Everyone's Going to be an Architect: Design Principles for Architectural Thinking in Agile Organizations

Bettina Horlach
University of Hamburg
horlach@informatik.uni-hamburg.de

Andreas Drechsler
Victoria University of Wellington
andreas.drechsler@vuw.ac.nz

Ingrid Schirmer
University of Hamburg
schirmer@informatik.uni-hamburg.de

Paul Drews
Leuphana University of Lüneburg
paul.drews@leuphana.de

Abstract

Organizational agility is a prominent aim for companies to thrive in today's volatile business environments. One common building block of agility are (semi-) autonomous teams for continuously fulfilling and surpassing customers' needs. However, these teams still need to see the enterprise's 'big picture' of strategic objectives, business processes, and IT landscape to prevent organizational inertia or technical debt. This requires architectural thinking to inform these 'non'-architects' decision-making. To aid companies towards achieving sustainable agility, we propose six design principles as underlying logic on how to realize architectural thinking in agile organizations. The results are based on insights from interviews with sixteen employees and consultants with expertise on architecture management and organizational agility across several industries. Our work closes a gap in the agility literature, which so far mainly focused on non-generalizable blueprints for agile setups without showing their underlying logics, or approaches and role set-ups for enterprise-level architecture management.

1. Introduction

In today's hypercompetitive business environments with the power shifting to customers [1], more and more companies strive for organizational agility by becoming proactive in sensing customers' needs and responding with speed and innovation to fulfil and surpass customer demands (e.g. [2,3]). Otherwise, customers can often easily select another service (provider) from a vast array of opportunities in the market [1]. A common building block for increasing agility are (semi-)autonomous teams with high decision-making power and ownership for (parts of) services, their delivery and improvement [4]. These organizations seek to scale agile principles and values beyond the traditional agile 'realm' of software development to become an agile organization.

Despite the teams' autonomy, they need to be aligned with each other and the strategic objectives, as the organization may expect certain outcomes being delivered that may require collaboration across several teams [5]. As individual teams often have only a local view on 'their' services and aim to fulfil their goals, organizational mechanisms are needed for leveraging synergies and dependencies between teams and services [6,7]. The absence of such mechanisms may result in unsustainable decisions that may cause organizational inertia in the long-term [8], technical debt [9] and inconsistent, redundant, or conflicting solutions [10].

Both in traditional and agile organizations, Enterprise Architecture Management (EAM) is seen as one approach for addressing these issues [8,11,12]. Traditionally, EAM involves modeling, planning and controlling changes from an architectural perspective [13,14] in a top-down, centralized way with a dedicated function predefining architectural standards [15]. This enforcement-centric view of EAM, however, is at odds with the agile teams' autonomous nature and may create organizational inertia itself by preventing teams to respond to market changes based on their rules. With distributed decision-making in agile organizations, mostly by 'non'-architects, everyone instead of solely the EAM function [13] needs an 'architectural thinking' mindset [10] to consider the consequences of their actions for the whole organization in their decisions and balance the demands in a sustainable manner. Given the limited insights on how such an architectural thinking could be facilitated in agile organizations, we conducted an exploratory qualitative study to answer the question: *Which principles can foster architectural thinking in agile organizations to support organizational agility?*

The remainder of the paper is structured as follows. In section 2, we briefly describe organizational agility, architectural thinking and management as conceptual foundations for our analysis. Afterwards, we outline our research methodology. In section 4, we highlight our main results, the six principles for architectural thinking in agile organizations. Finally, we discuss our findings and conclude with future research opportunities.

HICSS

2. Research background

Organizational agility may involve optimizing existing service offerings to improve current market positions and innovating business opportunities for new product-market domains [16] for fulfilling agility's dimensions of sense and response [2,3,17]. Thus, agility implies three dimensions: 1) co-opting customers in exploring and exploiting opportunities to leverage their voice in service delivery, 2) leveraging the suppliers' and service providers' assets, knowledge, and competencies through partnerships and alliances for fast service delivery and new opportunities, and 3) a dynamic organizational and operational setup [2]. The latter implies scalability and (re)integration in processes, structures, and knowledge [2,17,18], so that organizational capabilities are in a constant flux [18,19].

Many companies started their agile journey with small agile teams to increase delivery speeds [4]. With digitalization deeply intertwining IT and business logic [20,21], these teams are increasingly becoming cross-functional with business and IT team members to bridge operational level gaps. To be effective, agility on the team level also requires a corresponding agile mindset, structures, and processes on the enterprise level to enable a fast and continuous delivery flow with as little friction as possible. The term 'agile mindset' means being aware to delight the customer with continuous learning and not only to adopt a set of agile practices [1]. A variety of frameworks for scaling agility, such as Disciplined Agile and the Scaled Agile Framework (SAFe), also provide structures and processes [22]. These are based on short and interconnected planning and feedback cycles for coordinating the teams' incremental work products. Various coordination mechanisms within and across all levels, like scaled product ownership and Scrum of Scrums can be set up to help linking the teams to the company's strategic objectives [6,7,23]. Nevertheless, companies strive for organizational agility in many ways, as the current debate on bimodal organizations shows [24,25].

