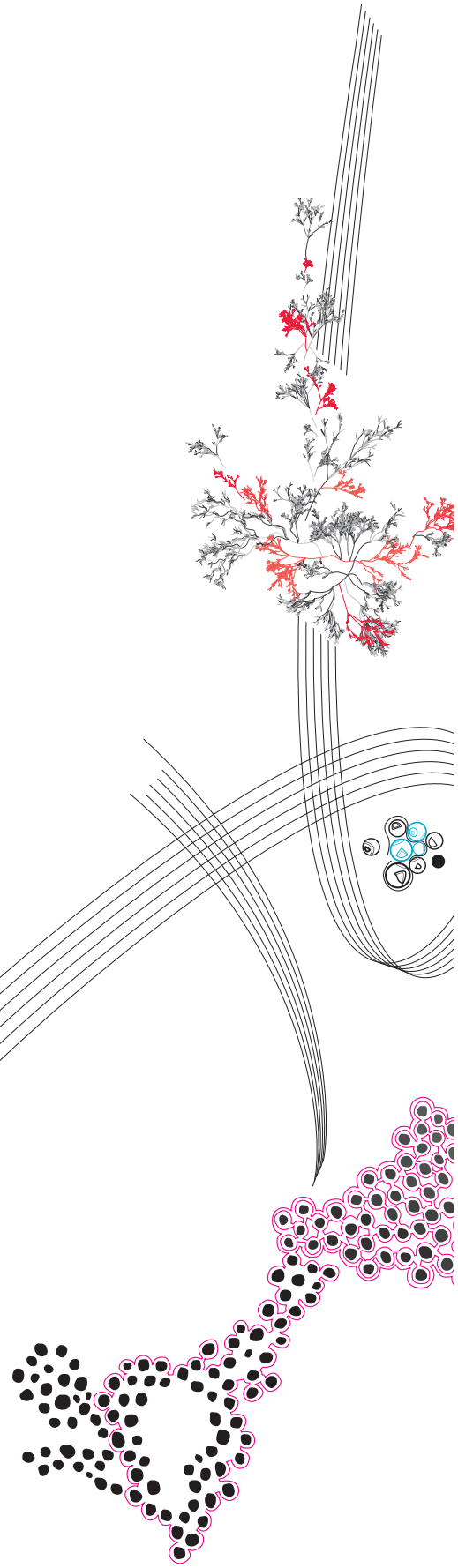

MSc Business Information Technology

Master Thesis

A New Approach for Modern Enterprise Architecture for Technology-native Organizations

Kimia Mahjoub

University Supervisors:

Dr. R. Guizzardi - Silva Souza

Dr.ir. M.J. van Sinderen

SAP LeanIX Supervisor:

Dr. A. Aldea

March, 2025

Department of Business Information Technology
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

SAP LeanIX

Abstract

This study explores a new approach to Enterprise Architecture (EA) tailored for technology-native organizations by integrating agile principles into EA to have flexibility, scalability, and governance at the same time. A design science research approach was used to identify the problem by gaining information from the theoretical perspective on the challenges, motivations, and required changes of agile EA. In addition, interviews among SAP LeanIX stakeholders were conducted, which provided a practical perspective to identify the problems of existing frameworks and tools related to EA and the issues of integration with agile methodologies. Based on the findings of the research topic and the interviews, a new EA method was designed that incorporates agile practices. This method was implemented in the SAP LeanIX workspace. The usefulness of the designed method was evaluated using expert feedback. According to the results from the evaluation, the designed method proved useful and easy to use for technology-native companies.

Key words: Enterprise architecture (EA), Agile principles, Technology-native organizations, SAP LeanIX

Contents

1 Introduction

This chapter provides an introduction to the research, including an explanation of its background, problem statement & motivation, research questions & objectives, and finally the research methodology used for this study.

1.1 Background & context

In today's rapidly evolving business landscape, organizations must constantly adapt to changing circumstances, including technological advancements, regulatory shifts, and market dynamics. Enterprise architecture (EA) serves as a strategic framework for managing these transformations, offering a structured approach to align business strategies, processes, technology, information, and organizational goals [1]. Its main purpose is to create a clear blueprint that outlines the structure and operations of the organization, including its business processes and IT systems [2]. Furthermore, EA promotes collaboration across teams by standardizing processes and tools, ensuring that everyone in the company follows the same guidelines tailored to the specific needs of the business [3]. By implementing EA, organizations can identify requirements, address system challenges, and develop solutions. Despite its benefits, EA also faces several challenges such as complexity, the time and effort required for its development and maintenance, limited user acceptance, and costly, slow implementation processes, which can sometimes lead to project failures [4].

