Scaling Agile Company-Wide

The Organizational Challenge of Combining Agile-Scaling Frameworks and Enterprise Architecture in Service Companies

Van Wessel, Robert M.; Kroon, Philip; De Vries, Henk J.

DOI
10.1109/TEM.2021.3128278

Publication date
2022

Document Version
Final published version

Published in
IEEE Transactions on Engineering Management

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

Robert M. van Wessel, Philip Kroon, and Henk J. de Vries

***Abstract*—Many organizations have embraced agile methods. Studies show a trend of large-scale application of agile frameworks company-wide. Emergent architecture design as part of an agile approach is effective at the project level but causes issues when services need to interact seamlessly at the enterprise level. Enterprise architecture (EA) can provide such coherence. Combining the scaling agile methods with EA is challenging. However, such a combination could benefit from the flexibility that agile approaches offer and provide the consistency and long-term focus that EA pursues. This article uses the longitudinal case study research to explore how organizations can effectively govern Agile and EA in large-scale agile transformations. Our case analysis shows that methods for scaling Agile do not provide sufficient guidance to properly handle the transformation from existing EA practices to an Agile EA combination company-wide. We propose how EA can be applied effectively in large-scale agile transformations despite the two seemingly conflicting approaches of Agile and EA. Based on our findings, we propose a conceptual model for future research that incorporates factors that take EA into account in the governance of agile-scaling frameworks. Our findings extend current literature on coordination mechanisms between architects and agile teams in large-scale agile transformations, thereby balancing emergent and intentional architectures.**

***Index Terms*—Agile methods, agile-scaling frameworks (ASFs), collaborations in technology management, enterprise architecture (EA), new service development, organizational change, project management, software process management.**

I. Introduction

GLOBALIZATION and digitization have impacted international service sectors [1]. Customer expectations have grown, technology has advanced, and new regulatory requirements have emerged [2]. The challenge for service organizations is to keep up with these developments and meet the requirements related to user experience, performance, privacy, and interoperability [3]. This requires seamless interaction within and between complex service systems [4] while anticipating changes [5].

To increase the speed and flexibility of the services offered to customers, many organizations have embraced agile methods. This started with small projects of interactively and iteratively building and testing software. Nowadays, agile methods are being applied in large projects, multiteam settings, and large organizations [6]–[8]. This article focuses on the latter. However, the more agile the projects are, the higher the risk that the coherence of the portfolio of services and the processes and IT systems facilitating these is at stake [9]. Traditionally, this is the field of enterprise architecture (EA) that aims to develop IT-enabled systems in a coherent way, while standards allow for interoperability between these systems [10]–[12]. Agile methods have been criticized for lack of attention to architecture [13], [14]. Ross *et al.* [10] related EA to business processes and IT infrastructures that deal with organizations' integration and standardization requirements. EA provides a long-term view of a company's processes, systems, and technologies so that individual projects can build capabilities. It also deals with interrelationships outside the immediate enterprise, thereby focusing on interoperability requisites.

Agile approaches include emergent architecture design. They are effective at the project level but cause issues when services need to interact seamlessly at the enterprise level [15]. A traditional architectural approach resembles a waterfall approach, as both

1) use sequential processes,
2) distinguish distinct roles in subsequent development phases,
3) prescribe detailed front-end plans and rely on documentation,
4) deal with clear guidelines and milestones throughout development projects [16], [9].

Combining Agile with EA is more challenging because of scaling and complexity factors [17]. Coherence problems at the enterprise level may cause tensions in terms of governance and management. Applying agile methods entails that the

Manuscript received 30 April 2021; revised 24 September 2021; accepted 27 October 2021. Date of publication 15 December 2021; date of current version 1 November 2022. Review of this manuscript was arranged by Department Editor G. Marzi. *(Corresponding author: Robert M. van Wessel.)*

Robert M. van Wessel is with the Department of Technology and Operations Management, Rotterdam School of Management, Erasmus University (RSM), 3062, PA Rotterdam, The Netherlands (e-mail: rwessel@rsm.nl; apexis@ymail.com).

Philip Kroon is with a major international consulting firm, 1057 JD, Amsterdam, The Netherlands (e-mail: philip.t.kroon@gmail.com).

Henk J. de Vries is with the Department of Technology and Operations Management, Rotterdam School of Management, Erasmus University (RSM), 3062, PA Rotterdam, The Netherlands, and also with the Department of Values, Technology and Innovation, Faculty of Technology, Policy and Management, Delft University of Technology, 2628, CD Delft, The Netherlands (e-mail: hvries@rsm.nl; h.j.devries-2@tudelft.nl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TEM.2021.3128278.

Digital Object Identifier 10.1109/TEM.2021.3128278

architecture should evolve incrementally rather than being imposed by enterprise architects. Various agile-scaling frameworks (ASFs), such as scaled agile framework (SAFe), DA2.0, and large-scale scrum (LeSS), are available to scale agile practices to large projects and enterprises [18]. However, each has limitations regarding the attention paid to EA [15]. Some studies address how to apply Agile in the context of changing requirements but a stable architecture, cf., [19]–[21]. However, how to organize the codevelopment of both paradigms has not been addressed sufficiently [22].

This article uses longitudinal case study research to explore the effects of large-scale agile transformations on the organization of EA. More specifically, it examines the impact of implementing ASFs enterprise-wide on the role and processes of EA in the large service organizations. It raises questions on how to deal with these two conflicting paradigms. Is it possible to combine the strengths of both Agile and EA? What are the effects when both the architectural and agile paradigms are used at scale in an organization? Such a combination could benefit from the flexibility that agile approaches offer, and at the same time, provide the consistency and long-term focus that EA pursues. Therefore, we intend to better understand how organizations can effectively adopt both Agile and EA in large-scale agile transformations.

The rest of this article is organized as follows. The next section provides a literature overview of Agile and EA perspectives. Section III describes our research method, and Section IV presents the results of the longitudinal case study. Section V discusses our findings and outlines the limitations, and Section VI provides our conclusions and gives implications for future research and practice.

II. Literature

Organizations that develop and offer IT-supported services face the challenge of creating a coherent set of services and systems that meet fast-changing needs. This requires both architecture and flexible development. Practitioners have developed various frameworks to address this challenge. However, there is an inherent tension between the frameworks related to EA, such as the open group architecture framework (TOGAF) and department of defense architecture framework (DoDAF) [23], and those which facilitate enterprises to become Agile, such as SAFe and Disciplined Agile [22].

Traditionally, organizations that developed IT systems to support services used waterfall-like approaches [24], [25]. These provided clear guidelines and milestones for the development trajectory. However, if the entire development cycle is planned, emergent properties and requirements revealed during the development process cannot easily be incorporated. The waterfall method "*doesn't reflect [the] iterative nature of exploratory development.*" [26, p. 96] Recognizing the potential benefits of iterative software development over the linearity of the waterfall model, developers started using other, more iterative models [27]. Iterative constructs in business date back as far as 1957 and the earliest reference to iterative development can be traced back to 1968. Yet, they only started their rise to popularity in the late 1990s [28]. Acknowledging the benefits of iterative over linear approaches laid the foundation for what is understood as "Agile" today. However, the introduction of Agile as a widely embraced concept did not surface until the new millennium [27].

Agile methods are most suitable in projects that are subject to frequently changing or unpredictable requirements and incorporate IT applications that are innovated repeatedly, resulting in high degrees of uncertainty and project risks [29], [30]. These methods also help innovators dealing with instability and uncertainty in technological and market environments [31], [32]. They are not suitable in all projects [33], and their potential "*impact on the people, the process, and the project must all be considered.*" [34, p. 8] Traditional stage-gate methods are used in larger projects with stable and predictable requirements [29], [35]. Hybrid models, which combine traditional and agile development approaches [36], [37], have been proposed as well [38]–[40]. However, the coexistence of two different ways of working causes challenges [41], especially from an organizational point of view [42].

In many cases, agile software development outperforms the waterfall method [43], [44] in terms of time, budget, and user satisfaction. The advantages of using agile teams include increased transparency and project control, mutual learning and understanding within agile teams, and easy and quick prioritization of requirements. However, agile methods do not scale well. There is a little focus on architecture in agile development which may lead to bad design decisions, and integration testing of the individual team deliverables is difficult [9]. On average, the development performance and product quality using Agile are better [45]. However, for globally distributed development teams, no significant differences were found between the agile projects and traditional software development [46], with no conclusive evidence, yet, about the superiority of Agile over Waterfall [47].