Agility increasingly involves the use of information technology (IT) to provide strategic directions, which calls for a suitable IT infrastructure [2,3]. This is only realistic if everyone has a clear blueprint of the IT architecture and its link to business functionalities [11,26]. Some frameworks [22] and an increasing number of research approaches (e.g. [8,11,27]) refer to (enterprise) architecture management, traditionally an IT corporate function [13], as one main mechanism for designing and integrating this blueprint, e.g. via to-be architectures [14]. Recently, researchers pointed out that, in an agile world, the EAM role changes from enforcing technological standards, approving projects, and tracking changes in the enterprise architecture (EA)

enterprise-wide and on a detailed level to advising teams in their architectural decisions [12,27], resulting in a decentralized decision-making process [9,28]. Thus, the new EAM roles mainly focus on cross-team issues with harmonizing governance requirements across teams and guiding them through business and technical roadmaps [15,23]. Similar to traditional settings, most approaches propose two distinct architecture roles [26,27]: First, specific enterprise architects from IT can help resolving technical dependencies on a portfolio level and support shaping the overall strategic vision [9,12]. Second, strategic governance and corresponding solution/ system/ software architects or (chief) architecture owners on the program or team level can give guidance for individual programs, projects, or teams [9,23]. Team members, e.g., senior developers, with architectural knowledge mainly become responsible for architectural decisions within teams on a detailed level, as they are close to the code [11]. For overall coordination, e.g. for architectural guidelines [27], group-based decision-making [9], e.g. via communities of practice [27,29], is prominent, as it fosters everyone's commitment based on raised common concerns or domains [29].

While EAM provides knowledge about the whole organization, it is criticized in agile organizations as being too far from the actual delivery and planning and, thus, not valuable for stakeholders [10,11]. Especially local decision-makers on the operational level argue that EAM does not provide the necessary information [11]. Yet, sensing potentials and obstacles by knowing about (parts of) the system and their interdependencies is still essential, as working, fitting, and integrated services truly delight the customer [1]. This connected mindset of architectural thinking [10,13] is even more critical, as business and IT logic merge with digital [20,21], but governance and management structures are highly distributed. In addition, the link to the strategic objective is to ensure the fit of aspired services with the enterprise picture. While maintaining this mindset may involve the support of EAM functions, as research suggests [23,26,28], it is not discussed by architectural thinking approaches. Instead, architectural thinking mainly implies practices for considering holistic, long-term service aspects and fundamental system design and evolution principles in decision-making by 'non'-architects [13], which may be both from business and IT [10]. Yet, it remains unclear on how such an architectural thinking could actually be put into practice in agile organizations.

3. Research methodology

To contribute first insights on how agile organizations realize architectural thinking for

supporting organizational agility, we conducted an exploratory qualitative cross-industry study. The study participants were selected based on three criteria: First, they have experience in or are responsible for architecture within their organization or provide consulting services to clients with focus on architecture management. Second, the organization(s) the expert works in or consults for is undergoing a transformation towards organizational agility by reshaping (parts of) business and IT. Finally, the participants hold a position with in-depth insights on the overall organizational system. Table 1 gives an overview of all participants.

Table 1. Participants of the empirical study

Interview	Position	Main Industry Afilliation
INT-1	Program Manager [1]	Telco
INT-2	Program Manager [1]	Telco
INT-3	Chief Product Owner [1]	Telco
INT-4	Enterprise Architect [1]	Banking
INT-5	Enterprise Architect [2]	Banking
INT-6	PMO [2]	Energy
INT-7	PMO [1]	Government
INT-8	CEO Tool Vendor A [1]	Energy, Transport, Health
INT-9	CEO Tool Vendor B [1]	Government, Banking
INT-10	CEO Tool Vendor C [2]	Insurance, Telco, Energy
INT-11	Consultant [1]	Government, Telco, Banking
INT-12	Consultant [1]	Banking, IT, NGO
INT-13	Consultant [1]	Banking, Energy
INT-14	Consultant [1]	IT, Government, Banking
INT-15	Consultant [1]	Insurance, Utilities, Government
INT-16	Consultant [1]	Retail, Utilities, IT

Interview format: [1] Face-to-face meeting   [2] Video call

We used semi-structured interviews, preferably in face-to-face meetings (see Table 1), for a detailed exploration of the participants' experiences and views. Based on our understanding of architectural thinking and agility, we asked each participant to thoroughly describe their or their key clients' organizational setup and how decisions in relation to architecture are taken and realized. This includes planning and monitoring processes, procedures for design and documentation of decisions, and the roles involved. The interview sessions lasted 45-75 minutes and were audio-recorded and transcribed. If further details were required, additional interviews were conducted by phone or video call. For triangulation and further insights, we also reviewed public and private internal and external documentation. These included process specifications and architectural documentation such as meta-models or service designs, where permitted.

For our analysis, we conducted an abductive qualitative analysis inspired by the grounded theory coding process of open-axial-selective coding [30].

Initially, the first author assigned open codes to the transcripts such as "architecture as support function". In addition, we compared the codes based on the abstracts of the manifestation of organizational agility with sense and response and different process areas of EAM (plan, model, communicate, transform and document) in this stage. The codes then were constantly compared, which resulted in consolidated codes such as "groups for discussing architecture across teams" based on the identified common character of enabling synergies. To attain the final empirical results, we continued to iteratively consolidate the codes by their commonalities in relation to agility, until the final principles emerged. In case of conflicts, the authors discussed the different perceptions until they reached a joint assessment.

