Investigating the Establishment of Architecture Principles for Supporting Large-Scale Agile Transformations

Ömer Uludağ*, Henderik A. Proper†,‡ Florian Matthes*
*Technische Universität München, Germany
Email: {oemer.uludag,matthes}@tum.de
†Luxembourg Institute of Science and Technology, Luxembourg
‡University of Luxembourg, Luxembourg
Email: e.proper@acm.org

***Abstract*—The widespread use of agile methods shows a fundamental shift in the way organizations try to cope with unpredictable competitive environments. In large-scale agile settings, multiple development activities need to be coordinated to achieve desirable enterprise-wide effects and agility. A powerful instrument to effectively guide and steer large-scale agile endeavors is the formulation and usage of architecture principles. Despite their raison d'être to guide large organizational transformations, extant studies on how principles can be used to support large-scale agile transformations are still lacking.**

Against this backdrop, we present a multiple-case study involving five German companies that aims to shed light on the establishment of architecture principles to support large-scale agile transformations. Based on our results from sixteen semi-structured interviews, we present current practices as well as challenges faced by organizations during the application of architecture principles. In addition, we show a set of principles used to support large-scale agile transformations.

***Index Terms*—Architecture principles, large-scale agile transformation, multiple-case study**

I. Introduction

Today's business environments are characterized by uncertainty and turbulence, stemming from factors such as regulatory uncertainties, continuously changing customer demands, and rapidly evolving technological advancements [1]–[3]. As a consequence, enterprises pursue large-scale agile transformations to deal with dynamic environments and to sustain their survival [4], [5].

These transformations entail new managerial challenges [6], [7] such as doubts and skepticism towards the new way of working [8], [9] or middle managers blocking transformation processes [8], [9]. Further challenges include coordination and alignment issues between large-scale agile activities as well as between agile and non-agile business units [8], [10]. An additional issue is the unclear interplay between top-down architecture governance efforts and bottom-up autonomy of agile teams [11], [12].

The latter two challenges can be addressed by applying architecture principles. Architecture principles provide a simple-rule-like instrument to effectively guide large-scale agile transformations by restricting design freedom in a purposeful manner while avoiding "*Analysis Paralysis*" or "*Big Design Upfront*" [13]–[15].

Notwithstanding the importance of architecture principles for large-scale agile transformations, extant literature disregards (1) the way architecture principles are formulated and applied in this context, (2) the challenges faced during their application, (3) and the principles used to support large-scale agile endeavors. Against this backdrop, we formulate the following research questions:

RQ 1: How are architecture principles established to support large-scale agile transformations?

RQ 2: Which challenges do organizations face when formulating and using architecture principles?

RQ 3: Which architecture principles are used to support large-scale agile transformations?

The remainder of this paper is structured as follows. In Section II, we portray the research design of our paper. Section III introduces a set of definitions and reports on the current state of research. Section IV presents the results of the multiple-case studies. In Section V, we discuss our main findings and limitations before concluding our study with a summary of our results and remarks on future research in Section VI.

II. Research Methodology

Given that our research is motivated by a practical problem, we applied case study research as it provides an in-depth overview of real-life situations and contemporary phenomena [16]. Hereafter, we outline the design of this case study in line with the guidelines of Runeson and Höst [17].

Case Study Design: Our main goal is to explore current practices and challenges in the establishment of architecture principles in large-scale agile transformations. Based on this objective, we formulated three research questions (see Section I). Our study employs a multiple-case study design with five organizations that allows cross-case analysis [18]. The cases were purposefully selected because they undergo major large-scale agile transformations and have a long history in using architecture principles in their EAM (enterprise architecture management) endeavors. This allows us to observe how they adapted the application of principles and which difficulties they experienced during this adaptation. We selected cases from various industries to avoid industry bias. Table I provides an overview of the case organizations and interviews.

TABLE I
CASE ORGANIZATIONS AND CONDUCTED INTERVIEWS

Code name	Employees	Interviews	Position of interviewees
Global insurance company ("GlobalInsureCo")	140,000+	2	Enterprise architect
Car manufacturer ("CarCo")	130,000+	4	Chief technology officer; enterprise architect; group lead IT; requirements engineer; scrum master
IT company ("ITCo")	7,000+	2	Enterprise architect; product owner
Retail company ("RetailCo")	50,000+	4	Chapter lead business process architecture; enterprise architect; product owner; scrum master
Public sector insurance company ("PublicInsureCo")	6,700+	4	Agile developer; enterprise architect; head of IT governance department

Data Collection: We focused primarily on first- and third-degree data collection techniques [19]. We used first-degree methods to get in direct contact with the subjects and to collect data in real time. To this end, we conducted sixteen individual and group interviews. In almost all companies, at least one senior executive, one enterprise architect (EA), and one member of an agile team (AT) were interviewed to gain a diverse perspective on the subject under investigation and to triangulate our findings. The interviews followed a semi-structured questionnaire and were rather conversational to allow interviewees to explore their views in detail [18]. Each interview lasted 30-70 minutes and was primarily conducted in face-to-face meetings. At least two researchers were present to facilitate observer triangulation [17]. We supplemented our interview findings with third-degree data collection techniques that allowed us to analyze existing work artifacts and available data. Here, slide decks and wiki pages of the cases provided us with detailed information about the documentation and application of architectural principles. The purposeful selection of several data sources and roles from different case organizations facilitated the triangulation of data sources [20].

Data Analysis: We recorded and transcribed the interviews and then coded them with the slide decks and wiki pages using open coding [21]. The data analysis procedure was supported by the data analysis software MAXQDA[1]. We then consolidated the preliminary codes and checked them for consistency and completeness. Subsequently, we combined groups of code phrases into concepts that were later related to the formulated research questions.