One way to implement EA is through SAP LeanIX. SAP LeanIX is a SaaS application that allows you to manage and optimize an organization its EA [5]. It helps with the modernization of IT landscapes and continuous business transformation. SAP LeanIX offers a data-driven and automated approach enhanced with AI, allowing organizations to make decisions and collaborate more effectively [6]. SAP LeanIX offers its service to various customers such as traditional organizations and technology-native organizations.

Traditional organizations follow traditional business models and are often organized with clear hierarchies, governance, and centralized decision-making [7, 8]. Technology-native organizations on the other hand, typically operate with agile and scalable business models that leverage digital technology. These types of organizations prioritize flexibility, collaboration, innovation, and knowledge-sharing [8].

To support this need for flexibility and adaptability, many technology-native organizations adopt Agile methodology. Agile methodology is a project management framework that breaks projects down into iterative phases, commonly known as sprints [9]. Its iterative and incremental approach enables short feedback loops, which in turn results in better stakeholder collaboration and better adaptability to changing requirements [9, 10].

Comparing EA with Agile methodology highlights key differences in their approaches. EA provides a strategic long-term framework for aligning business strategies, processes, technology, and organizational goals [2]. In contrast, Agile focuses on short-term, iterative project delivery [9]. The comparison also shows that these two approaches may hinder their integration. While EA takes a top-down approach, shaping an organization's overall structure through long-term planning [11], Agile works at the project level, breaking down tasks into smaller parts to speed up development through short-term planning.

1.2 Problem statement

While EA provides a structured framework for aligning business strategies, processes, and technology [2], its traditional implementation is often rigid and slow to adapt [4]. This creates challenges for technology-native organizations, which prioritize agility, modularity, and rapid change [8]. Technology-native organizations operate in dynamic environments using continuous integration, automation, and iterative development to keep up with their fast-changing business goals. However, traditional EA practices do not fully support these needs, as they focus on stability and long-term planning.

Because of this, technology-native organizations are not being able to manage their EA in the way they need to and there is a need for a more modern EA design. This study specifically explores the challenges that technology-native organizations face in implementing EA, and how EA can be adapted in a more agile and flexible manner.

Summarizing this, the problem can be formulated into the following problem statement: **The current state of EA practices is not well-aligned for technology-native companies**.

1.2.1 Company assignment & motivation

Motivated by the problem mentioned in section 1.2, SAP LeanIX would like to explore how a modern EA practice can facilitate technology-native organizations. SAP LeanIX offers use cases and solutions for traditional companies' challenges such as moving to the cloud, but lacks in use cases for technology-native companies. SAP LeanIX its current challenge is to provide better services to technology-native companies using their tool. The project aims to develop a new method for EA that aligns EA practices with agile methodologies.

SAP LeanIX has identified two deliverables for this thesis assignment:

- An artifact that describes the modern EA practice that can support technology-native organizations.
- An implementation of this modern EA practice in an SAP LeanIX prototype.

1.3 Research questions & objectives:

In the previous section, the problem statement has been discussed. The overall goal of this study is to develop a modern EA method that can support technology-native organizations by making EA practice more agile and flexible. This section includes the main research question, sub-questions, and objectives as follows:

Main research question

- *How can the EA process be improved for technology-native organizations by designing an EA method that satisfies the requirements of agility, scalability, and governance to help stakeholders create an adaptable and well-managed architectural framework?*

To address the main research question, the following sub-questions have been formulated:

Sub-research questions

1. *What is the state of the art on agile principles in EA?*

   This research question aims to analyze the current methodologies in this field. The goal is to identify the challenges, methods, or frameworks that offer any solution for agile EA or their combination. This study aims to analyze the gap in the existing literature.

2. *How can a modern EA method be designed to support technology-driven organizations?*

   This research question aims to identify the requirements for designing an EA method that covers agile principles which will be useful for technology-native companies.

3. *To what extent does the designed EA method help technology-native organizations to work better in agile environments?*

   The research question aims to evaluate how useful the designed method is in helping technology-native organizations.