For generalizability and validity [31], we evaluated the principles with three additional experts working as architects in an agile organization. The participants were asked in face-to-face meetings for qualitative feedback on the principles' structure, applicability, level of detail, and utility [32]. We also requested feedback to identify further design principles to refine our results. However, the findings showed that our results are comprehensive; only minor revisions like extending some principles' descriptions in style and phrase were needed. Requested patterns on concrete recommended actions, however, were shifted to our next research steps.

4. Results

In this section, we describe the six identified design principles for organizational agility with architectural thinking (see Table 2). Similar to the seminal work for EAM principles by [33], we show the rationale behind each principle based on the goal of agility, and the resulting implications for shaping architectural thinking in form of proposed implementation mechanisms. Our principles are designed to be as independent as possible from organizational setups and specific reasons for pursuing agility, so that architectural thinking can be realized in various ways, e.g. via (agile) projects or via stable 'product teams' with an end-to-end focus on service delivery. While the principles show the general form and function [34], the mechanisms, however, act as exemplary representations we identified in the analyzed companies. Thus, organizations may choose to implement principles with different mechanisms.

4.1. Architect around the business ecosystem

The first identified design principle is to extend the EA perspective towards the surrounding business ecosystem to be able to sense environmental changes and respond in a timely manner. All interviewed experts

have declared in unison that today's environments require a thorough understanding across the organization of who the external customers are, what value is for them, and how they create value to be able to always deliver the 'right' product or service offerings. In other words, doing so would require companies to think from the external customers' perspective across business and IT. This implies identifying and understanding customers' experiences and behavior in day-to-day life and extracting the resulting customer needs and problems. In addition, this involves thinking about integration points of the customer value creation with the organization by identifying which parts of the overall value creation the company contributes. As INT-4 states, the question is: "*How do I look at that whole end to end flow across my organization from my customer, from an outside perspective?*" and analyze the resulting customer problem. According to the interviewees, companies should therefore be able to 'visualize' this information in order to communicate and subsequently act on it. Most interviewees spoke about the *customer value stream*, which then splits into multiple customer journeys that address specific customer needs, resulting in customer problems that need to be solved. Since (parts of) customer journeys

may rely on business partners, e.g., by offering their services (semi) exclusively on a platform, organizations may also be required to understand the partner journey(s) and continuously be aware of fulfilling the supplier experience to prevent weakening the affected parts of the customer value stream. Finally, companies need to monitor third parties such as regulators and auditors, as they also indirectly influence the customer value. Since most business ecosystems are highly volatile, identifying, monitoring, and analyzing the ecosystem needs to become a truly continuous activity.

4.2. Continuously map internal and external views

A prerequisite for acting on identified (changing) customer needs is the ability of organizations to continuously adjust their internal service design and delivery. First, from a strategic architecture perspective, this may require mapping the customer value stream, its journeys, and the integration points to the organizational value proposition(s), which define the internal promises by the organization towards the customer within one or multiple business models. Second, the interviewees recommend extending the mapping to the enterprise

Table 2. Design principles of architectural thinking for supporting organizational agility

Design Principle	Rationale by Organizational Agility	Main Implications for Shaping AT
DP1: Architect around the business ecosystem	– Understand customer value & its creation – Identify business partners' role in value creation – Continuously evaluate ecosystem for gaps	– Provide (linked) information about the ecosystem (e.g., customer value streams, customer & partner journeys)
DP2: Continuously map in- and external views	– Outline company's role in value creation – Ongoing mapping & gap analysis of external demands with company's value propositions and long-term strategic goals – Continuous mapping & gap analysis of external needs with operational internal delivery (e.g. service features)	– Provide (linked) information on enterprise vision, strategy, business model(s), external needs & problems – Integrate architecture in portfolio decision-making to analyze the link of (business) processes, capabilities & internal delivery with external needs
DP3: Create value-oriented architecture support	– Continuous alignment of internal service delivery to customer & business value – Ongoing monitoring that services fit to the expected value	– Support portfolio management in tailoring 'ideal' delivery organization – Put alignment mechanisms in place across the organization, e.g., shared purpose and metrics.
DP4: Empower local stakeholders to make architectural decisions timely	– Empower decision-making as much as possible within set frame (aligned autonomy) – Enable fast, but informed decision-making	– Decentralize architectural decisions as local as possible (e.g., skills in teams) – Ensure fast support across company by, e.g., shared architecture service function
DP5: Provide long-term guidance for continuous architecting	– Foster continuous improvement & innovation (service, business model, process) – Enable adjustments to portfolio in case of novel/ complex locally derived innovations that require overarching decisions	– Establish collaboratively built and easily adjustable architectural vision – Consolidate and integrate models and data from time to time, e.g., via chapters – Support exchange among 'architects'
DP6: Make architecture discussable and visible	– Identify dependencies and collaboration partners early as possible, e.g., to resolve issues – Prevent unnecessary rework based on misunderstandings	– Enable non-architects to understand architectural models – Make architecture decisions & rationales transparent & easy to find

vision as the organization's self-perception and to the corporate strategic goals to ensure that no frictions exist among the organization's strategic moves. The ongoing changes in the ecosystem lead to continuous mapping, however, and the enterprise vision may need adjustment over time. Third, mapping involves continuous analysis if the company has the right capabilities and corporate services for fulfilling and surpassing customer needs.