[1]https://www.maxqda.com/

III. Background and Related Work

A. Large-Scale Agile Transformation

Literature on the large-scale adoption of agile methods often involving a discussion about the meaning and conceptualization of "*agile in the large*" [22], [23]. The term "*large-scale agile development*" has multiple interpretations: (1) the use of agile methods in large teams, (2) the employment of agile methods in large organizations, (3) the application of agile methods in large multi-team settings or (4) the usage of agile practices in organizations as a whole [24]. In this paper, we focus on the latter two interpretations. In line with the authors of [6] and [8], we understand the large-scale adoption of agile methods on an organizational level with multi-team settings consisting of 50 or more people or at least six teams. The second part in the term "*large-scale agile transformation*" refers to the transition from traditional development approaches to agile methods. This may involve a one-time big bang transfer to agile methods in a large setting or a step-by-step approach in which an agile pilot is then scaled up into a large setting [6].

B. Agile architecture

In [25], the authors discuss the connection between agility and architecture, in particular involving suggestions towards *architecture processes in an agile context*.

The latter needs to strike a balance between the "bottom up" orientation of an agile approach with the "top down" tendency of traditional architectural approaches. This results in the suggestion to indeed take the overall architecture process as put forward by [26], involving *creating*, *applying*, and *maintaining* an architecture, on board, but to treat these three core activities as parallel *streams* rather than a cyclic *sequence*.

In [25], it is also suggested that is a good (agile) practice to ensure that architecture largely emerges from projects, and is not just invented from the top down. The focus should be on those aspects that really need to be decided at an enterprise-wide level. In line with this, it is also suggested to focus architectural work towards a *risk-driven approach* and the management of (cross project) *refactoring and technical debt*.

C. Flavours of principles

Architecture principles provide a powerful, simple-rule-like instrument to effectively guide and steer large-scale transformations by restricting design freedom in a purposeful manner [14]. As such, and in line with the discussion in Section III-B, they provide a means to materialize the guiding and steering role of EAM by focusing on the essential considerations [13]–[15], [25], while avoiding "*Analysis Paralysis*" or "*Big Design Upfront*".

It is important to clarify the two important flavours of principles that play a role in EAM. As argued in [27], in an architectural context, there are two important interpretations of the notion of principle. The first interpretation considers a principle as a *law or fact of nature underlying the working of an artifact* [27]. As an example, consider the Archimedes principle from the field of nautic engineering, which states that "*any object, wholly or partially immersed in a fluid, is buoyed up by a force equivalent to the weight of the fluid displaced by the object*". Closer to our world, An example from general systems engineering is the law of *requisite variety* [28], which conveys the fact that the number of states of a control mechanism must be greater than or equal to the number of states if the part of the system that needs to be controlled.

The second interpretation considers a principle as a *declarative statement that normatively prescribes a property of something* [27]. General examples include the *ten Commandments* from the Bible, e.g., *Thou shalt not murder* and *Thou shalt not commit adultery*. In an EAM context, an examples would be: *Clients can access the entire portfolio of services offered by any part of the government by way of all channels through which government services are offered*. Such principles may also express fundamental beliefs on how things 'ought to be', such as: *No wrong doors* (suggesting that clients should be helped by which ever channel they approach the enterprise) and *The customer is always right*.

D. Architecture principles

The two flavours of principles as discussed above, are fundamentally different in nature. A *law or fact of nature* (is expected to) be true by its very definition. It may be required to gather (further) evidence to prove / falsify its correctness, however, no additional "compliance" checking mechanism is needed. Nevertheless, these principles are also relevant in an architectural context, as they may provide the *motivation* for decisions [29], [30].

Principles as being *declarative statement that normatively prescribes a property of something* will not hold by definition. They require some additional mechanism, and associated effort, to enforce, or at least 'invite', compliance to the principle. Principles this sense, potentially provide the means to articulate high level design decisions, providing directions / guidance towards more specific design [26], [27], [31], [32].

Both flavours of principles are relevant in an EAM context. Nevertheless, most scholars and practitioners, including the IEEE standard on architectural descriptions [33] and TOGAF [34], focus on the interpretation as a *declarative statement that normatively prescribes a property of something*, which is then referred to as *architecture principles* [27].

However, the author of [30] suggests that, in conformance to the use of the word *principle* in other disciplines, e.g., civil engineering, it would have actually been better to use the word *architecture principle* in the sense of a *law or fact of nature*.

In this paper, we follow the interpretation as currently used within the field of EAM, as being a *declarative statement* [27].

E. Generic Architecture Principles Process

In [27], the authors suggest a generic process to handle the life-cycle of architecture principles (see Fig.1). As discussed in [27], this generic process has been distilled from existing methods, research, and case studies on architecture principles. The process itself consists of eight sub-processes:

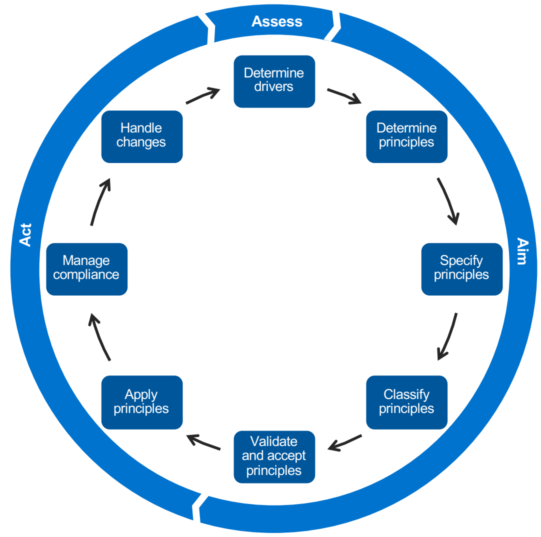

Fig. 1. Generic process for handling architecture principles [14]

Determine drivers where the relevant inputs for determining architecture principles are collected, such as the goals and objectives, issues and risks.

Determine principles where the drivers are translated to a list of (candidate) architecture principles.