Since 2000, based on the Agile Manifesto [27], many agile methods have been introduced, including areas beyond software development, such as project management, manufacturing, and management in general [13], [48]. Scrum is the most popular method [49], [50] and has become the *de-facto* standard for managing knowledge work, especially software development. It is designed for small teams (from three to nine members) who break up their work into so-called "sprints": timeboxed iterations typically lasting two weeks [51]. The steady flow of agile-driven endeavors inspired various practitioners to experiment and innovate with the agile methodology, leading to the new challenge of large-scale agile applications. These applications occur in large multiteam settings, such as "*50 persons or more, or at least six teams.*" [41, p. 88] Scaling agile methods [41] refers to applying agile techniques in larger projects or larger companies. ASFs scale agile practices to large projects and enterprises [18]. Large-scale agile transformation is the "*switch from existing work organization concepts or development approaches to agile methods … [that]… can cover a one-time big bang transfer to agile methods in a large setting or a stepwise approach where an agile pilot is subsequently scaled up into a large setting.*" [52, p. 3] It involves critical managerial challenges and consequences for the entire organization [52]. Because of the diversity of organizations, there is no standard roadmap on how to master agile transformations. These authors categorize the challenges

TABLE I
KEY CHARACTERISTICS OF SAFE, LESS, AND SPOTIFY FRAMEWORKS[1]

Agile framework	SAFe	LeSS	Spotify model
Publication date	2011	2016	2012
Country of origin	USA	Finland	Sweden
EA addressed	Yes	No	No
Team composition	Various architecture roles and separate Scrum teams	Classical Scrum composition (Product owner, Scrum Master, and feature team)	"Tribes" consisting of "Squads" and "Chapters" "Guilds" consisting of cross-Tribe specialists
Number of configurations	4	2	1

for large-scale agile transformations. Given the focus of our study (the governance of the combination of EA and Agile), we address method-, organization-, and culture-related challenges, such as poor customization of agile methods, inappropriate organizational structures, and incompatible social structures.

Adopting ASFs brings several benefits and challenges [13], [33], [41], [43]. Benefits include easier prioritization of business requirements, better management of dependencies, enabling frequent deliveries, and enhancing employee satisfaction, motivation, and engagement. Common challenges are cross-team coordination, the resistance to change and remaining power structures, lack of management buy-in, and maintaining an agile mindset. ASFs are a starting point for an agile transformation. However, they cannot be applied as cooking recipes as those offer insufficient guidance on challenges when dealing with scaling and complexity factors. More research is needed on the benefits and challenges and understanding which practices contribute to success [33].

Three frequently applied ASFs are the SAFe, LeSS, and the Spotify model [6], [50] (see Table I). These agile frameworks have their own set of principles, inspired by the Agile Manifesto.

Only SAFe provides EA coordination mechanisms [15], and these require tailoring to an organization's specific needs [53]. The benefits of SAFe include improved collaboration and dependency management between agile teams and increased transparency in the organization. Its main challenges are team formation, change resistance, organizational politics, and establishing an agile mindset [18].

Large service companies, particularly multinationals, need an architecture approach to the digital support of their services. This is because technologies, businesses, economies, politics, and societies are fundamentally transforming and consequently radically impact their operations, requiring complex and flexible IT-enabled systems [1], [4], [55]. Following the Zachman's framework for information systems architecture [56], [12], various EA frameworks have emerged over the past decades [57]. Consequently, many definitions exist. ISO/IEC/IEEE 42010:2011 (p. 2) defines architecture as: "*fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution*." Ross *et al.* [10, p. 47] define EA as "*the organizing logic for business processes and IT infrastructure, reflecting the integration and standardization requirements of the company's operating model*." Murer and Bonati [58] positioned EA as a tool to manage complexity resulting from evolving information systems in enterprises to meet business goals, such as boosting productivity, increasing market share, or excelling in customer service. The main objective of EA is to guide coherent design and implementation by providing holistic overviews and norms through governing technology decisions and direction settings by business and IT stakeholders [10], [11], [59]. One of the most popular EA frameworks globally is TOGAF [49], [60], [61]. Aside from its iterative element, the architecture development method, TOGAF shares many waterfall-related attributes.

EA is a top–down approach. Executives decide on the strategy and communicate this down the ranks of the company [60]. In contrast, Agile is a bottom–up approach. Small teams of employees work flexibly and respond rapidly to emergent issues and properties. This apparent contradiction does not mean that these concepts cannot strengthen each other [60], [62]. EA can be an empowering tool for increased agility on its own [10], [63]. Enterprise architects play an essential role in developing digital services. They determine which components are likely to be used and should be designed for reuse and stored in a repository [10], [64]. The traditional centralized role of EA of tracking changes, approving projects, and enforcing standards [10] shifts toward a decentralized decision-making process of advising agile teams in their architectural decisions [65]–[67], with enterprise architects willing to work toward a feasible and timely solution and accepting architectural imperfections for the sake of business value [68]. Many companies have decided to incorporate various agile principles into their EA, but there is disagreement on which aspects are important. Moreover, around 38% of the organizations struggle with outdated EA results that are not fit for the combination with agile practices [60]. Therefore, enterprise architects should get involved in agile projects. However, it remains to be seen if and how enterprise architects can adapt to a new way of working [7]. While the agile development methods are commonplace in many large organizations, insight into the relations and interactions with architecture methods at the enterprise level remains elusive [14]. Prior research has attempted to link traditional EA with the Agile Manifesto [60] or integrate TOGAF and Scrum [66]. However, the effect of ASFs on EA is not well understood [60], and the academic literature that explores the role of EA in agile contexts is still very scarce [65], [69]. The gap in the academic

[1]Sources used include: [Online]. Available: https://www.scaledagileframework.com/about/; https://less.works/less/framework; https://blog.crisp.se/2012/11/14

literature relates specifically to the lack of empirical studies that cover scaling agile methods combined with architectures, their implications, and their application in non-IT organizations.

Agile principle 11 of the Agile Manifesto [27, p. 3], "*The best architectures, requirements, and designs emerge from self-organizing teams*" suggests that Agile excludes EA frameworks because of its predefined structure. This is problematic for the growing number of major companies worldwide in their large-scale agile development efforts. The challenge of large-scale agile transformations is balancing emergent and intentional architecture and communicating and coordinating development activities across teams [15]. This inherent tension may cause organizational inertia because of inconsistent, redundant, or conflicting solutions and technical debt [70]. Agile approaches tend to neglect architectural design for information systems [9], [20]. Applying Scrum to EA management [71] or using "building blocks" based on Lean and agile principles to make the EA management more Agile could be an option. This would streamline architecture processes and foster collaboration and participation [19]. Other suggestions include principle-based intentional architecting [65] and integrating architectural thinking using design principles for architecture content and organizational setup [70]. The role of architecture in large-scale agile transformations is often overlooked, and agile teams often struggle due to the lack of suitable architectures [68]. These authors found that in agile environments, the focus of the architecture function shifts from architecture compliance checking toward a supporting expert role, focusing on the quality aspects of services, and aiming to reduce technical debt.

A structured literature review identified 20 ASFs, including LeSS, SAFe, and Spotify. Only three frameworks (SAFe, DA 2.0, and EADAGP) include the role of enterprise architects [15]. Findings suggest that enterprise architects develop the organization's roadmap in ASFs, assist business stakeholders, and agile teams with governance issues, guide them through the business and technical roadmaps, and help them identify potentially reusable assets and technical debts. Such an *Agile EA governance* approach, which includes EA in the agile governance of an organization, can support the large-scale agile development activities [65], [68]. Enterprise architects can facilitate interteam architecture exchange by exerting normative and mimetic pressures to meet intentional architecture [72]. The main challenge in coordination between teams in large-scale agile transformations is their autonomy. However, they depend on each other to align their efforts on architecture, requirements, integration, testing, and deliverables [73]. Next, to maintain an overview in the large-scale setting and managing priorities, they need to deal with the architecture and technical dependencies. Communities of practice and approaches in multiteam programs that blend agile and traditional practices can support the development teams [73]. A case study describes a mixture of agile and traditional meetings to exchange knowledge and establish relationships early in a program. Later on, a gradual transition to unscheduled meetings took place [74]. Our study extends this article with the combination of Agile and EA in large-scale agile transformations. A systematic mapping study disclosed a significant body of knowledge of ASFs in the field of large-scale agile development [22]. Based on 133 publications, the study identified open research questions for ten research streams, such as communication and coordination, architecture, and autonomous teams. They also formulated research questions regarding the combination of large-scale agile transformation and EA. These questions relate to

1) the role of enterprise architects, their collaboration with agile teams, and typical challenges;
2) coordination mechanisms at the intra and interteam level;
3) balancing decision-making power between architects and agile teams;
4) balancing emergent and intentional architectures; and
5) managing technical debts and driving large-scale agile transformations.