To respond with ease and speed, flexibility in the business architecture is perceived as necessary, as first setting up standard procedures and processes, and then mapping the service delivery around them is seen as too rigid for fast shifts. Instead, the key is to think about the 'ideal' internal responses to recurring customer needs and problems in one or multiple journeys, which we call 'solution thinking'. A solution is the (innovative) response to a slice of customer value by addressing one specific recurring customer need. It involves corporate services to offer to the customer, and the steps towards achieving the slice of customer value, called 'internal value streams'. Value streams are designed end-to-end, as INT-12 states: "*It starts with the customer and it ends with the customer and even areas like legal and finance and so forth, supporting them, are part of that same single value stream.*" The internal value stream uses and alters (parts) of business processes, includes required business capabilities, internal services, all innovation, design, development and delivery activities and the used (parts of) the application systems and infrastructure. Together with the needs and customer services, solutions form a new 'comprised entity' spanning across the whole EA and its borders to customers. In addition, they span across and expand the usual 'end-to-end logic' in software development and delivery (plan-build-run) which has traditionally been outside of EA's scope. Solutions are fluid in nature, as the response to the problem might change due to different needs. Thus, they span both capital (capex) and operational expenditures (opex) due to their end-to-end logic from a plan, build, and run view. With this overarching logic, solutions also merge traditionally separated business and IT thinking for service provision. Due to this novelty, interviewees call for new architectural representations for enabling such an integrated and holistic view from customer with external value streams and journeys to solutions with customer services, internal value streams and its elements. To date, no such representations are known.

Most interviewees stated that the responsibility for continuous mapping would be best placed within the portfolio management, as it is the link between solution ideas or concepts and their realization. However, the setup and focus will change, as INT-13 explains: "*I think the portfolio planning would take the place of strategic planning that we may call it today. So I think we scale up portfolio management to bridge across*

capex and opex. We staff, where was portfolio offices or perhaps EPMOs, with people who have high levels of business acumen, business management, strategic knowledge and management skills and less of the project management process staff, which you tend to see a lot of today." The all-embracing nature of the portfolio would then require an overarching vision of the organization, both from a business and a technological perspective, which makes architecture one integral component in the portfolio design and decision-making. While this may imply enterprise architect involvement, labelling the role 'value managers', who would need a strong architectural mind-set, would be more suitable for an emphasis on seeing and switching between the customer and the business value and its implications for necessary changes in the company's solution landscape.

4.3. Create value-oriented architecture support

Solely mapping the ecosystem and the strategic level is seen as insufficient for organizational agility, as this may prevent a seamless delivery because the operational level is still 'siloed'. Instead, the whole operational side of solution delivery should be aware of and be aligned with the customer needs in their daily operations and their long-term mind-set.

The first alignment mechanism involves the required work items to satisfy the customer needs, traditionally depicted in form of programs and projects. In line with solution thinking, organizations are moving towards thinking around 'for what' they are working (value) instead of planning concrete features. Value involves both 1) the external customer value, but also 2) the internal business value with brand and staff, 3) strategic, and 4) financial value. The gaps from mapping external and internal views (see section 4.2) are then contrasted to the value quadrants and result in "*a very clear vision statement, which is supported by perhaps a purpose statement. And out of that it has five or six strategic goals.*" (INT-13) Based on the goals, which serve as the overall purpose, the portfolio items – outcome-oriented changes or new solutions – are derived based on the thinking "*You've got 20 million, what is the best thing we can achieve? And that will deliver one or more contributions to those [four] value [..] quadrants.*" (INT-13) The delivery functions then pick up the items for value delivery and value capture via their backlogs. In the consequent backlog planning on operational level, value implies for all parties, especially those being involved in decision-making like product owners, to move away from the typical internal product focus based on "*How do I get the best, the most features of my product?*" (INT-15). For creating alignment throughout the organization, mechanisms with customer-oriented metrics and KPIs following approaches like Objective

and Key Results (OKR) can be used to map value to individual benefits and services. As those usually do not include other parts of a value stream, further illustrations for picturing the extract of the overall value creation – like business capability models – can also support.

Following the value stream logic, the second alignment mechanism is to 'group' teams around a common customer problem-based purpose, based on the overall solutions in the portfolio. Grouping in this sense implies having a shared mind-set among a team on what to achieve via the purpose. That does not necessarily imply a structural setup, although some companies start to restructure in this way with stable product teams. For larger solutions, teams are sometimes organized around domains or tribes as the collection of teams with the same purpose. Each domain includes a maximum of 100-150 people, since a higher number does not enable stable social relationships between its members. Within the tribe, teams are responsible for one or more services and systems, as INT-3 states: "*Every single customer that joins basically you then need to have in the CRM and you then use that to understand all their usage so that you can bill that. Those systems all play out that way. So from a systems perspective, if you follow along the [customer] journey, de facto it ends up becoming sort of product tribe. So some IT systems will be exclusive to a product tribe or [customer] segment tribe. But they will be all over that as well.*" For continuity, insights on changed customer needs shall flow back from teams to the portfolio for reconfiguration.