Specify principles where the candidate principles are specified in detail, including their rationale and implications.

Classify principles where architecture principles are classified in a number of dimensions to increase their accessibility.

Validate and accept principles where architecture principles, their specifications and classifications are validated with relevant stakeholders and formally accepted.

Apply principles where architecture principles are applied to construct models and derive design decisions in downstream architectures, requirements and designs.

Manage compliance where architects ensure that the architecture principles are applied properly, and dispensations for deviations may be given.

Handle changes where the impact of all sorts of changes on the architecture principles is determined and new method iterations may be initiated.

Being a generic process, it would have to be translated to organization-specific process catering for the situation at hand [14]. This of course, also involves the integration with the architecture and agile practices used in a specific organization. For example, the generic process as such does not specifically require a top-down or bottom-up approach to the actual formulation of the principles.

F. Collaborative formulation of architecture principles

In general, and in an agile context in particular, the formulation of architecture principles requires a close collaboration between different stakeholders. To this end, the generic process as depicted in Fig. 1 can be refined with a situation specific collaborative strategy. The field of collaboration engineering [35] has proposed the notion of ThinkLets ("*a known pattern*

TABLE II
OVERVIEW OF THE CASE ORGANIZATIONS

	GlobalInsureCo	**CarCo**	**ITCo**	**RetailCo**	**PublicInsureCo**
Transformation begin	Beginning of 2016	Beginning of 2017	Beginning of 2015	End of 2017	Mid of 2016
Reasons for the transformation	Effectiveness, quality, and efficiency; Time-to-market; Modern working environments; Customer-centricity	Development speed; End-to-end responsibilities	Finding young and motivated employees; Short response times	Time-to-market; Flexibility to changing conditions; Customer experience; Collaboration between business & IT	Business model changes; Time-to-market; Customer satisfaction; Flexibility in software development
Used scaling frameworks	Large-Scale Scrum; Scaled Scrum; Spotify Model	Large-Scale Scrum; Scaled Agile Framework; Scaled Scrum	Large-Scale Scrum; Scaled Agile Framework; Scaled Scrum	Scaled Agile Framework; Scaled Scrum; Spotify Model	Enterprise Scrum; Nexus; Scaled Agile Framework; Scaled Scrum
Transformation approach	Factory approach with dedicated co-locations	Driven by the CIO and executives; Line of business piloting	Steered by a central transformation team	Piloting large projects; Line of business piloting; Driven by management board	Piloting large projects; Line of business piloting
Transformation scale	Enterprise-wide transformation	Enterprise-wide transformation	Enterprise-wide transformation	Enterprise-wide transformation	Enterprise-wide transformation

of collaboration among people working together toward a goal", [35]) to construct collaborative strategies. Different collaborative strategies for the formulation of architecture principles have been explored in e.g., [13], [36], [37].

IV. Establishing Architecture Principles to Support Large-Scale Agile Transformations

In this section, we consider the results of a multiple-case study in the use of architecture principles to support large-scale agile transformations.

A. Case Descriptions

Table II provides an overview of the investigated organizations and their ongoing large-scale agile transformations.

B. Current Practices

We used exploratory questions to identify current practices in establishing architecture principles. Thereby, we asked the interviewees about following categories: (1) drivers and goals, (2) responsibilities, (3) documentation, (4) communication, (5) and compliance. Our results are presented below.

First, we asked interviewees to indicate their external drivers for defining architecture principles and the goals they pursue with the application of principles. An overview of drivers and goals can be found in Table III and Table IV[2].

We observed several similarities between the case companies regarding responsibilities for defining and creating architecture principles, e.g., across all case organizations, EAs were responsible for the process of establishing architecture principles. Second, everyone was able to propose architecture principle candidates. Based on these, EAs specified architecture principles. At GlobalInsureCo and CarCo, ATs could specify principles who were supported by EAs. Here, EAs refined principles and provided feedback. Final approval of

[2]For space reasons, we only listed recurring drivers and goals that have been mentioned in at least two organizations.

TABLE III
MAIN EXTERNAL DRIVERS TO ESTABLISH ARCHITECTURE PRINCIPLES

External Drivers	**Global-Insure-Co**	**CarCo**	**ITCo**	**Retail-Co**	**Public-Insure-Co**
Regulatory Requirements	✓	✓	✓	✓	✓
Technological Advancements	✓	✓	✓	✓	✗
Changing Business Models	✗	✓	✓	✓	✓
Changing Market Conditions	✗	✗	✗	✓	✓
Product Innovations	✗	✓	✗	✗	✓

architecture principles was subject to a central architecture committee or an architecture community consisting of EAs and AT members. The participants also stated that architecture boards have been increasingly replaced by architecture communities.

All case companies used specification templates for documenting architecture principles. Table V provides an overview of the respective attributes. All case companies used at least the sections *Name* and *Description* for documenting their principles. The comparison also shows that they all used similar attributes to TOGAF, i.e., *Name*, *Rationale*, and *Implications* [38]. GlobalInsureCo, RetailCo, and PublicInsureCo extended the specification template by a *Binding Nature* attribute to indicate recommended or mandatory principles. In addition, RetailCo used the *Validity Range* section to express the organizational units or levels where a principle should be applied. RetailCo and PublicInsureCo added an *Examples* section to illustrate how a principle can be applied. Last but not least, a *See Also* section provides references to other principles or standards. The case companies used their team collaboration tools to make their architecture principles available.