To conclude, the literature reveals effective management practices, including criteria on organizational readiness, the degree and type of scaling required, and the organizational changes needed to implement an agile scaling framework in large-scale transformations [6], [10], [41]. However, governing such approaches in combination with EA methods causes new challenges. Researchers disagree on which aspects of EA are important in an agile context [60], how enterprise architects can adapt to an agile way of working [7], the relationships between agile and EA methods [14], and how these can strengthen each other [60], [62]. In addition, issues remain regarding balancing emergent and intentional architecture, ensuring vertical coordination between top–down governance of architecture efforts, and bottom–up autonomy of agile teams and aligning horizontal coordination between multiple agile teams [73], [75]. Other issues include managing (product and) service development activities in various settings (projects and in the entire organization) and defining the role of the enterprise architect in such venues [7], [15], [41], [65]. Therefore, as the current body of knowledge lacks these detailed insights, we formulate the following research question: "*What kind of governance is needed in organizations to combine EA and large-scale application of agile methods?*" To answer this question, we also investigate the important EA characteristics in an agile context. We distilled five constructs [76] from the literature described previously that are not explicitly defined as such (see Table II). These serve as the basis for our empirical research to further explore the impact of the application of large-scale agile transformations on the role and processes of EA in organizations.

III. Methodology

We conducted a longitudinal multiple-case study to gain more insight into the constructs and how they influence each other. This method is appropriate as there is little previous research on this topic, and it provides an in-depth overview of real-life situations and contemporary phenomena [79]. We collected data from three European service companies that carried out large-scale agile transformations. We selected large companies, with over 500 full-time equivalents (FTEs), because we expect the need to combine an agile approach with EA to be more compelling and challenging in large than in small and medium-sized enterprises (SMEs). Criteria for case selection were as follows:

1) various large companies within the service sector;
2) IT support is essential for service delivery;

TABLE II
CONSTRUCTS THAT EMERGED FROM THE LITERATURE

Construct	Description
Agile scaling framework characteristics	Descriptive features of the agile framework – e.g., its intended application (project, program, enterprise), whether EA thinking is incorporated, how teams are composed, and the number of possible configurations. (cf. [15], [17], [33]).
Scope of agile scaling framework application	The reach and range [77] of the agile framework. Reach refers to the utilization of the framework in the company (from a single department to the entire enterprise). Range refers to the functionality that is facilitated by the framework in terms of activities (e.g., in the case of a bank from mortgage services to all services delivered). (cf. [33], [50], [69]).
Attention paid to Enterprise Architecture	The level of consideration the organization shows, with a view to action, for EA practices such as processes, methods, and tools. (cf. [19], [22], [66], [67], [68]).
Waterfall retainment	The level of stage-gate methods and practices after agile framework adoption by the organization. Examples include: 1) the use of sequential processes, 2) distinct roles in subsequent development phases, 3) detailed front-end plans and extensive documentation, 4) the way of dealing with clear guidelines and milestones over the course of development projects. (cf. [36], [38], [41]).
Type of agile governance	The type of decision rights and accountability structure to encourage desirable behavior in the implementation and use of the Agile and EA in an organization (based on [78]). (cf. [15], [67], [75]).

3) the organization has transformed to a large-scale agile development or is at least well on the way; and
4) accessibility of interviewees for an extended period of time for longitudinal research purposes.

This resulted in three case companies, two of which are active in financial services and one in telecommunications. The three organizations apply different scaling agile methods. Interviews took place in April and May 2018 and from December 2020 to February 2021, so this covers a timespan of almost three years.

In 2018, we interviewed nine contractors employed by an international consultancy firm. The interviewees had performed various roles during large-scale agile transformations at their respective clients. All of them had first-hand EA-related working experience at organizations that adopted a large-scale agile approach. Their experience in many client companies allowed them to reflect on their observations better than a case company employee would be able to do. Their affiliation suggests that they had common ways of working, so differences between cases were primarily due to company characteristics and how the ASF was implemented. And in terms of the contents of their advice, these consultants provided tailor-made advice without any predefined preferences.

Our case study protocol included general rules, procedures, and a questionnaire. The questionnaire included open questions about

1) company-specific features of the agile framework,
2) scope of the implementation,
3) experiences with the agile implementation, including how the agile transformation changed EA-related roles and processes,
4) stage-gate elements that were retained,
5) type of agile governance: What kind of decisions are made by whom and how are these decisions monitored.

At the start of each interview, we ensured that all interviewed parties shared the same understanding of the specific principles of ASFs. Each semistructured interview took approximately 80 min, was audio recorded, and subsequently transcribed. We informed all interviewees of the purpose of the recordings and the anonymization of the resulting transcript. Transcripts were made immediately and sent to the respondents for validation and approval. In addition, we used documents from the companies and information from the third-party websites that discussed these transformations.

For the within-case analysis of the first interview round, we imported the transcribed text files into the qualitative data analysis software ATLAS.TI. We coded the results using an integrated approach that employs both a deductive and inductive (ground-up) development of codes [80]. It was deductive because the literature revealed core concepts we could use in preparing the questions. These also served as a basis for some of the codes that were characteristic of the answers given. It was inductive as the second category of codes emerged from the interviews and company-specific documents and publications and were not foreseen in advance. Such findings sometimes lead to additional questions in the free-format part of later semistructured interviews. We ended up with 49 codes per ASF, which we distributed over the constructs underlying our questionnaire. In the cross-case analysis, we examined the similarities and differences among the three cases. To increase the validity and reliability [79], [81], we triangulated the interview data

TABLE III
COMPANY PROFILES

Company	*ASF (t=1)*	*ASF (t=2)*	*Industry*	*Size*	*Reason for implementation*	*Transformation*
A	SAFe	SAFe + Spotify	Telecom	Large	De-risking large programs; staying competitive	Stepwise approach
B	LeSS	LeSS+ Spotify & SAFe	Financial Services	Large	Aligning departments in the organization; increasing agility	Stepwise approach
C	Spotify	Spotify + other SAFe elements	Financial Services	Large	Reducing time-to-market and overall costs	Big Bang approach

TABLE IV
HIGHLIGHTS OF THE LARGE-SCALE AGILE TRANSFORMATION PER CASE COMPANY

Period	Company	Interviewees	Distinguishing characteristics	Results
I: April and May 2018 transformation phase	A	Program Design Authority Lead (1), Program Director (2), Solution Architect (3).	Transformation processes and teams were managed tightly Business departments influenced the architecture roadmap Top-down governance approach hindered decentralized decision-making	Successful agile transformation by adopting existing EA practices Significant staff reductions Difficulties in communicating the role of enterprise architects
	B	Solution Architect (4), Program Manager (5), Assignment Manager (6).	Organization allowed loose implementation style Differences in interpretation of guidelines	Significant staff reductions, including governance positions Some agile ideas were discarded Architects struggling to ensure alignment across agile teams
	C	Business Integrator (7), Project Manager (8), IT Integrator (9).	Radical transformation: 50% of IT staff was made redundant, including architecture governance layers Waterfall characteristics still add value	The role of EA completely neglected Difficulties in redefining the form of EA needed
II: December 2020 – February 2021 Application phase	A	Program Design Authority Lead (1), Solution Architect (3).	Amalgamation of SAFe and Spotify model Additional management layers and roles were retroactively built into the ASF to safeguard the integration and monitoring of objectives of large-scale agile transformation	At the portfolio level, Quarterly Business Reviews (QBRs) were an effective governance mechanism, where priorities across all domains were determined every quarter The organization expected architects to play a supporting role in tribes and QBRs
	B	Program Manager (5), Innovation Manager (10).	The key governance roles in the ASF included a Product Owner (Business representative), an IT delivery manager (short-term view), and an Enterprise architect (long-term view) who decided what work needed to be done in agile teams	Communication between tribes greatly improved with the introduction of QBRs Dependencies were better managed: architects identified these and provided suggestions for alignment
	C	Business Integrator (7), Project Manager (8), Strategy Consultant (11).	Lack of detailed architectural governance Management resorted to more traditional ways of management by adding new integrator roles, including some EA-related roles and processes	Improved turnaround times and better anticipation of new insights, due to improved cooperation between Business & IT departments, resulting in prioritizing business features and stepwise delivery of business functionalities Agile mindset not internalized due to lack of attention to cultural change

with publicly available information from third-party websites on the transformations of the case companies. For this cross-case analysis, we imported our findings in an ATLAS.TI project file, followed by an analysis similar to the within-case analysis. Subsequently, we identified, documented, and presented all similarities and differences to the interviewees.

From December 2020 to February 2021, we interviewed seven contractors from the same consultancy firm, five of whom had also been interviewed in 2018. One of the new interviewees recently joined the consultancy firm after working at one of the case organizations for over 18 years. He was, therefore, able to provide detailed insider information. The interviews were held online and lasted approximately 60 min. We applied the same case study protocol as before but revised the questionnaire to increase the focus on the governance aspects of the ASF and EA combination [82]. We used the publicly available company project reports, presentations, and leaflets and shared company findings with the interviewees to allow for corrections and additions. Within-case differences in perception, if any, were discussed with the interviewees.