4.4. Empower local stakeholders to make timely architectural decisions

Time is a key building block of being agile. With respect to the organizational setup, time implies to ensure that decisions are made in a timely manner with as little friction as possible. Frictions can be prevented by empowering the autonomous teams to make decisions as local as possible and within the shortest possible timeframes (such as 2 or 4-week sprints). As INT-13 states, "*[..] the difference from today is that those self-managing teams are very, very clear on what it is they're trying to achieve. And again, because the highest leaders have set up strategic direction in five or six goals, crafted a number of business objectives and communicate that really well down for your organization. And at the same time emphasize their communication capabilities so that everybody in the organization has a very clear perspective on what it is that they as an individual and they as a team all be empowered and almost self-leading to a point [..].*" Cascading decisions can be translated to the architecture into a similar fashion, so that architectural decisions

should be made as local as possible, preferably within a team. To establish a certain coherence among local decisions, organizations can set up high-level business and technology-related overall architectural standards at portfolio level around security or the choice of cloud vendors, but these would usually be more outcome-oriented guidelines instead of concrete rules. Via the backlog items, these architectural guidelines / standards are then cascaded down do domains and teams, which then decide how to act upon them. Thus, agile organizations foster decentralization of architectural analysis, modeling, and decision-making as much as possible in order to reduce friction and to limit coordination efforts. Lean practices such as Hoshin Kanri, Kata, or the A3 method [35] can be used for local decision-making, as they capture "*What are the [customer] pain points? Based on these pain points, the definition of the current state, what's the definition of awesome? [..]. I need a definition of awesome. I need a target state of some sort, not just flail around and struggle with the current state, but dream a little bit what the target state might be like. And then based on that tension, identify some specific counter measures. What initiatives do you think would be helpful in getting you towards that target state? And then pick the most important out of those and break it down into three specific small tasks that you can do [..]*" (INT-14).

To ensure decentralized decision-making in a complex context, different architectural skill sets (e.g., those of software, business, or solution architects) ought to comprehensively grasp the implications. These skill sets should preferably be with people having these roles within the teams. The interviewees proposed two not mutually exclusive general possibilities to support the skill development or provide extra support for complex situations such as large cross-domain initiatives or projects, which are usually combined: 1) one or multiple shared services that are dedicated to architectural thinking or 2) access to all necessary architectural information (see also section 4.6 below). The first alternative seems to be the most prominent at the moment. Although slightly differing in design, most organizations have some 'architectural keeper' in place. As INT-6 states "*[…] you'll see somebody at or adjacent to the development team level. So quite often on the one level up from the development team […] you'd have some tech leads that say who would be doing some solution design. [And] we have an architect sitting there, so that they can have that discussion and negotiation with whatever role it is that's forming the product management function. So that you can be: Okay, you want to do this stuff and we understand why from a market perspective that makes sense. It order to achieve that, we need to invest in the platform that's going to slow down what you want to do. But then*

actually it will speed us up later. So you need the right voice at the right level to be able to have that discussion." However, most interviewees emphasize that everyone should be 'wearing an architecture hat', so that specific architect roles – especially for technology – are not always required. Moreover, people with architectural awareness are key, as INT-12 sums up: "*Architecture is a competency, not a function [...] There needs to be somebody with that competency, with the knowledge and view of architecture. And in the technical space, any reasonably experienced technologist should be able to step into that space.*"

4.5. Provide long-term guidance for continuous architecting

Even though agile teams are empowered with a high decision autonomy, unnecessary redundancies and resulting costs are to be avoided in agile organizations like in traditional ones. Thus, interviewees state that some form of overarching architectural vision and guidance is needed to prevent long-term deficiencies in the architecture. As INT-11 elaborates: "*Let's take the target architecture out and it's more than that target [..]. It's more nudging where things are going. But I think we still have matariki. We still have a north star [...] we do have long-term goals for architecture. Absolutely.*" This 'north star' is mainly business-related and represented in the portfolio with items depicting individual goals to be fulfilled. However, some interviewees also mention a more specific common technological vision to enable swift switching between teams by preventing a plethora of different tooling and technological bases and the resulting time-consuming efforts to learn the new skill.

The way of achieving a truly shared and sustainable architectural vision or guidance may involve a collaborative process to develop such a vision together across the organization instead of imposing one top-down. Besides Scrum of Scrums, a community of practice is the most common practice. These are groups with representatives from different teams, which have a collective view across all teams, define guidelines, and are accountable for them. Most also manage the personal development of architecture roles of members, as so-called 'chapters'. While the representatives may be in architectural positions, everyone with an architectural mind-set (who 'wears an architectural hat') is eligible as member. However, the participants' main job is outside the chapter, as INT-12 states: "*And it's part of other work. I've seen a situation where there was one person who was permanently and only in that community of practice, they ended up taking very much a librarian position as well. [..] So that single individual was very, very busy looking across everything [solely in*

this group] that was going on, coordinating a lot." Thus, people with architectural roles still ought to be hands-on working within teams or being an explicit shared service function instead of being solely documenting – and, thus, not value-contributing – 'PowerPoint architects'.

Since the main purpose of the teams as 'ear' of the customer is to deliver continuous improvement and innovation to always fulfill and surpass customer expectations, it's in the organization's interest to foster continuous architecture across all levels in and among teams and domains. This implies a need for a certain flexibility in the architectural vision. Chapters can play an important role in this regard, as they can check the new ideas, especially with new technologies involved, and can alter the guidelines based on their knowledge. Further, chapters may be innovation radiators themselves by proactively thinking about possible uses and 'business cases' for technology innovations. Teams then have the opportunity to choose to integrate these ideas into their own backlogs. If ideas are 'too new' or very complex, they could be radiated up to the domain backlog or even to the portfolio to be evaluated and perhaps selected as a (part of a) a new solution.