In all case organizations, architecture principles became valid

TABLE IV
MAIN GOALS FOR APPLYING ARCHITECTURE PRINCIPLES

Main Goals	**Global-Insure-Co**	**CarCo**	**ITCo**	**Retail-Co**	**Public-Insure-Co**
Software Quality / Code Quality	✓	✓	✓	✓	✓
Efficiency / Effectiveness	✓	✓	✓	✓	✓
Reusability / Standardization	✓	✓	✗	✓	✓
Common Direction / Alignment	✗	✓	✓	✓	✓
Balancing Up-front & Emergent Architecture	✗	✓	✓	✓	✓
Compliance / Security	✓	✓	✗	✗	✓
(IT) Strategy	✗	✓	✓	✓	✗
Operational Reliability	✓	✗	✗	✓	✗
Flexibility / Adaptiveness / Responsiveness	✗	✗	✗	✓	✓
Empowered Teams	✗	✗	✗	✓	✓
Time-to-Market	✗	✗	✗	✓	✓

TABLE V
DOCUMENTATION TEMPLATES OF ARCHITECTURE PRINCIPLES

Attributes	**Global-Insure-Co**	**CarCo**	**ITCo**	**Retail-Co**	**Public-Insure-Co**
Name	✓	✓	✓	✓	✓
Definition	✗	✗	✗	✓	✓
Description	✓	✓	✓	✓	✓
Rationale	✓	✓	✗	✗	✓
Implications	✗	✓	✗	✓	✓
Binding Nature	✓	✗	✗	✓	✓
Validity Range	✗	✗	✗	✓	✗
Examples	✗	✗	✗	✓	✓
See Also	✓	✗	✗	✓	✓

for all future development endeavors after final approval by an architecture board or community. Apart from a few exceptions, e.g., due to security issues, existing IT systems were not affected by new principles. This rule is also called *"grandfathering"*. The cases used both pull and push strategies to communicate new architecture principles. Pull strategies included publishing architecture principles in central EAM repositories and wikis where ATs had to actively gather information. In contrast, push strategies comprised the communication of principles via status calls, emails or architecture communities that relay the information to the ATs.

Last but not least, we asked the interviewees how they ensure the compliance of architecture principles. Our main observation was the lack of appropriate tools and (automated) control mechanisms to verify and control compliance. The compliance check was mainly performed manually, either through a random review of an architecture board (GlobalInsuraceCo, CarCo) or through a sporadic review of a team architect (RetailCo, ITCo). Some participants also stated that the verification was based on *"trust"* (CarCo). Although code reviews at the end of product delivery would provide a way to identify compliance deviations, participants explained that continuous compliance verification within the Sprints would be more preferable to steer ATs in the right direction and save future and expensive refactoring processes. An interesting approach we observed was that in some cases the architecture principles were used as acceptance criteria in the Definition of Ready (DoR)[3] (CarCo, RetailCo).

C. Challenges

We were also curious about the challenges the case organizations faced in formulating and applying architecture principles. Table VI provides an overview of identified challenges. In total, we identified twelve challenges of which nine were observed in at least two case companies. Three interesting observations can be made: First, in all cases, we observed the challenge of ATs not understanding the benefits of adhering to architecture principles. As a consequence, they resisted the application of principles (see Challenge #1) and prioritized business value over architectural improvements (see Challenge #4). Second, we identified the assurance of compliance with architecture principles as another major challenge for the case companies due to lack of appropriate tools (see Challenge #2), lack of assessment criteria (see Challenge #6), and unclear implications of non-compliance (see Challenge #10). Third, another major challenge constitutes the actual implementation of the principles, given the lack of support from EAs (see Challenge #8) and the lack of feedback cycles to express difficulties in applying the principles (see Challenge #9).

D. Principles Supporting Large-Scale Agile Transformations

In order to identify a set of suitable architecture principles used in practice to support large-scale agile transformations, we first reviewed academic and practitioner literature and collected proposed architecture principles. Our primary sources were peer-reviewed EAM-related publications (cf. [39]) as well as practitioner publications offering comprehensive collections of principles (cf. [14], [38], [40], [41]). In total, we identified 127 proposed architecture principles, which were analyzed based on their statements and rationales. During the analysis, we consolidated similar architecture principles and removed duplicates. We excluded architecture principles having too abstract statements and those contradicting agile values or principles [42]. Based on this analysis, 30 of 127 architecture principles were considered suitable and made available to respondents. During the interviews, we asked interviewees to rate and justify the extent to which a particular principle is considered relevant for supporting their large-scale agile transformations, as well as the extent to which it is applied within the organization. In addition, the respondents were able to bring their architecture principles used in their organization to the interviews to compare them with our collection and supplement it if necessary. In many cases, we were able to match their principles with one of ours. Only in a few cases,

[3]The DoR is utilized to describe user stories ready for implementation.

TABLE VI
RESULTS OF THE CROSS-CASE ANALYSIS FOR THE IDENTIFICATION OF CHALLENGES IN THE APPLICATION OF ARCHITECTURAL PRINCIPLES

No.	Identified Challenges	GlobalInsureCo	CarCo	ITCo	RetailCo	PublicInsureCo	Occcurences
#1	Lack of understanding and acceptance why ATs should adhere to architecture principles	✓	✓	✓	✓	✓	5
#2	Difficulties in controlling compliance of ATs with architecture principles	✓	✓	✓	✗	✓	4
#3	Difficulties in communicating new architecture principles to ATs	✓	✓	✗	✓	✗	3
#4	Conflicting priorities in terms of delivering business value vs. long-term architecture improvements	✓	✓	✗	✗	✓	3
#5	Difficulties in regularly adapting architecture principles to new environments	✓	✗	✗	✗	✓	2
#6	Lack of criteria to indicate whether ATs fulfill compliance	✓	✓	✗	✗	✗	2
#7	Lack of transparency where architecture principles to be considered can be found	✓	✓	✗	✗	✗	2
#8	Lack of concrete implementation guidance	✓	✓	✗	✗	✗	2
#9	Lack of feedback cycles to determine whether architecture principles can be applied by ATs	✓	✓	✗	✗	✗	2
#10	Lack of escalation mechanisms in case of non-compliance	✓	✗	✗	✗	✗	1
#11	Lack of understanding when architecture principles are valid	✗	✗	✓	✗	✗	1
#12	Improper understanding of agile values complicates the application of architecture principles	✗	✗	✗	✗	✓	1

respondents mentioned organization-specific principles that were excluded from our results for reasons of comparability.

