. We also examined in two other case studies how a governmental project organization uses an ASF and takes architectural considerations into account when developing software. We used a case study protocol comprising general rules, procedures, and a questionnaire. The questionnaire contained open-ended questions on ASF company-specific features, scope of and experiences with the implementation, whether stage-gate elements were retained, and what type of agile governance was applied. Each semistructured interview lasted approximately 80 min was audio recorded and transcribed. Transcripts were sent to respondents for validation and approval. Subsequently, we imported the

transcribed text files into the qualitative data analysis software ATLAS.TI and coded the results. In the cross-case analysis, we examined the similarities and differences among the cases. We identified, documented, and presented the results to the interviewees to allow for corrections and additions. Finally, the findings from these five cases were presented in a roundtable session with experts in this field from an international consultancy firm. They have first-hand EA-related work experience at organizations that have adopted an ASF. This roundtable session provided even more insights into the effects of large-scale agile transformations on the organization of EA.

IV. Cases

Our five case studies were carried out in four organizations in various settings. A summary of the organizations and their key characteristics is provided in Table 1.

The three service *organizations in the private sector* combined Agile and EA in large-scale agile transformations. Company A is a large telecom operator providing digital cable television, Internet, and telephone services to residential and commercial customers (case A). Companies B and C are multinational financial services' companies (cases B and C). The scope of the ASF covered both their primary and secondary processes. The research examined the transformation phase (interviews during April and May 2018) and the application phase (interviews from December 2020 to February 2021). All three organizations had the following objectives for this transformation:

1) a shorter time-to-market of new products and services;
2) more flexibility in meeting customer requirements;
3) lower development costs;
4) a stronger competitive position.

Company A and B used a stepwise approach, whereas company C switched with a big bang. The transformation had a major impact on the work of the architects, but EAs roles and processes remained unchanged or were abolished altogether. Attention to EA dropped sharply in the transformation phase but picked up again later because it turned out that the required attention to long-term and organization-wide perspectives was lacking.

Positioning enterprise architects within the scaled Agile context was difficult. In company A, the transformation phase was hampered by adhering to the existing EA practices (they used TOGAF, a well-known EA method and framework). The transformation in company B led to significant staff reductions, the EA governance was dropped, and architects struggled to ensure alignment between the newly formed agile teams. In company C, the ASF did not address the role of architecture, so EA was neglected company wide, resulting in significant difficulties in redefining its EA needs.

These examples show that implementing an ASF led to difficulties in positioning EA in employees' new agile way of working.

All three agile frameworks chosen by the organizations (SAFe, LeSS, and the Spotify model, respectively) have their advantages and disadvantages. Agile scaling methods do not seem adequate to support the transformation of the existing EA practices to an Agile EA combination company wide. Even SAFe, which contains EA elements, appears to be insufficiently fit for this purpose.

During the ASF application phase, all three companies moved away from their initial ASF choice and combined it with elements from other ASFs. Company A combined SAFe with Spotify, company B added Spotify and SAFe elements to its LeSS implementation, and company C added SAFe elements to its Spotify implementation. This resulted in better cooperation between teams and in meeting architecture requirements, such as ensuring that interfaces built by different Agile teams conformed to agreed standards.

In company A, quarterly business reviews (QBRs) proved an effective governance mechanism. The company expected architects to support tribes (a collective of multiple agile teams) and program increment events. Solution architects worked hands-on with agile teams to ensure the architecture roadmap was clear, to remove impediments, and monitor implementations. In company B, communication between tribes also improved by introducing QBRs. Dependencies were better managed because architects identified them and provided suggestions for

Table 1. Key Characteristics of the Case Organizations.

Case organization	Industry sector	Size	Transformation approach	ASF used	Main reasons for implementation
Case A	Telecom	Large	Stepwise approach	SAFe + Spotify	Derisking large programs; staying competitive
Case B	Financial services	Large	Stepwise approach	Less + Spotify&SAFe	Aligning departments in the organization; increasing agility
Case C	Financial services	Large	Big bang approach	Spotify + SAFe	Reducing time-to-market and overall costs
Cases D&E	Public sector	Medium	-	Spotify + SAFe	Improving alignment and coordination

alignment. In company C, improved lead times and anticipation of new insights resulted in better prioritization of business features and incremental delivery of business functionalities. However, due to a lack of attention to cultural change, the agile mindset was not internalized.