4.6. Make architecture discussable and visible

The new thinking embedded in the previous principles challenges traditional notions of architectural models and modelling, which are usually quite abstract and in a language that specifically addresses architects and their peers. Especially with team members and product owners from the business side and other non-architect roles involved in architecture decision-making, there ought to be ways to make the complexity of both business and technical architecture and their relations understandable and discussable. The specific format of how architecture should be represented was secondary for the interviewees. For them, the main issue is to enable everyone to grasp the chosen architecture representation format and therefore its content. Most interviewees propose value stream mapping as the high level business architecture representation. As INT-14 states: "*Hold on, we know this. There's value stream mapping, right? Value stream mapping for people that are familiar with the concept is tremendously powerful because essentially it says: How do I know how value gets to the customer? What needs to be sort of happening to get the value?*" Other existing approaches such as customer personas are often used alongside, as they can put the customer perspective in simple words. For defining the resulting workload, mechanisms from agile software development such as themes, epics, and features were frequently mentioned. Those approaches can describe an expected outcome as a representation of value, both from a business and technical perspective,

and from portfolio level (themes, epics) down to domain (epics, features) and team level (features, stories, tasks).

However, making architecture discussable has a second dimension: visibility of architectural decisions. As INT-11 states: "*But it's only when they took the time to stand back for a day, because there was lack of visibility of work and when you visualize the work, you can suddenly see that each development team is not asking much of infrastructure. But in the next three months, they're all asking stuff of infrastructure and there's this huge snowstorm of work on the infrastructure board.*" Therefore, formats such as big room planning or obeya rooms in particular, can be used within the organization. An obeya room (also called 'war room' or 'big room') is a room filled with all information that is relevant for decisions and managing a group. Information is depicted on the walls – usually by manual boards and paper or post-its – includes objectives, expected outcomes, actions, issues and metrics [36]. Everyone can walk into an obeya room to get information and find out the corresponding person or team(s) in case of identified dependencies. The rooms are usually installed on each level from portfolio to team to grasp the whole picture, especially when planning capacity and work. Thus, planning meetings with representatives from different groups are usually held in those rooms to have the underlying architecture present at any time when discussing and making decisions.

Finally, tool-based consolidation and integration of models and data from time to time can support both a common understanding on architectural decisions and a common information retrieval point, independent of a physical space. Usually conducted by architectural shared service members, they mainly store the data in knowledge management tools such as Microsoft SharePoint or Atlassian Confluence. Together with data automatically gathered from cloud services and development tools, teams can create a 'self-reporting architecture' based on that information for gaining insights on the underlying complexity of decisions. However, the most important overall issue is achieving architectural transparency. As INT-14 sums up: "*So if you are capturing it somewhere, then print it out, put it on the wall and make sure that you have people with pencils or pens or whatever and let them sketch and change it. So don't let it kind of linger in an information refrigerator, make it an information radiator.*"

5. Discussion and Conclusion

In this paper, we propose six principles for how agile organizations can enable architectural thinking for aiding the goal of organizational agility. The principles were developed in an exploratory empirical analysis of sixteen interviews with companies and consultancies in the private and public sector. Taken together, these principles are intended to support the realization of architectural thinking throughout an organization alongside its journey towards increased agility.

Realizing our six principles would lead to several changes to traditional set-ups for EAM functions. First, the architecture scope would change so that it extends end-to-end from the customers and partners to the underlying technical solution components. Second, architecture has now an increased role in and importance for strategic, tactical, and operational decision-making from the portfolio level downwards. Third, the architectural decision-making approach now includes a larger number of roles, and perhaps even puts 'traditional' enterprise architects into a supporting role. Lastly, all the new non-architect roles would have to be aided in their decision-making by making architecture and impacts of changes visible and discussable.

While there is an increasing number of approaches for architectural management in agile organizations, most of them only address parts of the 'big picture' – for instance, by proposing lightweight approaches [9,10] or by focusing on the interaction between dedicated architects and agile teams [14,15,27]. Some also show deeper insights on how an architecture management function changes in (singular) agile IT organizations [28]. In contrast, our principles address a set of aspects – both regarding the architecture content and the organizational setup – that may help to establish architectural thinking beyond traditional architecture functions or roles. The overarching link between external environment and internal organization, although increasingly addressed in theoretical concepts such as business ecosystem architectures [37,38] – but either without the notion of agility, or reduced to solely calling for such a partner-orientation without providing architectural solutions [2,28] – potentially reduces frictions between architecture layers (e.g. customer and business operations, or business operations and IT). This may foster sound architectural decisions, which may in turn help agile teams to accelerate their organization's ability to deliver new or changed services.

At its core, agile architectural thinking still requires stepping out of local decision-making contexts and thinking about the 'big picture' [11,12], but now with many more people and roles involved. Also, in agile architectural thinking the global architecture view is achieved not anymore through efforts by dedicated enterprise architects making sense of the architectural complexity as "*heroes [that] don't scale*" (INT-9), but through collaborative discussions on architectural issues among peers and across teams and domains (see also [14,28]). While some organizations may have (or still have) specific architectural functions as shared services

involved in and enforcing these conversations [8,27], we see organizations that tend to follow a new philosophy of architecture as a competency instead of a full-time role, which 'delegates' the responsibility to non-architects. Similar to existing literature [27,29], these organizations employ a group-based approach interlinked across the organization for gaining a consensus on architectural decisions, perhaps supported by a shared architecture vision. However, the approaches also differ, as the group is also used for personal development of the people wearing an 'architectural hat'. On the operational level, these people may be mainly (senior) developers, whereas on the strategic level they may be mainly portfolio or value managers. Our findings therefore also indicate a future point of convergence between strategic portfolio and value management on the one hand and strategic EAM on the other hand in the organizational agility context.