1.4 Research methodology

This study uses a design science approach from the book Design Science Methodology for Information Systems and Software Engineering by R.J. Wieringa [12]. The design science approach consists of the design cycle and the engineering cycle. The engineering cycle, as seen in Figure 1.1, is a problem-solving process that has the following main steps:

- Problem investigation: What phenomena must be improved and why?
- Treatment design: Design one or more artifacts that could treat the problem.
- Treatment validation: Would the designs treat the problem?
- Treatment implementation: Treat the problem with one of the designed artifacts.
- Implementation evaluation: How successful has the treatment been?

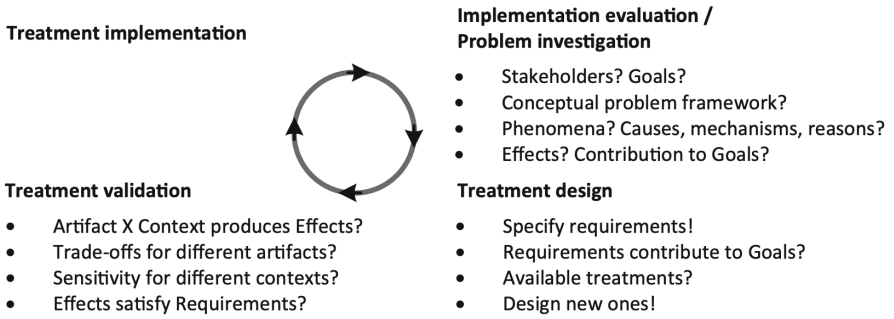

FIGURE 1.1: The engineering cycle created by R.J. Wieringa in the book Design Science Methodology for Information Systems and Software Engineering [12].

The design cycle is a process within the engineering cycle where a designed and validated treatment is implemented in the problem context, and the implementation is evaluated [12]. The design cycle includes the problem investigation, treatment design, and treatment validation.

This study follows the design cycle as the research method and uses it in the following chapters. Chapters 2 and 3 focus on problem investigation combining theoretical and practical perspectives. In Chapter 2, a literature review explores the existing research and theoretical foundations related to this topic. In Chapter 3, information is gathered in practice and the real world through practical insights from interviews. Chapter 4 focuses on treatment design based on insights from these interviews and the literature review. Chapter 5 shows a use case implementation using the designed method. Chapter 6 focuses on validation, using expert opinions. The research approach is designed to balance practical feasibility with depth of investigation within the time frame that has been given.

2 Problem investigation in theory

This chapter examines the current landscape of agile principles and EA to address the first sub-research question. This chapter is divided into two parts. In the first part, information for problem investigation from a theoretical perspective is gathered by conducting a literature study. This part is covered in Section 2.1, Section 2.2, and Section 2.3. The second part provides further background information that explains key concepts related to the problem investigation both in theory and practice. This part is covered in Section 2.4.

2.1 Systematic literature review approach

This section aims to provide background knowledge on the agile method for EA by analyzing relevant studies to help investigate the problem. A systematic literature review (SLR) has been conducted to find key insights such as challenges, motivation, required changes for adopting agile EA, collaboration between teams, and methods that are used in agile EA. Additionally, research papers that have been reviewed during the SLR, but were less relevant to this study, are included in Appendix A.

2.1.1 Methodology

SLR is a method that identifies and evaluates all available research relevant to a specific research question or topic area. It ensures that the review is reliable, unbiased, and comprehensive. SLR has three main phases: planning, conducting, and reporting the review [13]. Figure 2.1 shows the main steps of each phase used in this study.

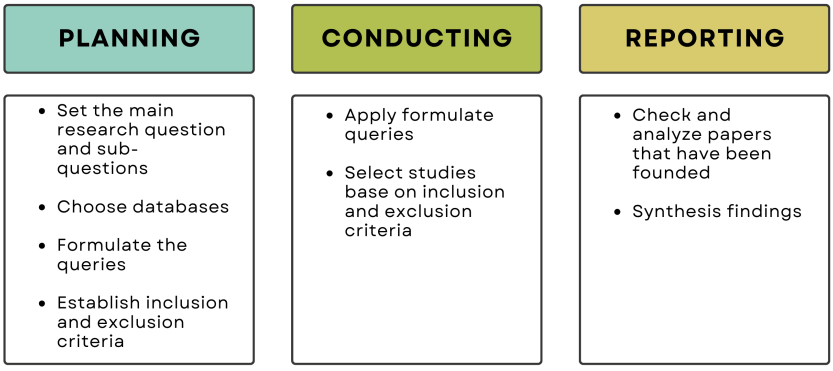

FIGURE 2.1: The three phases within SLR and their main steps used in this study, based on the procedures for performing systematic reviews by B. Kitchenham [13].