A similar trend emerged in company D, the *public sector organization*. We studied two cases related to software development projects. The first project (case D) concerned social security services (data collection: April 2022). Around 80 people worked on this project. Next to general project staff, such as project secretaries and technical support staff, the project comprised six agile teams with, on average, four developers, two testers, a story owner, a product owner, a solution architect, and a scrum master. The lead developer recognized the problems of combining Agile and EA. In 2018, the team started with the Spotify model mainly because employees had experience with it. After about a year, SAFe elements were added as the team felt that Spotify did not sufficiently address the architecture requirements. This combination communicated the architecture vision and provided the rules and development patterns for the required software architecture. These topics were regularly promoted to agile teams and were also tested. In this way, the solution architects were involved in the developments, and it was possible to share their architecture guidelines across teams. The lead developer picked up signals about problems and ambiguity from the agile teams, gave direction, and presented options for tackling issues in weekly refinement sessions.

The second project (case E) (data collection: June and November 2022) concerned an integrated system of travel documents. Solution architects played an important role in refinements and retrospectives and participated in daily scrum meetings, especially during the project start-up phase. They ensured that user stories contributed to the functional scope defined in the architecture and advised on architectural changes. Products and services were placed in the perspective of the organization's overall service provision and teams adhered to the agreed standards. In this way, they helped to avoid the duplication of work.

Company D implemented presprint activities, also known as "sprint zero." This preparatory phase involves setting preconditions, preparing technical facilities, recruiting the development team, completing mandatory documentation, and mitigating risks to enable a smoother development process when the sprints start. Documentation includes architecture and design documents (software, infrastructure), functional and nonfunctional requirements, guidelines, and required standards. These activities enhance cross-team collaboration and involve architects that can oversee EA across agile teams at an early stage.

Based on these five case studies, complemented with feedback from the round table session, we propose how these two seemingly contradictory worlds of Agile and EA can be effectively combined in large-scale agile transformations. These are novel insights into coordination mechanisms between architects and agile teams, balancing emerging and target architectures.

V. Lessons Learned

In our case organizations, each with a high degree of complexity, none of the three ASFs provided sufficient direction to guide the transformation of the existing EA practices to an Agile EA combination company wide. SAFe requires strong leadership to ensure that the framework is aligned with all levels of an organization, including the required EA governance. LeSS works well in organizations that develop services for a single customer, and the Spotify model is best suited for organizations with a small service set. Moreover, the latter two frameworks disregard EA. As a result, management resorted to reintroducing the traditional ways of steering and giving explicit direction through new integrator roles (the private sector companies) or software delivery managers (the public sector organization) alongside product owners to ensure proper end-to-end solutions and integral service delivery. In addition, regular cross-team planning sessions were implemented. Waterfall-like deliverables support agile teams included architecture vision, architecture roadmaps, intentional architectures, and implementation guidelines. Enterprise architects communicated the architectural vision, target architectures, and roadmaps and ensured that the priorities were met. Solutions architects were active and supportive in guiding agile teams through roadmaps and guidelines [26], balancing emerging and intentional architectures. Indeed, such adjustments paid off with more consistent service development and delivery for customer services that seamlessly interacted at the enterprise level. This resulted in a so-called Agile EA Sweet Spot (see Figure 1)—a balance between an agile team's self-governance and EA governance.

If agile teams are indifferent to self-governance, they will probably also be indifferent to architecture. If they only focus on architecture without EA guidance, deliverables with fragmented architectures will not interoperate adequately. If enterprise architects push their EA too much, they will be bypassed by the agile teams, making EA redundant. And if enterprise

architects adopt a policing role, conflicts with the agile teams will occur. In the Agile EA Sweet Spot, solution architects balance emerging and intentional architecture by actively participating in agile teams. In their supporting role, they can guide and advise teams and use the traditional EA resources, including architecture vision, target architectures, roadmaps, and implementation guidelines.

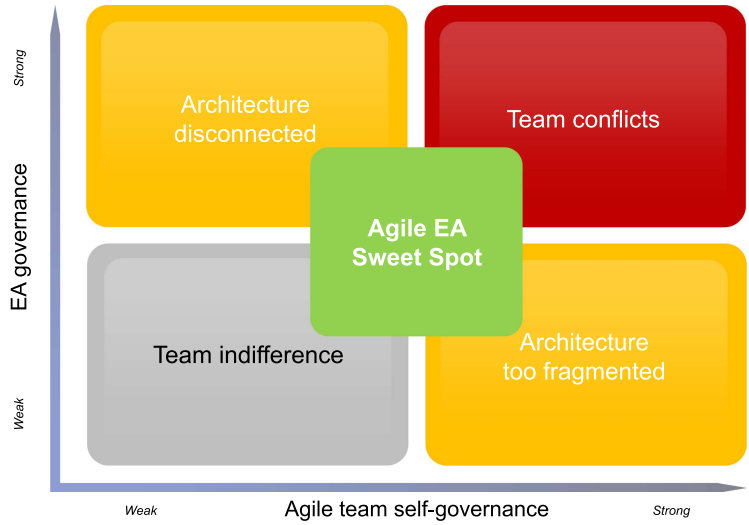

Figure 1. Agile EA sweet spot.