Increasing the level of agility also often leads to an increased speed and volume in changes in the service structure and landscape, which leaves the architecture to be in a continuously 'unfinished' state [9], similar to the organization itself being chronically unfrozen [18,19]. The decentralized innovation management empowers teams to continuously think about improving existing services and creating new ones (see also [24,25]). This essentially decentralizes the place of architectural thinking, which, at first, reduces the overall architectural transparency. In contrast to the traditional EAM literature, which advocates for documenting a complete as-is picture or having a detailed target architecture model, agile organizations respond to the puzzle of local architectural decisions, planning, and knowledge by merely striving for a shared outcome-oriented architectural vision and sufficiently detailed architecture models to support teams' architectural thinking and conversations. Both the vision and underlying architecture models are kept rather lightweight by focusing on only having the necessary information present, and in a format that is clear for many different backgrounds, since – as mentioned before – many non-architects are now involved in the architecture-related decision-making.

In line with business ecosystem research [37,38], we finally extend Sambamurthy et al.'s (2003) [2] agility types of customer and partner agility. We highlight the organization's continuous awareness of the entire ecosystem surrounding an organization (including – but not limited to – customer and partner actions, behaviors, and needs), and the resulting identification of all these changing needs as potential drivers for a subsequent rapid internal response. Here, our notion of solution thinking provides a specific mechanism for bridging this continuous external awareness via thinking from a customer problem perspective within the whole organization and channeling the gained insights towards actual internal strategy as well as operational service (re)design and delivery. Our findings illustrate that a corresponding architectural representation can help substantially to grasp the complexity in the corresponding decision-making processes, and also to identify the actual gaps between the various external needs and the internal capabilities in the process of the delivery of a new or changed product or service that successfully addresses the changed needs. We therefore also extend the traditional EAM scope, which usually does not consider the wider business ecosystem [26,28].

Of course, our research is not without limitations. Most importantly, we build our results on knowledge gained in an exploratory empirical study with representatives from multiple organizations in a single country. To address the resulting generalizability and validity issues, we recruited interview partners from a variety of industries and job positions. In addition, we deliberately included consultants in our study to use their experiences with different clients (and, with some, in different countries as well) to gain an even broader view on architectural thinking and organizational agility. For achieving a further generalizability of our results, we plan to extend our study to multiple countries as a next research step. We also strongly encourage future studies to empirically validate our findings.

A second limitation of our research is the complexity of architectural management in agile organizations. With architecture being a multi-level concept spanning all organizational levels, and agile organizations focusing on decentralization as much as possible, many different roles are involved in the decision-making. Although we addressed this limitation with interviewing multiple perspectives in our study, the views from a number of roles such as (chief) product owners and agile team members are still missing. Thus, we recommend corroborating our findings with in-depth case studies from multiple perspectives. This could also lead to further insights on the interrelations of architecture with strategic management, especially strategy formulation and portfolio management, in agile organizations.

6. References

[1] S. Denning, "Agile's Ten Implementation Challenges", Strategy & Leadership 44 (5), 2016, pp. 15–20.
[2] V. Sambamurthy, A. Bharadwaj, and V. Grover, "Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms", MIS Quarterly 27 (2), 2003, pp. 237–263.
[3] O.-K. D Lee,. V. Sambamurthy, K. H. Lim, and K. K. Wei. "How Does IT Ambidexterity Impact Organizational Agility", Information Systems Research 26 (2), 2015, pp. 398–417.
[4] N.B. Moe, B. Dahl, V. Stray, L.S. Karlsen, and S. Schjødt-Osmo, "Team Autonomy in Large-Scale Agile", 52nd Hawaii

International Conference on System Sciences, 2019, pp. 6998-7006.

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

[6] Disciplined Agile, Disciplined Agile Framework. http://www.disciplinedagiledelivery.com (accessed 05/30/19)

[7] Scaled Agile, Scaled Agile Framework. http://www.scaledagileframework.com (accessed 05/30/2019).

[8] R. Parsons, "Enterprise Architects Join the Team," IEEE Software 22 (5), 2005, pp. 16-17.

[9] A. Martini, and J. Bosch, "A Multiple Case Study of Continuous Architecting in Large Agile Companies: Current Gaps and the CAFFEA Framework", 13th Working IEEE/IFIP Conference on Software Architecture, 2016, pp. 1–10.

[10] S. Aier, N. Labusch, and P. Pähler, "Implementing Architectural Thinking: A Case Study at Commerzbank AG", CAiSE 2015 Workshops, 2015, pp. 389-400.

[11] M. Canat, N. P. Català, A. Jourkovski, S. Petrov, M. Wellme, and R. Lagerström, "Enterprise Architecture and Agile Development: Friends or Foes?", 22nd International Enterprise Distributed Object Computing Workshop, 2018, pp. 176–183.

[12] S. Hanschke, J. Ernsting, and H. Kuchen, "Integrating Agile Software Development and Enterprise Architecture Management", 48th Hawaii International Conference on System Sciences, 2015, pp. 4099–4108.