2.1.2 Literature review questions

The first sub-question formulated in Section 1.3 serves as the main research question for this literature review:

What is the state of the art on agile methods for enterprise architecture?

To explore this question from different perspectives, the following sub-questions have been formulated:

1. What are the challenges and motivations for adopting agile methods in enterprise architecture?
2. What changes are needed to adopt agile methods in enterprise architecture?
3. How can these meet agile teams' expectations for collaboration in large-scale development?
4. Which methods, models, or frameworks are used in agile enterprise architecture?

2.1.3 Search strategy & Study selection

Two public databases have been used in this study to find the most appropriate and relevant papers. These databases are:

- Scopus
- ScienceDirect

Both Scopus and ScienceDirect provide a broader range of research metrics, allowing access to different valid source types such as journals and conference proceedings. Three queries have been tried to find the best query for this study. These queries were as follows:

1. **("Enterprise architecture" OR "EA") AND ("agile methods")**
2. **("Enterprise Architecture" OR "EA") AND (“agile method” OR “agile software development” OR “agile digital transformation”)**
3. **("Enterprise Architecture" OR "EA" OR “agile enterprise architecture”) AND (“agile method”) AND (“software development”)**

These queries will be referred to as Query 1, Query 2, and Query 3, respectively.

Query 1 was selected as the final query for this study because Query 2 resulted in only a few relevant papers, which were already included in the results of Query 1, and Query 3 resulted in a lot of similar papers as Query 1. However, the results of Query 1 were most relevant for this study.

After searching Query 1 in the databases, a total number of 437 results were found, of which 313 of these documents were from Scopus and 124 documents were from Science Direct.

The potentially relevant studies that have been found, were assessed to determine their actual relevance. The goal was to ensure that the selected studies were unbiased and on predefined criteria related to the research questions. For study selection, it is important to set inclusion and exclusion criteria [13]. The inclusion and exclusion criteria used in this study were as follows:

Inclusion criteria

1. Inclusion of literature in the English language.
2. Inclusion of literature from all years (no filter for years).
3. Inclusion of study areas within computer science, business management and accounting, engineering, decision sciences, and social sciences.
4. The inclusion of source type is limited to journals, books, and conferences.

Exclusion criteria

1. Exclusion of duplicate literature in databases.
2. Exclusion of literature that has inadequate information. Some of them have the same keyword. However, they are not useful for answering the research questions.

These criteria were applied to narrow down the number of literature papers found using Query 1. After the initial filtering, the abstracts and conclusions of the remaining papers were reviewed to check their relevance to the sub-questions. This strategy identified the most relevant papers which were saved for further analysis. Moreover, in cases of duplicate literature, where the same papers were updated, the most recent and updated versions were used for the study.

After filtering the inclusion and exclusion criteria, there were 13 papers from Scopus and 5 from ScienceDirect. After identifying these papers, a snowballing technique offered by Wohlin [14] was applied to discover additional relevant papers from the references of the selected papers. After applying snowballing, 7 new papers were added to the selection. So, in total 25 papers are useful for answering the research questions. A diagram of this process can be seen in Figure 2.2.