Table 2. EA and Agile Perspectives.

Recommendations for EA related to Agile
Establish architecture governance that ensures the link between EA and solution architecture.
Reduce the design stress of agile teams by framing their design freedom.
Allow integrator roles and processes to oversee EA across teams and provide top-down guidance to facilitate cross-team EA coordination and balance team autonomy.
Leverage the traditional EA resources to support agile teams. Communicate the organization's architectural vision, target architectures, and roadmaps and ensure that priorities are followed. Start this already during presprint activities.
Promote the EA way of thinking through development patterns, reuse of components, uniformity of methods, and solutions across agile teams.
Let solution architects ensure alignment between teams, especially if one team's output is the other's input. They exchange knowledge, monitor cohesion between teams, and provide guidance in cross-team planning sessions that are scheduled regularly.
Provide training and coaching toward the agile mindset to ensure that the organization does not revert to previous habits. This also holds true for its management.
Allow the chief architect to play a key role in aligning enterprise architects. Ensure the right work attitude in the EA team, so cooperation between enterprise and solution architects is a matter of course.
Recommendations for Agile related to EA
Widen the perspective of agile team members so that they look beyond their current project. Ensure a product still fits into the bigger picture. Advise on possible architecture changes.
Stimulate solution architects to participate in agile teams as genuine members with an active and supportive role (not as a "police officer"). They attend refinements, retrospectives, and occasionally daily scrums.
Let solution architects translate architecture principles and frameworks into concrete technical implications for team members.
Let solution architects assist with roadmaps and implementation guidelines and to facilitate the use of standards, standard components, and development patterns.
Let solution architects assist product owners in detailing "user stories."
Let solution architects identify issues and potential obstacles (impediments) and act constructively as sparring partners to solve these.
Allow solution architects to check whether "user stories" contribute sufficiently to the project scope and adhere to the EA.
Allow solution architects to form a separate team (a chapter of architects—Spotify).

VI. Managerial Implications

How can Agile be combined with EA organization wide? EA facilitates the autonomy of agile teams and ensures maximum independence among teams so that they can operate autonomously, each developing their own products and services. Such a split up of teams minimizes cross-team EA coordination, minimizes coupling between teams, and maximizes cohesion of business capabilities. Thanks to this maximum independence among teams, they can operate autonomously as much as possible.

Our case companies showed that both an incremental and a big bang transition are feasible. With a stepwise approach, an organization is typically more in control as it raises awareness and understanding, and teams can learn from earlier implementations and subsequent increments. A big bang transition process would be more appropriate in small and medium-sized enterprises (SMEs), as these organizations do not need that many increments and adjustments can easily be made when necessary. During the transition phase, the original ASF of choice can be complemented with elements from other ASFs resulting in a hybrid ASF. For example, company A initially choses SAFe and added elements of Spotify, such as tribes and guilds, that complemented their SAFe-based framework. Table 2 summarizes good practices from the five case studies.

In addition, the following five tenets should be considered when organizations apply Agile

organization wide and reinforce it with EA.

1) *Choose your ASF based on the specific organizational needs and combine frameworks if required*: Each ASF has specific advantages and disadvantages. These should match your organizational objectives and possibilities (i.e., maturity, organizational policies, and politics). Consider combinations (hybrid ASFs). Currently available ASFs do not provide sufficient guidance on EA.
2) *Redefine EA and accompanying roles and processes prior to transformation*: Ensure that EA and associated roles and processes change from an abstract and traditional hierarchical approach to an active and supportive one that adds value to the agile teams before a large-scale transformation. Use EA to ensure maximum independence among agile teams so that they can operate as autonomously as possible.
3) *Preserve some stage-gate elements of EA*: Retain some traditional elements of EA, such as target architectures, roadmaps, and implementation guidelines (including the standards to be used) to counterbalance a rigid agile implementation. Start familiarizing teams with these artifacts during presprint activities.
4) *Facilitate cross-team EA coordination*: Allow for cross-team EA coordination, balance team autonomy with top-down guidance, and ensure continuous attention to maintain the organizational architecture vision across agile teams in close collaboration with all stakeholders.
5) *Employ active and enabling solution architects:* Assist agile team members in translating architecture principles and frameworks into concrete implications and development patterns. Support the use of standards and standard components and help refine "user stories" and test these against the agreed-upon EA.