IV. RESULTS

A. Case Studies

We present the large-scale agile transformation and examine the impact of EA and its underlying processes. Table III gives an overview of the case companies, with the ASF in use at both timeframes of our longitudinal study. For the sake of anonymity, the three organizations of this case study are designated with letters A, B, and C. Initially, Company A adopted SAFe, B adopted LeSS, and C adopted the Spotify model (Period 1: transformation phase). To our surprise, all eventually transposed into a hybrid ASF solution (Period 2: application phase) as they combined the elements from SAFe and Spotify. For all three case organizations, the range of the ASF concerned all services provided. The reach was the entire enterprise, except for Company C, which was limited to its retail and wholesale departments. Table IV provides an overview of the transformation characteristics and results in the two timeframes of the longitudinal case study: the implementation phase and operational phase. Next, we describe the cases, followed by the cross-case analysis.

1) Company A: Company A is a large telecom operator, providing digital cable television, Internet, and telephone services to residential and commercial customers. It is a result of several mergers in the last decade. The main objective of the large-scale agile transformation was to remain competitive. The decision to build an agile organization was taken before the actual implementation of SAFe. Before the last merger, the two companies had chosen distinct ASFs. At the start of agile transformation, top management support was lacking. During the merger, the IT department pursued SAFe, whereas the Business division did not want to part with the Spotify model because of its appealing simplicity. The company finally chose SAFe as the leading ASF since it defined the role of architecture and provided employees with clarity through extensive documentation.

Transformation phase: The ASF scope (reach and range) covered the whole enterprise and all business and supporting activities: B2C, B2B, Finance, HR, etc. When asked what architecture-related elements were crucial for the successful implementation of SAFe, the interviewees mentioned that the case organization positioned enterprise architects in higher ranks of the organization to oversee all chains involved. The Design Authority Lead responsible for the agile transformation commented on the EA before the transformation: "*old school [traditional]* EA *does not provide in-depth advice, which makes it redundant*." Conversely, a solution architect who focuses on its solution only is ineffective as well. Therefore, enterprise architects were positioned somewhere between these extremes. Third, organization A should be capable of change. Therefore, strong leadership and a dedicated workforce were required to quickly master new processes and tools for company A to successfully transform to SAFe. The Program Director reinforced this: "*if architects are unwilling to change and adapt to a new way of working, then you [organization A] cannot succeed*."

Traditional EA-related processes were identified, and a SAFe flavor was added during the implementation project. Staff was trained in SAFe/Agile to work in this new way. Key roles and concepts from TOGAF remained unchanged under the SAFe implementation. The Design Authority Lead appointed several architects to SAFe's highest level to help operate and steer the organization. At that level, priorities were set during quarterly business review (QBR) sessions, which turned out to be an effective governance mechanism. A QBR resembles increment planning in some agile frameworks, which reflects on the past quarter's results and takes the lessons learned to guide the planning for the quarter to come. The Program Director pointed out another important change: The business departments were given more influence in shaping the architecture roadmap, so in that sense, the actual "*architecture did not change, but the processes related to practicing architecture changed significantly*."

1) Application phase: Recently, the company created a new department, the Design Authority, to direct programs and projects. The relationship between agile working and EA is most evident at program increment events where (solution and enterprise) architects are involved in discussions with tribe leads and squad members. Architecture guidance for the organization's new services at the senior/board level is required as well as more lead architects per tribe, who work hands-on with the teams (squads) to ensure that the roadmap is clear and impediments are removed. The organization expects architects to support the tribes. This is different from what they used to do. There was a misalignment between what an architect delivered (high-level plans and design documents reviews) and what the development teams needed. Therefore, the organization established two architecture roles: a) enterprise architects who report to the CIO and a tribe lead, and b) lead architects who are a part of a tribe, perform hands-on work, and monitor implementations.

The organizational structure follows the Spotify model team setup (with tribes and squads) that operates autonomously as much as possible. They use concepts and roles from SAFe to deliver large complex changes. However, many service matters concern cross-tribe issues. Consequently, safeguarding the integration and monitoring the solution is a challenge. Therefore, additional management layers and roles have been added to the ASF. As a result, the organization is not as Agile as it is supposed to be.

Overall, their agile approach enables the organization to prioritize changes and, thus, achieve greater flexibility. However, the organization has issues with integrating legacy systems into their agile way of working. Significant improvements have been made to digital channels for sales and support, resulting in increased customer satisfaction. However, the Design Authority Lead argued that these benefits could also have been achieved with other ASF hybrids.

2) Company B: Company B is a multinational financial services company. It wished to align all its departments and increase service delivery agility. Initially, they adopted LeSS. Their main reason for choosing LeSS was that their competitors had implemented other ASFs, so LeSS would differentiate them from the competition.

Transformation phase: In the first period of our investigation, Company B used LeSS for guidance rather than enforcing strict implementation. Company departments experimented semiautonomously with working large-scale agile. During the implementation, the traditional business architects were replaced by the Product Owners. This shift was in line with LeSS: enterprise architects' traditional and relatively passive role was no longer acceptable. They had to become more proactive and join forces with people who worked closer to the respective domains. The Assignment Manager was not convinced: "*LeSS may not help [company B] to properly maintain the architecture at the correct level because it will seriously compromise the added value of enterprise architecture*." He continued: "*… and the inherent simplification of EA roles due to the LeSS framework could and should be tackled.*"

When scaling agile company-wide, an increasing number of departments implemented agile projects. However, these implementations did not fit together well. Therefore, the organization decided to provide more top–down guidance and added elements from Spotify and SAFe.

Application phase: Currently, the reach and range of the agile approach cover the entire enterprise, and business-supporting activities. The key governance roles in their ASF are a Product Owner (business representative), an IT delivery manager

(short-term view), and an enterprise architect (long-term view). They are responsible for the work carried out in the squads. This trilateral cooperation already existed before the organization started with the agile transformation, and it continued to work reasonably well and led to smooth decision-making. The architecture function of the organization operates somewhat detached from the squads. It is responsible for checking whether the proposed solutions met the predefined architecture principles. The strategy consultant suggested that "*architects could take a more active role in IT delivery, because of the long-term vision they are committed to*."

The organization has applied company-wide agile methods for almost three years now, but is struggling with the small independent self-steering teams. Each tribe and squad pursues its own goals, and nobody seems to feel end-to-end responsibility. Squads are capped to around ten team members. A lot of time is spent on coordinating the squads. Management provides little guidance, as they assume agile teams to be self-managing. However, the newly designated management roles provide more direction to squads and tribes. Management support for the large-scale application of agile methods is an issue because managers often stick to existing structures.

The QBRs have improved communication between tribes, and dependencies are better managed now. Architects play an important role in quickly identifying the dependencies between tribes and indicating what should be discussed in the QBRs. The Strategy Consultant commented: "*This governance mechanism allows us to make clear choices. The QBR should be used as a checkpoint for the direction the organization is heading to. There's not much left of our waterfall method, except for these QBRs that are actually also waterfall and can be used to set priorities efficiently*."

The application of their hybrid ASF has resulted in a slightly faster delivery of new services because the squads work closely together. However, according to the Program Manager, the delivery is "*not very high and [it] should be monitored better. There is little top-down Quality Assurance.*" The results include big differences between squads and no key performance indicator (KPIs) for the delivery speed. In the design phase, modifications can be made more rapidly. However, time-to-market of products and services has improved only marginally because the implementation phase has not been shortened. A positive effect of working Agile is the greater sense of responsibility in teams. Staff is more committed, and therefore changes are implemented faster. Teammates also dare to make more mistakes and learn from such mistakes and from each other. For many service managers, the large-scale agile transformation is difficult because team members openly express opinions about their managers' performance.

3) Company C: Company C is a multinational financial services company. Its main objective with the large-scale agile transformation was to reduce the time-to-market of its services and reduce IT development and maintenance costs.

1) Transformation phase: Company C implemented and enforced Spotify instantaneously across its retail division as part of a "big bang" reorganization. Agile coaches facilitated the cultural changes needed. According to the Business Integrator, the company "*let go 50% of its IT staff and added 20% new employees, resulting in an overall 30% decrease in IT-related jobs*." The IT development departments, most of the Retail organization, and some supporting functions (e.g., Legal and Compliance) were reallocated to agile teams. Later on, the wholesale division adopted the ASF as well.

Prior to the Spotify implementation, the company had an EA board with a dominant position in the organization. This governance layer was removed. The former architecture function was based on the TOGAF practices. According to the interviewed IT integrator, "*the organization was so focused on becoming fully agile, that it completely neglected the changing role of EA*." The EA processes remained the same but were not included in the agile governance. The role and responsibilities of the architects were not clearly defined, and there was no structure anymore to align the solutions for services with an architecture board. There was a chief architect in name (and on paper), but it remained unclear who was in charge of what. A global architect was responsible for an architecture description per business area.