Fig. 2 provides an overview of how the case organizations have assessed the relevance of the architecture principles. The architecture principle *"Applications do not cross business function boundaries"* was rated as the most relevant principle across all organizations. Interestingly, *"IT systems are standardized and reused throughout the organiziation"* was rated as the second most relevant principle, although, this principle is perceived as very restrictive by ATs. This principle is important for EAs to facilitate the reuse of IT systems and components across the organization [43].

A list of architecture principles implemented by the case organizations can be found in Fig. 3. We also identified a positive correlation between the relevance assessment of architecture principles and their actual application.

We also asked the interviewees to indicate potential conflicts between different stakeholders. We identified three typical conflicts. First, the interviewees saw the biggest conflict in architecture principle intending to standardize technology stacks of ATs or to specify their technology selection in the upfront. This conflict typically also leads to the field of tension between top-down governance and the autonomy of ATs [12]. This conflict was identified in 8 of the 30 proposed principles such as *"IT Systems Are Standardized and Reused Throughout the Organization"* or *"Applications rely on one technology stack"*. The second conflict was the shift from old architecture paradigms such as legacy systems to new architecture patterns such as microservice architectures. This conflict was identified in five architecture principles, e.g., *"Applications do not cross business function boundaries"* or *"Loose coupling of systems or services"*. Third, the two principles *"Buy application when it does not provide a significant competitive advantage in your core business"* and *"Vertical system slicing"* were regarded as promoting the heterogeneity of the IT landscape and conflicting with the idea of standardization.

Last but not least, we asked the participants to name architecture principles that would be unsuitable for supporting large-scale agile transformations. The respondents were not able provide concrete principles but described characteristics of principles that would hamper large-scale agile transformations. The majority of the interviewees regarded architecture principles being too restrictive or specific as inappropriate. Also, principles that would promote highly-centralized decisions or silos were considered unsuitable. Further, the respondents regarded principles that focus too much on cost savings and efficiency as not helpful.

V. Discussion

In the following, we discuss the main outcomes and limitations of our study.

A. Key Findings

Three key findings emerge from this multiple-case study.

First, our results show that architecture principles steer large-scale agile transformations, not only by connecting strategic considerations to the execution of transformation projects and restricting design freedom [14], [44], but also by aligning multiple large-scale agile endeavors and by creating a common understanding to achieve desirable organization-wide effects.

Second, the establishment of architecture principles also present some challenges for organizations, some of which have already been reported in [44], [45], such as difficulties in enforcing principles, measuring principle implementations, and involving relevant stakeholders. However, we have observed further challenges faced by organizations, such as difficulties in communicating new principles, providing concrete