[13] R. Winter, "Architectural Thinking", Business & Information Systems Engineering 6, 2014, pp. 361–364.

[14] M. Hauder, S. Roth, C. Schulz, and F. Matthes, "Agile Enterprise Architecture Management: An Analysis on the Application of Agile Principles", International Symposium on Business Modeling and Software Design, 2014, pp.1-10.

[15] Ö. Uludağ, M. Kleehaus, N. Reiter, and F. Matthes, "What to Expect from Enterprise Architects in Large-Scale Agile Development? A Multiple-Case Study", 25th American Conference on Information Systems, Cancun, 2019, pp.1-10.

[16] J. G. March, "Exploration and Exploitation in Organizational Learning", Organization Science 2 (1), 1991, pp. 71–87.

[17] E. Overby, A. Bharadwaj, and V. Sambamurthy, "Enterprise Agility and the Enabling Role of Information Technology", European Journal of Information Systems 15 (2), 2006, pp. 120–131.

[18] K. Conboy, and B. Fitzgerald, "Toward a Conceptual Framework of Agile Methods: A Study of Agility in Different Disciplines", ACM Workshop on Interdisciplinary Software Engineering Research, 2004, pp. 37–44.

[19] K. M. Eisenhardt, and J. A. Martin, "Dynamic Capabilities: What are They?", Strategic Management Journal 21 (10), 2000, pp. 1105–1121.

[20] A. Bharadwaj, O. A. El Sawy, P. A. Pavlou, and N. Venkatraman, "Digital Business Strategy: Toward a Next Generation of Insights", MIS Quarterly 37 (2), 2013, pp. 471–482.

[21] V. Nissen, and A. von Rennenkampff, "Measuring the Agility of the IT Application Systems Landscape", 13th Internationale Tagung Wirtschaftsinformatik, 2017, pp. 425–438.

[22] B. Horlach, T. Böhmann, I. Schirmer and P. Drews, "IT Governance in Scaling Agile Frameworks", Multikonferenz Wirtschaftsinformatik, 2018, pp. 1789–1800.

[23] Ö. Uludağ, M. Kleehaus, X. Xu, and F. Matthes, "Investigating the Role of Architects in Scaling Agile Frameworks", 21st International Conference on Enterprise Distributed Object Computing, 2017, pp. 123–132.

[24] I. Haffke, B. Kalgovas, and A. Benlian, "Options for Transforming the IT Function Using Bimodal IT", MIS Quarterly Executive 16 (2), 2017, pp. 101–120.

[25] B. Horlach, P. Drews, I. Schirmer, and T. Böhmann, "Increasing the Agility of IT Delivery: Five Types of Bimodal IT Organization", 50th Hawaii International Conference on System Sciences, 2017, pp. 5420–5429.

[26] T. Fallmyr, and B. Bygstad, "Enterprise Architecture Practice and Organizational Agility; An Exploratory Study", 47th Hawaii International Conference on System Sciences, 2014, pp. 3788–3797.

[27] Ö Uludağ,, S. Nägele, and M. Hauder, "Establishing Architecture Guidelines in Large-Scale Agile Development Through Institutional Pressures", 25th Americas Conference on Information Systems, 2019, pp. 1-10.

[28] P. Drews, I. Schirmer, B. Horlach, and C. Tekaat, "Bimodal Enterprise Architecture Management: The Emergence of a New EAM Function for a BizDevOps-Based Fast IT", 21st International Enterprise Distributed Object Computing Workshop, 2017, pp. 57–64.

[29] M. Paasivaara, and C. Lassenius. "Communities of practice in a large distributed agile software development organization–Case Ericsson." Information and Software Technology 56 (12), 2014, pp. 1556-1577.

[30] Strauss, A. L. and J. M. Corbin, Basics of Qualitative Research: Grounded Theory Procedures and Techniques, SAGE Publications: Newbury Park, 1991.

[31] I. Benbasat, and R. W. Zmud,. "Empirical Research in Information Systems: The Practice of Relevance", MIS Quarterly 23 (1), 1999, pp. 3–16.

[32] A. R. Hevner, S. T. March, and J. Park, "Design Science in Information Systems Research", MIS Quarterly 28 (1), 2004, pp. 75–106.

[33] G. Richardson, B. Jackson, and G. Dickson, "A Principles Based Enterprise Architecture: Lessons from Texaco and Star Enterprise", MIS Quarterly (14) 4, 1990, pp. 385–403.

[34] S. Gregor, and D. Jones, "The Anatomy of a Design Theory", Journal of the Association for Information Systems 8 (5), 2007, pp. 312–335.

[35] Bicheno, J., and M. Holwe, The Lean Toolbox, Picsie Books: Johannesburg, 2016.

[36] Morgan, J., and J. K. Liker, The Toyota Product Development System, Taylor & Francis: London, 2006.

[37] P. Drews, I. Schirmer, B.Horlach and C. Tekaat, "From Enterprise Architecture to Business Ecosystem Architecture: Stages and challenges for extending architectures beyond organizational boundaries," 18th International Enterprise Distributed Object Computing Workshop, 2017, pp. 57–64.

[38] Sebis, Business Ecosystem Modeling and Visualization. https://wwwmatthes.in.tum.de/pages/xr8agrmcuboh/Business-Ecosystem-Modeling-and-Visualization. (accessed 05/30/2019).