The Spotify model provides very little implementation documentation. The company made various customizations and additions to the model to make it fit for use. Team alignment was quite successful, but cross-tribe alignment was more challenging. Furthermore, the main objectives of projects and programs were difficult to coordinate across the organization. Therefore, the company introduced the Business Integrator role to compensate for the lack of an overarching governance structure inherent to the Spotify model. Business Integrators became responsible for prioritization across tribes and timely delivery. One of them described it as "*adding some classic project management*" to the framework. Due to the lack of detailed architectural governance, the Business Integrator had to get things done by himself and worked with an enterprise architect, a domain architect, and a solution architect to provide an interim solution to work toward the target architecture.

2) Application phase: Currently, coordinating work across teams does not automatically take place in the same agile manner. Requirements are mutually exchanged, which results in issues, such as implementing technical interfaces due to the large dependencies with other agile teams. It is important to log business requirements swiftly in QBRs, otherwise product owners are unable to include them in the next round of sprints. Initially, each tribe had its Agile Coach, and Spotify "chapters" and "guilds" were active. On the one hand, some events and ceremonies to align tribes were added to this setup. On the other hand, the Business Integrator commented: "*You should not stick too rigidly to all prescribed agile ceremonies, because if they do not add value in a squad or tribe, make it tailor-made*."

Waterfall characteristics partly remained after the Spotify implementation, especially during the initial phases of new developments and significant changes. The Business Integrator further noted: "*Cases where the waterfall approach is actually more useful, are projects that focus on data exchange and interfacing. Agile is very useful for customer-facing applications, and a hybrid model with waterfall aspects is applicable for other subjects*."

The large-scale agile transformation has resulted in shorter communication lines, better coordination between business and IT departments, faster decisions to get services into production, and increased staff satisfaction. In addition, it allows certain business features to be prioritized in IT-supported services. However, one very important factor has been neglected: An agile mindset requires a culture change. Initially, agile coaches were in charge of this culture change, but they were removed from the tribes to save costs. There is a serious risk that the benefits of working Agile in company C will eventually be nullified.

B. Cross Case Analysis

Each case company implemented an ASF to improve software development for its services. At the same time, they also needed EA to ensure consistency in the service portfolio and the supporting processes and systems. The cases confirm the tension between Agile and EA, and we describe how the companies dealt with this tension. We structure the cross-case analysis using the five constructs distilled from the literature.

1) ASF characteristics: All three companies adopted a distinct ASF. Common drivers for implementation included:

1) reduced time-to-market,
2) more flexibility in meeting customer demands,
3) reduced development costs,
4) competitive position (competitors implement agile methods as well).

In the operational phase, they faced several difficulties. Subsequently, each company took elements from other ASFs and added management roles for interteam coordination to guide the teams responsible for developing new digital services.

The case studies suggest that SAFe requires strong leadership for effective deployment and to ensure that the implementation adheres to the framework throughout all layers of an organization. LeSS works well in an environment where the service is made for a single customer but does not fit well in highly complex environments. The Spotify model seems to be better suited for organizations with a relatively small service set. The frameworks also differ in their guidance on how an organization addresses internal factors, such as multiple service lines, the degree of process complexity, and governance of EA and software development.

SAFe seems to be broadly applicable, also for organizations with a high degree of complexity. In contrast, the LeSS and Spotify models seem best for organizations with one or more focused service lines. The Spotify model provides very few implementation guidelines. Our analysis suggests that organizations should assess the current environment first to make an informed choice for an ASF, including its EA, formulate their requirements, and subsequently consider ASF's advantages and disadvantages specific to the organization.

2) Scope of ASF application: In case companies A and B, ASFs cover the entire enterprise. In company C, they cover its retail and wholesale banking divisions only. In all cases, the scope includes both primary and supporting functions (B2C, B2B, Finance, HR, etc.). Agile is most effective for front-office applications and less effective for applications with fewer users and back-office applications (e.g., interfaces between systems, applications focusing on meeting regulatory requirements). In the latter case, the iterative and interactive way of working does not add much value, as requirements are often already known from the outset.

3) Attention Paid to EA: Before the agile transformation, all companies had a customized EA framework suitable to their specific needs. Two of them used TOGAF for inspiration and added elements from other EA frameworks. The large-scale agile transformation changed all architects' work, but the EA roles and processes remained unchanged. From an agile perspective, EA in its traditional form is too abstract, and the Product Owner now carries out many activities of the traditional business architect. Our cases show that this discrepancy makes EA redundant in the transformation phase. However, the case companies that use Agile company-wide need EA for consistency of the various services that increasingly are being digitized. Solution architectures cannot fill this gap, as they focus on project results only. Therefore, in the application phase, the attention paid to EA has increased. Interviewees agreed that the roles and accompanying processes of EA need to be clearly defined before initiating a large-scale agile transformation.

4) Waterfall retainment: Initially, the case companies wanted to get rid of their traditional waterfall-oriented approach. However, all companies experienced that some waterfall-related characteristics of EA remained important in an agile context. Therefore, business integrators and (enterprise and solution) architects created implementation guidelines for agile teams. Business managers and enterprise architects set up architecture vision and roadmaps, and enterprise architects draft standards and intentional architectures. Furthermore, the agile approach turns out to be more effective for client and front office applications than for applications with fewer users and back-office applications. In the latter case, the iterative and interactive way of working does not add much value, as requirements are often already known from the outset or no multidisciplinary change issues exist that need attention.

5) Type of agile governance: Agile principle 11 states that architectures should emerge from self-organizing teams [27]. This relates directly to our research question. This principle holds true, but only under the right circumstances. Agile teams need to be aligned with the organization's overarching, architectural vision and, therefore, should be supported by EA governance. The task of an enterprise architect is not to control agile teams, but to communicate the organization's architecture vision, international architectures, and roadmaps, and to ensure priorities are carried out in logical order.

Our cases show that too much team autonomy and too little direction setting by management negatively affect the effective combination of ASFs and EA. As a result, management resorted to more traditional ways by explicitly setting directions and adding new integrator roles next to the Product Owners to secure end-to-end solutions and proper service delivery. These adjustments indeed paid off due to more consistent service development and delivery for customer services to seamlessly interact at the enterprise level.

In addition, QBR sessions are important checkpoints for the direction the organization is heading. QBRs allow regular progress monitoring and facilitate priority setting and intrateam alignment of requirements. Communication between the tribes has improved, and dependencies can be better managed. Architects are essential because they can quickly identify dependencies between tribes and indicate when these should be discussed in QBR sessions. Continuous management support, training, and coaching about the agile mindset are required to ensure the organization does not fall back into previous habits.

In the transformation phase, not all companies did equally well. Company A took some time before choosing an ASF. Company B initially gave too much autonomy to individual departments to implement the ASF, resulting in misalignment, and Company C got rid of its entire EA function. In the application phase, all companies are successful, although each has specific points of attention. Company A has difficulties integrating legacy systems in the agile way of working. Company B struggles with the mindset of its middle management, who feel their role has become less important, and Company C's agile culture is at stake.

The cross-case analysis showed that these three companies had the following similarities.

1) Drivers for ASF implementation included reducing time-to-market, improving flexibility in meeting customer demands, lowering development costs, and strengthening their competitive position.
2) The scope of the ASF covered both primary and supporting functions.
3) The transformation significantly impacted the architects' work, but the EA roles and processes remained unchanged. Attention to EA dropped in the transformation phase but increased in the application phase.
4) Waterfall-like deliverables to support agile teams included architecture vision, intentional architectures, architecture roadmaps, and implementation guidelines.
5) New integrator roles, better direction setting by management, and regular cross-team planning sessions were implemented. Enterprise architects were responsible for communicating the organization's architecture vision, intentional architectures, and roadmaps and ensuring that priorities were met.

V. Discussion

Scientific researchers have picked up the need for combining large-scale Agile and EA. However, they disagree on the relationships between Agile and EA methods and how these can strengthen each other [22]. In addition, it is unclear which aspects of EA are important in an agile context and how enterprise architects can adapt to the agile way of working [15], [68]. Our longitudinal case study of three service companies addresses this research gap [22], [67]. It reveals organizational success factors for effectively adopting EA and agile methods at scale:

1) first assess the roles and accompanying processes of EA to assess their impact when working Agile;
2) leverage traditional waterfall-oriented characteristics from the organizations' EA to facilitate cross-team EA coordination and balance team autonomy with top-down guidance; and
3) adopt elements from other ASFs to amend deficiencies in current ASFs as these do not provide sufficient guidance on EA, even those that include EA considerations.

Before a large-scale agile transformation, an organization should first assess the roles and accompanying processes of EA that need to be redefined to make them fit for use in an agile context at scale. They should formulate how to make EA roles and processes effective during and after the agile organizational transformation. Although literature [7], [41] suggests checking organizational readiness and customizing the approach for agile transformations in general, our article shows that this is also the case for combining Agile and EA. EA in its traditional form does not fit with the agile way of working. Solution architecture appears to be ineffective as well because of its focus on project results. Our findings suggest that without such precautions, Agile may impact EA negatively, even with ASFs that provide EA coordination mechanisms. To the best of our knowledge, this phenomenon has never been identified in earlier studies.