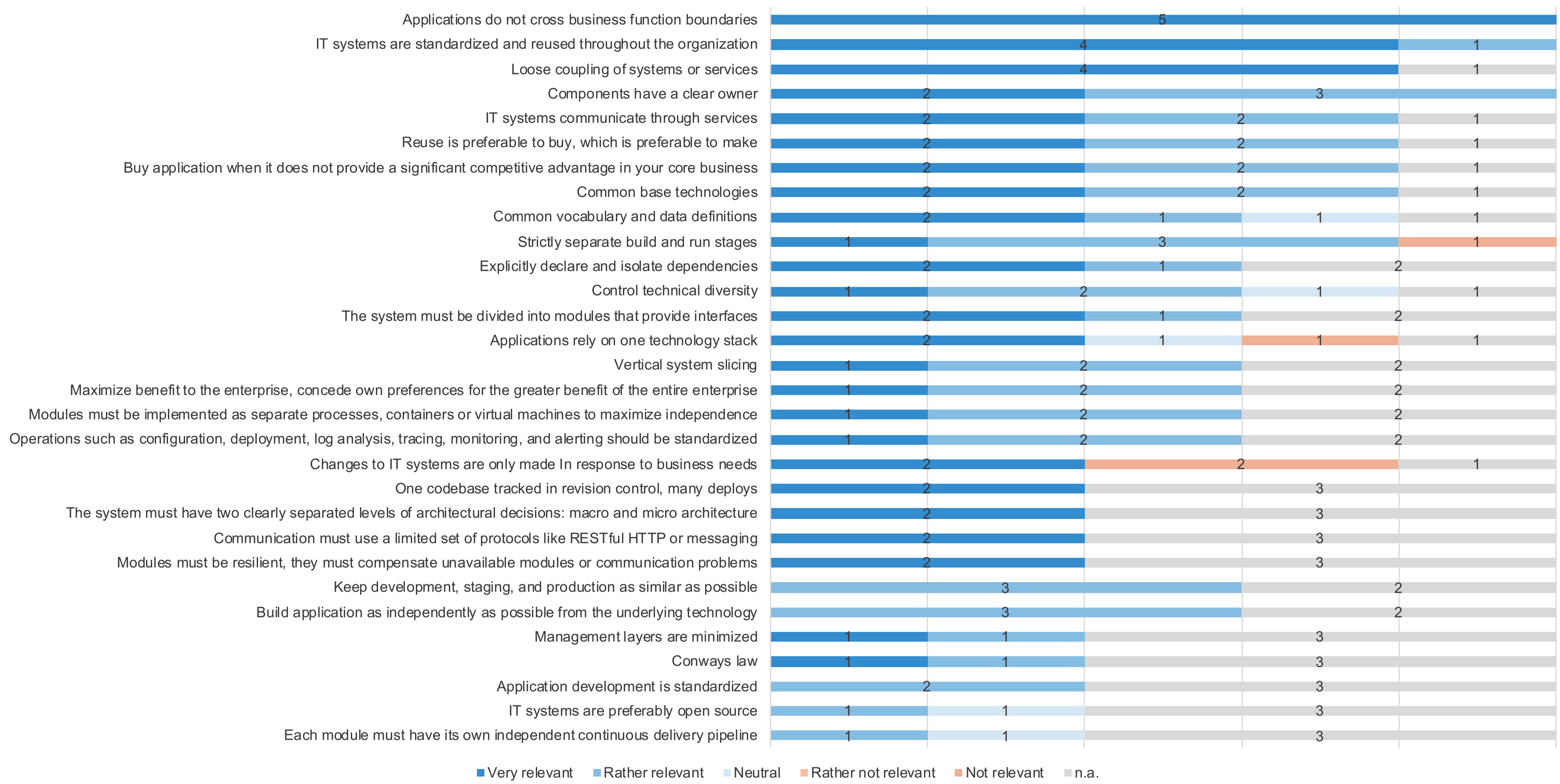

Fig. 2. Relevance of provided architecture principles for supporting the large-scale agile transformations of the case organizations (n=5)

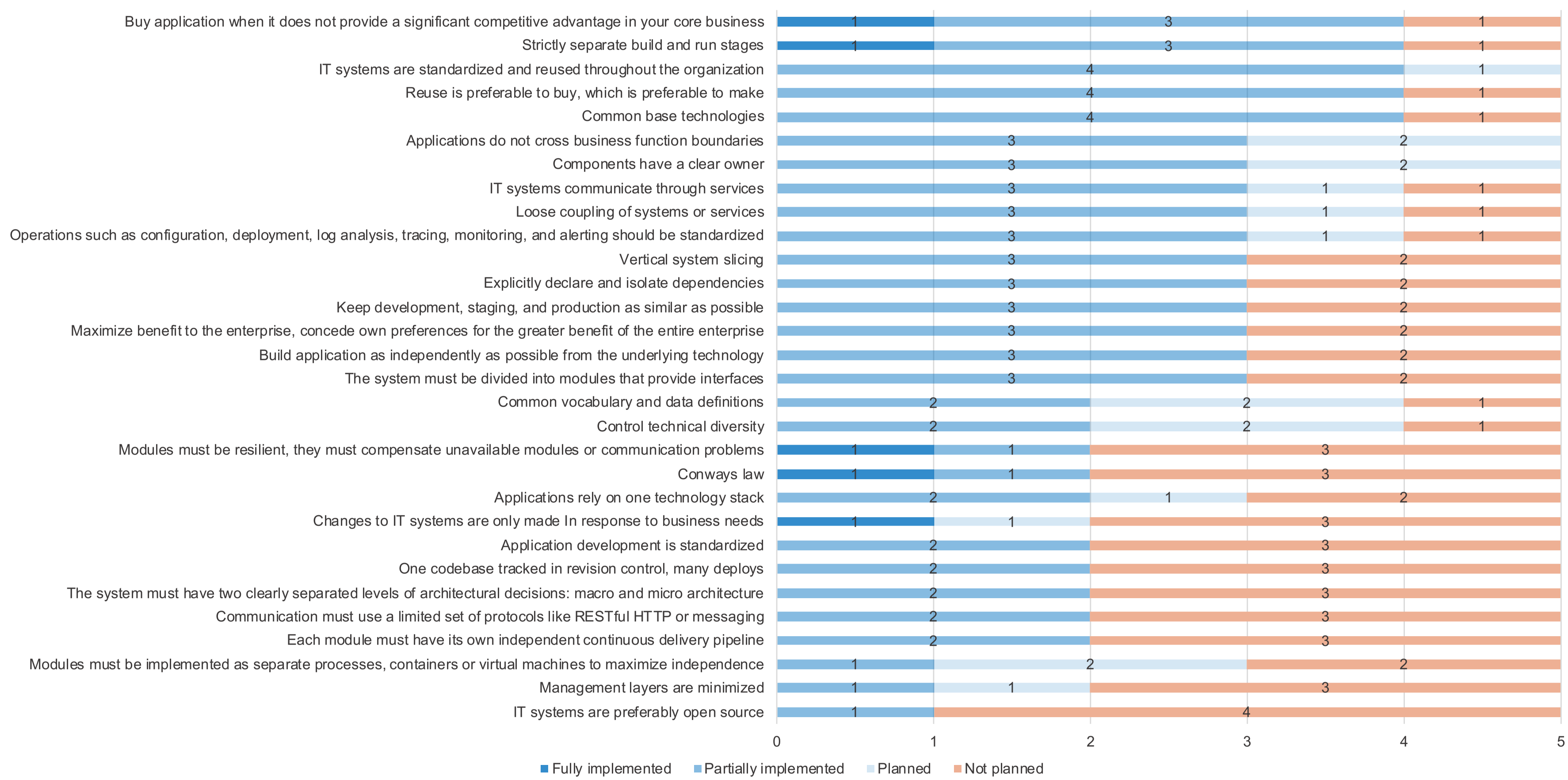

Fig. 3. Application of architecture principles across the case organizations (n=5)

implementation guidance, and demonstrating the benefits of principles.

Third, a common challenge of large-scale agile transformations is to find the right balance between emergent and intentional architecture [43]. In several interviews, ATs and EAs emphasized the importance of using architecture principles to achieve this balance, calling it *principle-based intentional architecting*. On the one hand, principles give an orientation aid for development endeavors to achieve a desired future architecture and facilitate to create (intentional architecture). On the other hand, principles provide a solution space in which ATs can move, without dictating how the specified principles should be realized (emergent architecture). In contrast to detailed architecture specifications, they also prevent "*Big Design Upfront*" by emphasizing the agile architecture principle "*build the simplest architecture that can possibly work*" [46].

B. Limitations

Our paper is not without limitations, wherefore we discuss possible validity threats to our observations along with an evaluation scheme recommended by Runeson and Höst [17].