VII. Conclusion

Instead of pushing EA aside and perceiving an agile scaling method as a silver bullet, we recommend organizations combining certain elements of these two seemingly conflicting approaches. This allows one to remain Agile (e.g., interactive and iterative working) while ensuring that organization-wide interests are met (e.g., communicating development patterns, stimulating reuse, and ensuring uniformity of working methods across agile teams). All organizations in such a situation are challenged to seriously consider an agile transformation in which EA is included in the agile governance. Organizations can empower their agile teams and enterprise and solution architects by combining Agile and EA, making the best out of both worlds, thus contributing to product and service innovations.

References

[1] M. West and D. Norton, "Market guide for enterprise agile frameworks," *Gartner*, 2017, Accessed: Jan. 25, 2023. [Online]. Available: https://www.gartner.com/document/3762264
[2] D. K. Rigby, J. Sutherland, and A. Noble, "Agile at scale: How to go from a few teams to hundreds," *Harvard Bus. Rev.*, vol. 96, no. 3, pp. 88–96, 2018.
[3] T. Dingsùyr and N. B. Moe, "Towards principles of large-scale agile development," in *Proc. Int. Conf. Agile Softw. Develop.*, 2014, pp. 1–8.
[4] T. Dingsùyr, N. B. Moe, T. E. Fñgri, and E. A. Seim, "Exploring software development at the very large-scale: A revelatory case study and research agenda for agile method adaptation," *Empirical Softw. Eng.*, vol. 23, no. 1, pp. 490–520, 2018.
[5] Z. Zhang and H. Sharifi, "Towards theory building in agile manufacturing strategy—A taxonomical approach," *IEEE Trans. Eng. Manage.*, vol. 54, no. 2, pp. 351–370, May 2007.
[6] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *J. Syst. Softw.*, vol. 119, pp. 87–108, 2016.
[7] K. Beck et al., "Manifesto for agile software development," 2001, Accessed: Jan. 25, 2023. [Online]. Available: https://agilemanifesto.org/
[8] J. Werewka and A. Spiechowicz, "Enterprise architecture approach to SCRUM processes, sprint retrospective example," in *Proc. IEEE Federated Conf. Comput. Sci. Inf. Syst.*, 2017, pp. 1221–1228.

[9] Digital.AI, “16th state of agile report,” 2023, Accessed: Jan. 25, 2023. [Online]. Available: https://digital.ai/resource-center/analyst-reports/state-of-agile-report/
[10] M. Berntzen, V. Stray, and N. B. Moe, “Coordination strategies: Managing inter-team coordination challenges in large-scale agile,” in *Proc. Int. Conf. Agile Softw. Develop.*, 2021, pp. 140–156.
[11] K. Petersen and C. Wohlin, “A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case,” *J. Syst. Softw.*, vol. 82, no. 9, pp. 1479–1490, 2009.
[12] C. Yang, P. Liang, and P. Avgeriou, “A systematic mapping study on the combination of software architecture and agile development,” *J. Syst. Softw.*, vol. 111, pp. 157–184, 2016.
[13] R. Duijs, P. Ravesteyn, and M. van Steenbergen, “Adaptation of enterprise architecture efforts to an agile environment,” in *Proc. Bled eConf.*, 2018, pp. 389–400.
[14] J. W. Ross, P. Weill, and D. Robertson, *Enterprise Architecture as Strategy: Creating a Foundation for Business Execution*. Cambridge, MA, USA: Harvard Bus. School Press, 2006.
[15] W. F. Boh and D. Yellin, “Using enterprise architecture standards in managing information technology,” *J. Manage. Inf. Syst.*, vol. 23, no. 3, pp. 163–207, 2006.
[16] T. Tamm, P. B. Seddon, G. Shanks, and P. Reynolds, “How does enterprise architecture add value to organisations?,” *Commun. Assoc. Inf. Syst.*, vol. 28, 2011, Art. no. 10.
[17] Ö. Uludağ, P. Philipp, A. Putta, M. Paasivaara, C. Lassenius, and F. Matthes, “Revealing the state of the art in large-scale agile development research: A systematic mapping study,” *J. Syst. Softw.*, vol. 194, 2022, Art. no. 111473.
[18] T. Dybå and T. Dingsùyr, “Empirical studies of agile software development: A systematic review,” *Inf. Softw. Technol.*, vol. 50, no. 9/10, pp. 833–859, 2008.
[19] G. van Waardenburg and H. van Vliet, “When agile meets the enterprise,” *Inf. Softw. Technol.*, vol. 55, no. 12, pp. 2154–2171, 2013.
[20] G. Theocharis, M. Kuhrmann, J. Münch, and P. Diebold, “Is water-scrum-fall reality? On the use of agile and traditional development practices,” in *Proc. Int. Conf. Product-Focused Softw. Process Improvement*, 2015, pp. 149–166.
[21] G. Marzi, F. Ciampi, D. Dalli, and M. Dabic, “New product development during the last ten years: The ongoing debate and future avenues,” *IEEE Trans. Eng. Manage.*, vol. 68, no. 1, pp. 330–344, Feb. 2021.
[22] C. Fuchs and T. Hess, “Becoming agile in the digital transformation: The process of a large-scale agile transformation” in *Proc. 39th Int. Conf. Inf. Syst.*, 2018, pp. 1–17.
[23] M. Hauder, S. Roth, C. Schulz, and F. Matthes, “Agile enterprise architecture management: An analysis on the application of agile principles,” in *Proc. 4th Int. Symp. Bus. Modeling Softw. Design*, 2014, pp. 38–46.
[24] Ö. Uludağ, H. A. Proper, and F. Matthes, “Investigating the establishment of architecture principles for supporting large-scale agile transformations,” in *Proc. IEEE 23rd Int. Enterprise Distrib. Object Comput. Conf.*, 2019, pp. 41–50.
[25] B. Mucambe, A. P. Tereso, J. M. P. Faria, and T. Mateus, “Large-scale agile frameworks: Dealing with interdependences,” in *Proc. 33rd Int. Bus. Inf. Manag. Assoc. Conf.*, 2019, pp. 3109–3119.
[26] Ö. Uludağ, M. Kleehaus, X. Xu, and F. Matthes, “Investigating the role of architects in scaling agile frameworks,” in *Proc. IEEE 21st Int. Enterprise Distrib. Object Comput. Conf.*, 2017, pp. 123–132.
[27] R. M. van Wessel, P. Kroon, and H. J. de Vries, “Scaling agile company-wide: The organizational challenge of combining agile-scaling frameworks and enterprise architecture in service companies,” *IEEE Trans. Eng. Manage.*, vol. 69, no. 6, pp. 3489–3502, Dec. 2022.