Organizations should leverage traditional waterfall-oriented characteristics from their EA approach and use these to complement the chosen agile approach. Some traditional elements are still required, such as integrator roles and related processes facilitating cross-team EA coordination and balancing team autonomy with top–down guidance. In agile governance, too much team autonomy and too little direction setting by management are detrimental to the effectiveness of the combination of ASFs and EA. Moreover, management support, training, and coaching toward the agile mindset are required to ensure an organization does not fall back to previous habits. Although identified in [41], our article extends these requirements to the combination of Agile and EA. EA is essential to maintain the organization's architectural vision throughout all agile teams. Architects are expected to play an active and supporting role to provide direction across teams through roadmaps and guidelines [15], thereby balancing emerging and intentional architectures. However, contrary to the suggestion of [67], the role of EA should not just focus on technology but should also include strategic and organizational aspects.

Because of a lack of detailed architectural governance, management resorted to some traditional EA roles and processes in all three companies. Our findings confirm that companies can benefit from a combination of existing frameworks for Agile and EA [60], [62], but this depends on how these companies organize themselves [66]. ASFs do not provide sufficient guidance to properly handle the transition from existing EA practices to an Agile EA combination enterprise-wide [33], with SAFe appearing more useful than Spotify or LeSS. Case companies B and C did not initially opt for SAFe, but chose elements of SAFe later on since this agile framework provides details on how to combine Agile with EA company-wide. But in company A, SAFe was complemented with aspects of another agile framework instead (notably Spotify). This extends the conclusion in [7] and [52] to

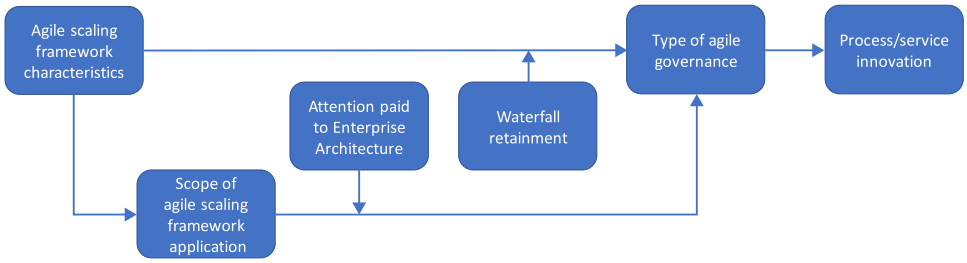

Fig. 1. Proposed conceptual model for future research.

the realm of ASF and EA that no 'best' approach for large-scale agile development and application exists.

We also examined how the ASFs affected the services offered by the case organizations, including service innovations (defined as a new service or a renewal of an existing service, which provides benefits to the organization that developed it [83]). Our case companies aimed at both service and process innovations but focused on better IT support. Most service innovations were incremental. Company A has made significant improvements in its digital channels for sales and support, resulting in increased client satisfaction. It also reduced time to market, especially in areas with straightforward service configurations. However, not all departments implemented Agile. The company needs more time to achieve service innovation for its telecom and multimedia services (e.g., Internet provision, interactive TV, and voice over Internet protocol (VoIP)). They have to adapt their core IT systems to overcome legacy issues that hinder service innovation that Tengstand *et al.* [54] also identified for SAFe. The extent to which vendors can use this new way of working is another bottleneck not mentioned in the literature, cf., [20]. In company B, the innovation capability improvements were limited to incremental changes in services. The development processes for new IT-supported services improved gradually. In Company C, the agile implementation itself was a radical process innovation. Service innovation was slightly faster, but too much development time was spent on legacy systems and meeting regulatory requirements. Only incremental changes in services were achieved, such as functional additions in mobile apps: small pieces of new functionality to increase for instance customer self-service. Each company achieved better flexibility in services development by better coordination among tribes and squads. More innovations took place in processes than in services. An explanation could be that these service companies have to abide by various laws and regulations (e.g., related to privacy and finance) that may hinder innovations. A second reason could be related to legacy systems whose agile integration is difficult [54]. Other reasons may be a lack of cultural change to transform organizational politics and to internalize an agile mindset [18], [72].

A. Limitations and Threats to Validity

Our study has some limitations. First, we applied logical generalization using three large case companies. Adding more cases could test and enrich our findings. Second, all three companies had to cope with legacy systems, hindering innovation. Other sectors might show more radical service innovations—a topic for further research. It would be interesting to examine whether our findings also hold for SMEs. Third, we only interviewed a limited number of contractors per company. Interviewing contractors employed by the same consultancy firm has the advantage of obtaining better comparability between cases, but future research could interview employees in different roles per company.

We examined whether there were any typical threats to validity in our case study research [79], [81]. We addressed internal validity (cause and effect relationship between our research process and the derived results) by establishing causal relationships through pattern matching and explanation building in our cross-case analysis using the identified codes from the ATLAS.TI project file. External validity (generalizability of the derived results) is addressed by means of analytical generalization. Construct validity (Do examined concepts represent what the researcher has in mind?) was addressed by providing descriptive texts for each construct in our questionnaire and by asking feedback on and confirmation of the transcripts. To counter research bias, we used a case study protocol and recorded, transcribed, and coded all interviews in a similar manner, and stored them in a case study database.

B. Proposed Conceptual Model

Based on our longitudinal case studies, we could further detail the constructs from our integrated approach (Table II) and get insight into relationships between them (see Fig. 1). "ASF characteristics" are our independent variable and comprises various ASF elements, such as the characteristics in Table I. The "Scope of ASF application" is a mediating variable to the "Type of agile governance." The "Attention paid to EA" and the level of "Waterfall retainment" act as mediating elements contributing to the "Type of agile governance." For the sake of future research, we have added the construct "Process/service innovation" because this shows the impact of the "Type of agile governance." Related impact variables include time-to-market and the number of ideas successfully turned into products [84].

VI. Conclusion

Companies that use Agile company-wide need EA for the consistency of the various services they offer and the processes and systems supporting those services. We considered how

organizations deal with the tension between Agile and EA in large-scale agile transformations and how this can be effectively mitigated through several governance mechanisms. Each case organization moved from a single ASF implementation, of which SAFe incorporates some EA coordination mechanisms, toward a way of working by adding elements from other ASFs. This suggests that each of these frameworks is insufficiently fit for the transition from existing EA practices to a fruitful Agile EA combination enterprise-wide, and that tailor-made solutions are required. This brings us to our research question *"What kind of governance is needed in organizations to combine EA and large-scale application of agile methods?"*

The agile teams, including the Product Owner, Scrum Master, and Agile Coach, need to be embedded in the Agile EA governance, which includes traditional management roles, architecture roles, and supporting EA processes. An organization needs to communicate its architectural vision top–down to all agile teams regularly, outline intentional architectures and standards to allow for long-term consistency between services at the enterprise level, and guide agile teams with architecture roadmaps and implementation guidelines. Management should provide top–down guidance and a clear direction to agile teams. Business departments influence the shaping of architecture roadmaps. Cross-team planning (QBR) sessions for intrateam alignment can be used as checkpoints for the organization's direction. Enterprise architects provide direction across agile teams while overseeing service chains. They create intentional architectures, quickly identify dependencies between agile teams, and indicate when they should be discussed in QBRs. This facilitates priority setting across all agile teams, improves communication between teams, and facilitates managing dependencies. Lead architects should work hands-on with the agile teams, ensure the roadmap remains clear and impediments are removed, and monitor implementations.

Even when reenvisioning EA in an agile context, some waterfall-related characteristics of EA remain important and should be maintained. Therefore, the organization should leverage value-adding waterfall-oriented characteristics from EA approaches and use these to complement a hybrid ASF. On the one hand, this more balanced way of thinking will benefit organizations, as people involved in agile projects tend to be project focused and have less attention for the broader business perspective. On the other hand, enterprise architects need to adapt considerably when an organization transforms to an agile way of working. Active involvement in an organization's governance allows architects to maintain the organization's architectural vision throughout all agile teams. These new insights contribute to the Agile body of knowledge and, more specifically, the theory on agile methods and large-scale agile transformations.

Based on the empirical findings, we proposed several governance mechanisms in an Agile EA context to enhance the development of coherent services and increase the overall flexibility of a service organization. Researchers can replicate our qualitative approach in different regions and sectors and validate our findings in these settings. As service companies must deal with laws and regulations, future research could look into sectors that are less regulated and what that would mean for the level of innovations realized. Other suggestions for future research include measuring performance indicators for process and service innovations, such as speed and cost of development, monitoring the Agile EA combination, and allowing further improvements. Our model (see Fig. 1) can be used as a starting point.