Construct validity refers to what extent the operational measures studied represent what the researcher has in mind. As a countermeasure, we used several sources for data collection, which included semi-structured interviews with different roles and internal documents of the cases. Furthermore, the interviews were transcribed and coded by one researcher and then reviewed by a second researcher. Moreover, key informants of the case organizations reviewed the results of our study.

Internal validity is not a concern since this study is exploratory and does not seek to establish causal relationships.

External validity refers to the generalization of the findings and the extent to which the results are of interest outside the investigated cases. To counteract this, we focused on the literal replication of our cases and on their analytical generalization.

Reliability refers to what extent the case study is conducted in a robust manner and whether replication by other researchers would yield the same results. To counter this, two researchers were always present in the interviews. Additionally, all reports sent to the companies were revised by another researcher and discussed with company representatives. Moreover, a case study database was created containing case study documents such as audio recordings, interview protocols, and slide decks.

VI. Conclusion and Outlook

Nowadays, organizations are urged to undergo large-scale agile transformations to respond readily to environmental changes [47], [48]. Architecture principles provide powerful simple-rule-like instruments to safeguard large-scale agile transformations by providing guidance for necessary coordination efforts [14]. Notwithstanding the importance of architecture principles to support large-scale agile transformations, extant studies on architecture principles still lack insights into how principles can be established. Against this backdrop, the main contribution of this study was to illuminate the establishment of architecture principles to support large-scale agile transformation through a multiple-case study of five German companies. In the following, we present the research questions and the answers to the questions.

RQ 1: How are architecture principles established to support large-scale agile transformations? RQ1 was addressed by revealing current practices of the case companies in formulating and using principles (see Section IV-B). Our findings show that EAs were responsible for the process of establishing principles. Our results also highlight that principles were communicated to ATs via communities, wikis or emails. We also found that the compliance checks on principles were mainly performed manually and sporadically.

RQ 2: Which challenges do organizations face when formulating and using architecture principles? Based on a cross-case analysis, we addressed RQ2 and identified twelve challenges the organizations faced in establishing principles (see Section IV-C). These included, amongst others, missing guidelines on how principles can be implemented or missing mechanisms for checking compliance of ATs with principles.

RQ 3: Which architecture principles are used to support large-scale agile transformations? Based on a structured literature review, we identified 30 architecture principles that were considered suitable for addressing RQ4 (see Section IV-D). Our findings show that the majority of the proposed principles were relevant to the case organizations and have already been largely implemented by the organizations.

Finally, this article leaves some room for further studies. We encourage researchers to perform explanatory case studies and to elaborate on the impact of the application of architecture principles on large-scale agile transformation outcomes. We also call for further case studies to reveal best practices for addressing our identified challenges.

Acknowledgements

This work has been partially sponsored by the *German Federal Ministry of Education and Research (BMBF)* via the Software Campus Project SaM-IT 01IS17049 and by the *Fonds National de la Recherche Luxembourg* (www.fnr.lu) via the ValCoLa project.

References

[1] P. Kettunen, "Adopting key lessons from agile manufacturing to agile software product development a comparative study," *Technovation*, vol. 29, no. 6-7, pp. 408–422, 2009.

[2] P. Kettunen and M. Laanti, "Future software organizations - agile goals and roles," *European Journal of Futures Research*, vol. 5, no. 1, p. 16, dec 2017.

[3] B. Sherehiy, W. Karwowski, and J. Layer, "A Review of Enterprise Agility: Concepts, Frameworks, and Attributes," *International Journal of industrial ergonomics*, vol. 37, no. 5, pp. 445–460, 2007.

[4] P. Besson and F. Rowe, "Strategizing information systems-enabled organizational transformation: A transdisciplinary review and new directions," *The Journal of Strategic Information Systems*, vol. 21, no. 2, pp. 103–124, 2012.

[5] D. Gerster, C. Dremel, and P. Kelker, "Agile Meets Non-Agile: Implications of Adopting Agile Practices at Enterprises," in *24th Americas Conference on Information Systems*, New Orleans, 2018.

[6] C. Fuchs and T. Hess, "Becoming Agile in the Digital Transformation: The Process of a Large-Scale Agile Transformation," in *39th International Conference On Information Systems*, San Francisco, 2018.

[7] T. Dingsøyr, N. Moe, T. Fægri, and E. Seim, "Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation," *Empirical Software Engineering*, vol. 23, no. 1, pp. 490–520, 2018.

[8] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87–108, 2016.

[9] Ö. Uludağ, M. Kleehaus, C. Caprano, and F. Matthes, "Identifying and Structuring Challenges in Large-Scale Agile Development Based on a Structured Literature Review," in *22nd International Conference on Enterprise Distributed Object Computing*, Stockholm, 2018.

[10] A. Scheerer, T. Hildenbrand, and T. Kude, "Coordination in large-scale agile software development: A multiteam systems perspective," in *47th Hawaii International Conference on System Sciences*, Hawaii, 2014.

[11] D. Kulak and H. Li, *The journey to enterprise agility: Systems thinking and organizational legacy*. Cham: Springer, 2017.

[12] Ö. Uludağ, N. Reiter, and F. Matthes, "What to Expect from Enterprise Architects in Large-Scale Agile Development? A Multiple-Case Study," in *25th Americas Conference on Information Systems*, Cancun, 2019.

[13] Ö. Uludağ, S. Nägele, M. Hauder, and F. Matthes, "Establishing Architecture Guidelines in Large-Scale Agile Development Through Institutional Pressures," in *25th Americas Conference on Information Systems*, Cancun, 2019.

[14] D. Greefhorst and E. Proper, *Architecture Principles The Cornerstones of Enterprise Architecture*. Springer, 2011.

[15] J. J. Korhonen and M. Halén, "Enterprise architecture for digital transformation," in *19th Conference on Business Informatics*, Thessaloniki, 2017.