Robert M. van Wessel received the master's degree in electrical engineering from Twente University, Enschede, The Netherlands, in 1992, and the Ph.D. degree in business administration from the Department of Information Systems and Management, Tilburg University, Tilburg, The Netherlands, in 2008. He is the Lecturer and Thesis Coach with the Rotterdam School of Management, Erasmus University, Rotterdam, The Netherlands. His research interests relate to the interaction of business and information technology, in particular agile methods, enterprise architecture, information and data management, information security and privacy, and standardization. He works as a Strategic Advisor in the field of digital public services and transformation with the government foundation ICTU, an independent consultancy and project organization within the Dutch government, and is the founder of Apexis. He was a recipient of the "Aart Bosman" Award 2008–2009 as best Information Management doctoral dissertation in the Benelux.

Philip Kroon received the bachelor's degree in economics from Utrecht University, Utrecht, The Netherlands, in 2016, and the master's degree in business information management from the Rotterdam School of Management, Rotterdam, The Netherlands, in 2018, where he graduated *cum laude*. His academic research interests relate to the adoption of agile scaling frameworks in organizations and their impact on the traditional enterprise architecture capability. In his professional work, he advises multinationals on IT M&A-related challenges (e.g., system integration, data migration, and decommissioning) and guide clients to realize their desired agile business transformations as a SAFe Program Consultant.

Henk J. de Vries received the graduate degree in geodetic engineering from the Delft University of Technology, Delft, The Netherlands, in 1982, and the Ph.D. degree in business administration from Erasmus University Rotterdam, Rotterdam, The Netherlands, in 1999. From 1984 to 2003, he served The Netherlands Standardization Institute NEN in different roles. Since 1994, he has been an Assistant Professor and an Associate Professor with Innovation Management Department, Rotterdam School of Management (RSM), Erasmus University, Rotterdam, The Netherlands. In 2018, he was appointed as a Professor of standardization management with RSM, Department of Technology and Operations Management, and Section Innovation Management. He is a Visiting Professor with the Faculty of Technology, Policy and Management, Delft University of Technology. He studies standardization from a business point of view and is the author or coauthor of more than 400 publications, including 20 books. From 2010 to 2021, he was a President of the European Academy for Standardization EURAS. In 2009, the International Organization for Standardization awarded his education on standardization as best in the world. The majority of his Ph.D. students received a national or international best dissertation award.