This article is of practical relevance for organizations that apply agile methods for larger projects or larger companies. It shows lessons learned from case organizations on applying ASFs and integrating EA. Our research suggests that combining an agile way of working and EA enhances service development and delivery, process innovation, and ultimately the overall flexibility of an organization, provided the following conditions are met:

1) redefine EA and accompanying roles and processes based on the specific organizational needs before any large-scale agile transformation;
2) preserve some stage-gate elements of EA, such as front-end (intentional) architectures, roadmaps, and implementation guidelines to counterbalance a rigid agile implementation; and
3) facilitate cross-team EA coordination, balance team autonomy with top–down guidance, and ensure continuous attention to maintain the organization's architectural vision throughout all agile teams in close cooperation with Business stakeholders.

Organizations should seriously consider an agile transformation that includes EA in their agile governance. To do so, organizations can empower themselves by enabling their architects to properly manage the transformation to an Agile EA of tomorrow.

ACKNOWLEDGMENT

The authors would like to thank the interviewees who took part in the interviews despite their busy schedules and the guest editors and four anonymous reviewers for their constructive comments.

REFERENCES

[1] D. Helbing, S. Balietti, S. Bishop, and P. Lukowicz, "Understanding, creating, and managing complex techno-socio-economic systems: Challenges and perspectives," *Eur. Phys. J. Special Topics*, vol. 195, pp. 165–186, 2011.

[2] P. Weill and S. L. Woerner, "Thriving in an increasingly digital ecosystem," *MIT Sloan Manage. Rev.*, vol. 56, no. 4, pp. 27–34, 2015.

[3] G. Lee and W. Xia, "Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility," *MIS Quart.*, vol. 34, no. 1, pp. 87–114, 2010.

[4] K. Schwab, "Globalization 4.0: Shaping a global architecture in the age of the fourth industrial revolution. The Davos 2019 manifesto," Geneva, Switzerland, World Economic Forum, White Paper, 2018. [Online]. Available: https://www3.weforum.org/docs/WEF_Globalization_4.0_Call_for_Engagement.pdf

[5] E. Overby, A. Bharadwaj, and V. Sambamurthy, "Enterprise agility and the enabling role of information technology," *Eur. J. Inf. Syst.*, vol. 15, no. 2, pp. 120–131, 2006.

[6] M. West and D. Norton, "Market guide for enterprise agile frameworks," *Gartner*, 2017. Accessed: Mar. 22, 2020. [Online]. Available: https://www.gartner.com/document/3762264

[7] 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.

[8] T. Dingsøyr and N. B. Moe, "Towards principles of large-scale agile development," in *Proc. Int. Conf. Agile Softw. Develop.*, Cham, Switzerland, 2014, pp. 1–8.

[9] 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.

[10] J. W. Ross, P. Weill, and D. Robertson, *Enterprise Architecture as Strategy: Creating a Foundation For Business Execution*. Boston, MA, USA: Harvard Business School Press, 2006.

[11] 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.

[12] M. M. Lankhorst, "Introduction to enterprise architecture," in *Enterprise Architecture at Work*, Berlin, Germany: Springer, pp. 10–19, 2013.

[13] T. Dingsøyr, N. B. Moe, T. E. Fægri, 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.

[14] M. T. Gamble, "Can metamodels link development to design intent,?" in *Proc. IEEE/ACM 1st Int. Workshop Bringing Architectural Des. Thinking Developers' Daily Activities*, 2016, pp. 14–17.

[15] 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.

[16] S. Nerur and V. Balijepally, "Theoretical reflections on agile development methodologies," *Commun. ACM*, vol. 50, no. 3, pp. 79–83, 2007.

[17] C. Maples, "Enterprise agile transformation: The two-year wall," in *Proc. 2009 Agile Conf.*, 2009, pp. 90–95.

[18] A. Putta, Ö. Uludağ, M. Paasivaara, and S. L. Hong, "Benefits and challenges of adopting SAFe-An empirical survey," in *Proc. Int. Conf. Agile Softw. Develop.*, Cham, Switzerland, 2021, pp. 172–187.

[19] S. Bente, U. Bombosch, and S. Langade, *Collaborative Enterprise Architecture: Enriching EA with Lean, Agile, and Enterprise 2.0 Practices*. Amsterdam, The Netherlands: Elsevier, 2012.

[20] 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.

[21] P. Kruchten, "Contextualizing agile software development," *J. Softw., Evol. Process*, vol. 25, no. 4, pp. 351–361, 2013.

[22] Ö. Uludağ, P. Philipp, A. Putta, M. Paasivaara, C. Lassenius, and F. Matthes, "Revealing the state-of-the-art in large-scale agile development: A systematic mapping study," *J. Syst. Softw.*, to be published. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2007/2007.05578.pdf

[23] T. Mikaelian, D. J. Nightingale, D. H. Rhodes, and D. E. Hastings, "Real options in enterprise architecture: A holistic mapping of mechanisms and types for uncertainty management," *IEEE Trans. Eng. Manag.*, vol. 58, no. 3, pp. 457–470, Aug. 2011.

[24] W. W. Royce, "Managing the development of large software systems: Concepts and techniques," in *Proc. 9th int. Conf. Softw. Eng.*, 1987, pp. 328–338.

[25] L. Cocco, K. Mannaro, G. Concas, and M. Marchesi, "Simulating kanban and scrum vs. waterfall with system dynamics," in *Proc. Int. Conf. Agile Softw. Develop.*, 2011, pp. 117–131.

[26] N. M. A. Munassar and A. Govardhan, "A comparison between five models of software engineering," *Int. J. Comput. Sci. Issues*, vol. 7, no. 5, pp. 94–101, 2010.

[27] K. Beck *et al.*, "Manifesto for agile software development," 2001, Accessed: Dec. 1, 2021. [Online]. Available: https://agilemanifesto.org/

[28] C. Larman and V. R. Basili, "Iterative and incremental developments: A brief history," *Computer*, vol. 36, no. 6, pp. 47–56, 2003.

[29] B. Boehm and R. Turner, "Using risk to balance agile and plan-driven methods," *Computer*, vol. 36, no. 6, pp. 57–66, 2003.

[30] A. Cockburn and J. Highsmith, "Agile software development, the people factor," *Computer*, vol. 34, no. 11, pp. 131–133, 2001.

[31] M. V. Tatikonda and S. R. Rosenthal, "Technology novelty, project complexity, and product development project execution success: A deeper look at task uncertainty in product innovation," *IEEE Trans. Eng. Manag.*, vol. 47, no. 1, pp. 74–87, Feb. 2000.

[32] J. Chen, R. R. Reilly, and G. S. Lynn, "The impacts of Speed-to-Market on new product success: The moderating effects of uncertainty," *IEEE Trans. Eng. Manag.*, vol. 52, no. 2, pp. 199–212, May 2005.

[33] Ö. Uludağ, A. Putta, M. Paasivaara, and F. Matthes, "Evolution of the agile scaling frameworks," in *Proc. Int. Conf. Agile Softw. Develop.*, Cham, Switzerland, 2021, pp. 123–139.

[34] M. Coram and S. Bohner, "The impact of agile methods on software project management," in *Proc. 12th IEEE Int. Conf. Workshops Eng. Comput.-Based Syst.*, 2005, pp. 363–370.

[35] B. Hobbs and Y. Petit, "Agile methods on large projects in large organizations," *Project Manage. J.*, vol. 48, no. 3, pp. 3–19, 2017.

[36] G. Van Waardenburg and H. Van Vliet, "When agile meets the enterprise," *Inf. Softw. Technol.*, vol. 55, no. 12, pp. 2154–2171, 2013.

[37] 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.

[38] L. Mahadevan, W. J. Kettinger, and T. O. Meservy, "Running on hybrid: Control changes when introducing an agile methodology in a traditional 'Waterfall' system development environment," *Commun. Assoc. Inf. Syst.*, vol. 36, no. 5, 2015, Art. no. 5.

[39] D. Batra, W. Xia, D. E. Van der Meer, and K. Dutta, "Balancing agile and structured development approaches to successfully manage large distributed software projects: A case study from the cruise line industry," *Commun. Assoc. Inf. Syst.*, vol. 27, no. 1, 2010, Art. no. 21.

[40] 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. Manag.*, vol. 68, no. 1, pp. 330–344, Feb. 2021.

[41] K. Dikert, M. Paasivaara, and C. C. Lassenius, , "Challenges and success factors for large-scale agile transformations: A systematic literature review," *J. Syst. Softw.*, vol. 119, pp. 87–108, 2016.

[42] R. G. Cooper, "Agile–Stage-Gate hybrids: The next stage for product development blending agile and stage-gate methods can provide flexibility, speed, and improved communication in new-product development," *Res.-Technol. Manage.*, vol. 59, no. 1, pp. 21–29, 2016.

[43] S. Hastie and S. Wojewoda, "Standish group 2015 chaos report." Accessed: Aug. 28, 2021. [Online]. Available: https://www.infoq.com/articles/standish-chaos-2015

[44] A. Solinski and K. Petersen, "Prioritizing agile benefits and limitations in relation to practice usage," *Softw. Qual. J.*, vol. 24, no. 2, pp. 447–482, 2016.