[16] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, *Selecting Empirical Methods for Software Engineering Research*. London: Springer, 2008, pp. 285–311.

[17] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, p. 131, 2009.

[18] R. K. Yin, *Case Study Research: Design and Methods*. London: Sage Publications, 2008.

[19] T. C. Lethbridge, S. E. Sim, and J. Singer, "Studying software engineers: Data collection techniques for software field studies," *Empirical Software Engineering*, vol. 10, no. 3, pp. 311–341, 2005.

[20] R. Stake, *The Art of Case Study Research*. Thousand Oaks: Sage Publications, 1995.

[21] M. B. Miles, A. M. Huberman, and J. Saldana, *Qualitative Data Analysis: A Methods Sourcebook*. Thousand Oaks: Sage Publications, 2014.

[22] K. Rolland, B. Fitzgerald, T. Dingsøyr, and K.-J. Stol, "Problematizing Agile in the Large: Alternative Assumptions for Large-Scale Agile Development," in *37th International Conference on Information Systems*, Dublin, 2016.

[23] T. Dingsøyr, T. E. Fægri, and J. Itkonen, "What Is Large in Large-Scale? A Taxonomy of Scale for Agile Software Development," in *International Conference on Product-Focused Software Process Improvement*, Helsinki, 2014.

[24] T. Dingsøyr and N. B. Moe, "Towards principles of large-scale agile development," in *International Conference on Agile Software Development*, Rome, 2014.

[25] M. M. Lankhorst and H. A. Proper, "Agile Architecture," in *Agile Service Development: Combining Adaptive Methods and Flexible Solutions*, ser. The Enterprise Engineering Series, M. M. Lankhorst, Ed. Springer, 2012, ch. 3, pp. 41–57.

[26] M. Op 't Land, H. A. Proper, M. Waage, J. Cloo, and C. Steghuis, *Enterprise Architecture - Creating Value by Informed Governance*, ser. The Enterprise Engineering Series. Springer, 2008.

[27] D. Greefhorst and H. A. Proper, *Architecture Principles - The Cornerstones of Enterprise Architecture*, ser. The Enterprise Engineering Series. Springer, 2011.

[28] W. R. Ashby, *An introduction to cybernetics*. London: Chapman & Hall, 1961.

[29] G. Plataniotis, S. De Kinderen, Q. Ma, and H. A. Proper, "A conceptual model for compliance checking support of enterprise architecture decisions," in *17th Conference on Business Informatics*, Lisbon, 2015.

[30] M. Paauwe, "Understanding architecture principles as working mechanisms," in *8th Enterprise Engineering Working Conference*, Luxembourg, 2018.

[31] M. Op't Land and H. A. Proper, "Impact of principles on enterprise engineering," in *15th European Conference on Information Systems*, St. Gallen, 2007.

[32] R. Wagter, H. A. Proper, and D. Witte, "The extended enterprise coherence-governance assessment," in *7th Workshop on Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation*, Barcelona, 2012.

[33] IEEE, "Systems and software engineering – Architecture description," *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)*, pp. 1–46, 1 2011.

[34] The Open Group, *TOGAF Version 9*. Van Haren Publishing, Zaltbommel, the Netherlands, 2009.

[35] R. O. Briggs, G. J. d. Vreede, and J. F. Nunamaker, "Collaboration Engineering with Thinklets to Pursue Sustained Success with Group Support Systems," *Journal of MIS*, vol. 19, no. 4, pp. 31–63, 2003.

[36] J. Nabukenya, P. van Bommel, and H. E. Proper, "Repeatable collaboration processes for mature organizational policy making," in *14th International Workshop of Groupware: Design, Implementation, and Use*, Omaha, 2008.

[37] J. Nabukenya, P. v. Bommel, H. A. Proper, and G. J. d. Vreede, "An Evaluation Instrument for Collaborative Processes: Application to Organizational Policy–Making ," *Group Decision and Negotiation*, vol. 20, no. 4, pp. 465–488, 2011.

[38] The Open Group, *TOGAF Version 9.1*, 10th ed. Van Haren Publishing, Zaltbommel, the Netherlands, 2011.

[39] J. L. Dietz and J. A. Hoogervorst, "The principles of enterprise engineering," in *Enterprise Engineering Working Conference*. Springer, 2012, pp. 15–30.

[40] INNOQ, "Independent systems architecture - principles," 2018, accessed: 2019-05-27. [Online]. Available: https://isa-principles.org

[41] P. A. Khosroshahi, M. Hauder, A. Schneider, and F. Matthes, "Enterprise architecture management pattern catalog v2," Technical University of Munich, Tech. Rep., 2015.

[42] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, and Others, "Manifesto for Agile Software Development," https://agilemanifesto.org/, 2001.

[43] Ö. Uludağ, M. Kleehaus, X. Xu, and F. Matthes, "Investigating the role of architects in scaling agile frameworks," in *Enterprise Distributed Object Computing Conference (EDOC), 2017 IEEE 21st International*. IEEE, 2017, pp. 123–132.

[44] D. Marosin and S. Ghanavati, "Steering transformations with architecture principles," in *Architectural Coordination of Enterprise Transformation*. Springer, 2017, pp. 147–151.

[45] R. Winter and S. Aier, "How are enterprise architecture design principles used?" in *th Workshop on Trends in Enterprise Architecture Research*, Helsinki, 2011.

[46] D. Leffingwell, R. Martens, and M. Zamora, "Principles of agile architecture," *Leffingwell, LLC. and Rally Software Development Corp*, 2008.

[47] V. Sambamurthy, A. Bharadwaj, and V. Grover, "Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms," *MIS Quarterly*, vol. 27, no. 2, pp. 237–263, 2003.

[48] J. Highsmith and A. Cockburn, "Agile software development: the business of innovation," *Computer*, vol. 34, no. 9, pp. 120–127, 2001.