[45] A. Tarhan and S. G. Yilmaz, "Systematic analyses and comparison of development performance and product quality of incremental process and agile process," *Inf. Softw. Technol.*, vol. 56, no. 5, pp. 477–494, 2014.

[46] H. C. Estler, M. Nordio, C. A. Furia, B. Meyer, and J. Schneider, "Agile vs. structured distributed software development: A case study," *Empirical Softw. Eng.*, vol. 19, no. 5, pp. 1197–1224, 2014.

[47] M. Bianchi, G. Marzi, and M. M. Guerini, "Agile, stage-gate and their combination: Exploring how they relate to performance in software development," *J. Bus. Res.*, vol. 110, pp. 538–553, 2020.

[48] Z. Zhang and H. Sharifi, "Towards theory building in agile manufacturing strategy—A taxonomical approach," *IEEE Trans. Eng. Manag.*, vol. 54, no. 2, pp. 351–370, May 2007.

[49] J. Werewka and A. Spiechowicz, "Enterprise architecture approach to SCRUM processes, sprint retrospective example," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, 2017, pp. 1221–1228.

[50] VersionOne, "15th annual state of agile report," 2021. Accessed: Aug. 28, 2021. [Online]. Available: https://stateofagile.versionone.com/

[51] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, no. 9-10, pp. 833–859, 2008.

[52] 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.*, San Francisco, CA, USA, 2018, pp. 1–17.

[53] T. Gustavsson, "Dynamics of inter-team coordination routines in large-scale agile software development," in *Proc. 27th Eur. Conf. Inf. Syst.*, Uppsala, Sweden, 2019, pp. 1–16.

[54] S. N. Tengstrand, P. Tomaszewski, M. Borg, and R. Jabangwe, "Challenges of adopting SAFe in the banking Industry–A study two years after its introduction," in *Proc. Int. Conf. Agile Softw. Develop.*, Cham, Switzerland, 2021, pp. 157–171.

[55] DIN/DKE, "German standardization roadmap version 2," Berlin, Germany: DKE, 2016.

[56] J. A. Zachman, "A framework for information systems architecture," *IBM Syst. J.*, vol. 26, no. 3, pp. 276–292, 1987.

[57] Q. Bui, "Evaluating enterprise architecture frameworks using essential elements," *Commun. Assoc. Inf. Syst.*, vol. 41, pp. 121–149, 2017.

[58] S. Murer and B. Bonati, *Managed Evolution: A Strategy For Very Large Information Systems*. Berlin, Germany: Springer Science Business Media, 2010.

[59] T. Tamm, P. B. Seddon, G. Shanks, and P. Reynolds, "How does enterprise architecture add value to organisations?," *Commun. Assoc. Inf. Syst.*, vol. 28, pp. 141–168, 2011.
[60] 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. Model. Softw. Des.*, 2014, pp. 38–46.
[61] S. Kotusev, "TOGAF-based enterprise architecture practice: An exploratory case study," *Commun. Assoc. Inf. Syst.*, vol. 43, pp. 321–359, 2018.
[62] M. M. Lankhorst, W. P. M. Janssen, H. A. Proper, and M. W. A. Steen, "Introducing agile service development," in *Agile Service Development*, Berlin, Germany: Springer, 2012, pp. 1–15.
[63] R. M. Foorthuis, M. van Steenbergen, S. Brinkkemper, and W. Bruls, "A theory building study of enterprise architecture practices and benefits," *Information Systems Frontiers*, vol. 18, no. 3, pp. 541–564, 2016.
[64] J. W. Ross, C. M. Beath, and M. Mocker, *Designed For Digital: How to Architect Your Business For Sustained Success*. Cambridge, MA, USA: MIT Press, 2019.
[65] 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.
[66] S. Hanschke, J. Ernsting, and H. Kuchen, "Integrating agile software development and enterprise architecture management," in *Proc. 48th Hawaii Int. Conf. Syst. Sci.*, 2015, pp. 4099–4108.
[67] Ö. Uludağ, M. Kleehaus, N. Reiter, and F. Matthes, "What to expect from enterprise architects in large-scale agile development? A multiple-case study," in *Proc. 25th Amer. Conf. Inf. Syst.*, Cancun, Mexico, 2019, pp. 1–10.
[68] 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.
[69] 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. Manage. Assoc. Conf.*, 2019, pp. 3109–3119.
[70] B. Horlach, A. Drechsler, I. Schirmer, and P. Drews, "Everyone's going to be an architect: Design principles for architectural thinking in agile organizations," in *Proc. 53rd Hawaii Int. Conf. Syst. Sci.*, 2020, pp. 6197–6206.
[71] S. Buckl, F. Matthes, I. Monahov, S. Roth, C. Schulz, and C. M. Schweda, "Towards an agile design of the enterprise architecture management function," in *Proc. IEEE 15th Int. Enterprise Distrib. Object Comput. Conf. Workshops*, 2011, pp. 322–329.
[72] Ö. Uludağ, S. Nägele, and M. Hauder, "Establishing architecture guidelines in large-scale agile development through institutional pressures: A single-case study," in *Proc. 25th Amer. Conf. Inf. Syst.*, 2019, pp. 1–10.
[73] 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.*, Cham, Switzerland, 2021, pp. 140–156.
[74] T. Dingsøyr, N. B. Moe, and E. A. Seim, "Coordinating knowledge work in multiteam programs: Findings from a large-scale agile development program," *Project Manage. J.*, vol. 49, no. 6, pp. 64–77, 2018.
[75] B. Horlach, T. Böhmann, I. Schirmer, and P. Drews, "IT governance in scaling agile frameworks," in *Proc. Multi-conf, Bus. Inform.*, Lüneburg, Germany, 2018, pp. 1789–1800.
[76] S. B. Bacharach, "Organizational theories: Some criteria for evaluation," *Acad. Manage. Rev.*, vol. 14, no. 4, pp. 496–515, 1989.
[77] P. G. Keen, *Shaping the Future: Business Design Through Information Technology*, Boston, MA, USA: Harvard Business School Press, 1991.
[78] P. Weill and J. W. Ross, *IT Governance: How Top Performers Manage IT Decision Rights For Superior Results*, Boston, MA, USA: Harvard Business Press, 2004.
[79] R. K. Yin, *Case Study Research Design and Methods*, 5th ed. Thousand Oaks, CA, USA: Sage, 2014.
[80] D. S. Cruzes and T. Dyba, "Recommended steps for thematic synthesis in software engineering," in *Proc. Int. Symp. Empirical Softw. Eng. Meas.*, 2011, pp. 275–284.
[81] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Softw. Eng.*, vol. 14, no. 2, pp. 131–164, 2009.
[82] K. O'Reilly, D. Paper, and S. Marx, "Demystifying grounded theory for business research," *Org. Res. Methods*, vol. 15, no. 2, pp. 247–262, 2012.
[83] M. Toivonen and T. Tuominen, "Emergence of innovations in services," *Serv. Ind. J.*, vol. 29, no. 7, pp. 887–902, 2009.
[84] M. Dziallas and K. Blind, "Innovation indicators throughout the innovation process: An extensive literature analysis," *Technovation*, vol. 80, pp. 3–29, 2019.

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 a Lecturer and Thesis Coach with the Rotterdam School of Management, Erasmus University, Rotterdam, The Netherlands. He is a Business Advisor with the Dutch Government Agency ICTU and is the founder of Apexis, a firm that provides consultancy services in his research fields, which include the interaction of business and information technology, in particular agile methods, enterprise architecture, information, and data management, information security and privacy, and standardization.

Dr. van Wessel was the recipient of the Aart Bosman Award 2008–2009 as best Information Management thesis 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 (*cum laude*) in business information management from the Rotterdam School of Management, Erasmus University, Rotterdam, The Netherlands, in 2018.

In his professional work, he advises multinationals on ITM&A related challenges (e.g., system integration, data migration, and decommissioning) and guides clients to realize their desired agile business transformations as a SAFe Program Consultant. His academic research interests include the adoption of agile-scaling frameworks in organizations and their impact on the traditional enterprise architecture capability.

Henk J. de Vries graduated as a Geodetic Engineer from the Delft University of Technology, Delft, The Netherlands, in 1982, and received the Ph.D. degree in business administration from the Erasmus University, Rotterdam, The Netherlands, in 1999.

From 1984 to 2003, he was with The Netherlands Standardization Institute NEN in different roles. Since 1994, he has been an Assistant Professor and Associate Professor with Innovation Management Department, Rotterdam School of Management (RSM), Erasmus University. In 2018, he was appointed as a Professor of standardization management, Department of Technology and Operations Management, RSM, Erasmus University, Section Innovation Management. He is also 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.

Prof. de Vries was the President of the European Academy for Standardization from 2010 to 2021. He was the recipient of the International Organization for Standardization ISO Award for his education on standardization as best in the world in 2009. The majority of his Ph.D. students received a National or International Best Dissertation Award.