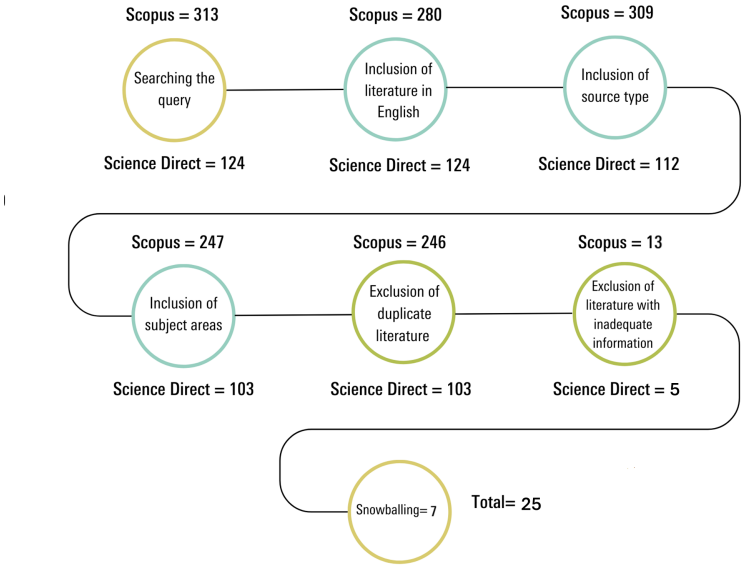

FIGURE 2.2: A diagram of the paper selection process based on the application of the inclusion and exclusion criteria.

The SLR first selected 25 papers but during the course of this study, 5 papers were found less relevant and moved to the Appendix A, resulting in 20 relevant papers for this literature review.

2.1.4 Study quality assessment

Quality assessment helps choose the best studies and understand how the quality of these studies might influence the final results. To assess the quality of the chosen studies, the following questions were asked:

1. Do the chosen studies provide precise answers?
2. Do the chosen studies provide answers that apply to the context of concern (e.g. technology-native companies)?
3. Do the chosen studies provide evidence for the provided answers?
4. Do the chosen studies provide guidelines for implementation?
5. Do the chosen studies report on implementation evaluations (i.e. applications in practice)?

Table 2.1 lists all 20 relevant papers selected to answer the sub-questions. It shows which sub-questions each paper addresses, with an "X" showing that the paper answers a question.

Ref.	Question 1	Question 2	Question 3	Question 4	Question 5
[Hauder et al., 2014]	X		X		
[Wessel et al., 2021]	X		X		
[Akinpelu et al., 2021]	X				
[Canat et al., 2018]	X		X		X
[Uludag et al., 2019]	X		X		X
[Askarinejad, 2012]	X		X		
[Buckl et al., 2011]	X			X	
[Nakayama et al., 2021]	X		X		
[Hopkins & Harcombe, 2014]	X				
[Guo et al., 2021a]	X				
[Guo et al., 2021b]	X		X		
[Cammin et al., 2021]	X		X		X
[Uludag et al., 2021]	X		X		
[Uludag et al., 2019b]	X	X	X	X	
[Hanschke et al., 2015]	X	X	X	X	X
[Kornyshova & Deneckère, 2022]	X				
[Uludag et al., 2022]	X				
[Duijs et al., 2018]	X				
[Medeiros et al., 2021]	X	X	X	X	X
[Medeiros et al., 2017	X	X		X	

TABLE 2.1: Quality assessment questions

2.2 Systematic literature review result

2.2.1 Challenges and motivations in adopting agile for EA

Motivation

According to Hopkins et al. [15], the balance between architecture and agility in large complex projects is important. Effective architecture is crucial for enabling agile development. Still, the important thing about it is to use "just enough" architecture to guide the project without overwhelming it with unnecessary complexity or processes. Traditional approaches where architects focus too much on theoretical designs. However, an architect is still crucial in large agile projects because they can view problems from multiple perspectives. The architect needs to specify key perspectives (like data, function, infrastructure, and integration) but only find those necessary for the project's success. This study recommends starting with a risk-based analysis to dive into complex areas early and perform proactive testing. This approach has some benefits which are:

- Reducing costs by allowing off-shore development and focusing on Total Cost of Ownership (TCO)
  - In off-shore development, where all parts of a project are outsourced to lower-cost regions, can work better if all teams operate as a unified group across locations so organizations need a single structure with consistent roles, processes, and tools in all locations. A global management team and regular executives should be sure that they are aligned with business goals. This approach doesn't have communication issues common in isolated teams and brings cost-saving benefits.
  - For reducing TCO in agile projects, they suggested designing systems that are easy to maintain and update over time. Key techniques include

avoiding duplicate systems, reusing existing services, standardizing tools and skills to simplify maintenance, and making systems easy for less specialized users to update.

- Minimizing rework through foresight and prototyping
  - Agile methods encourage starting with simple solutions and adding complexity as needed later, this assumes changes will always be easy and inexpensive, which isn't always the case, especially for core system components. Changes in these areas can become costly so architecture should predict where changes may be hard in the future and plan for them beforehand. They should use technology that allows changes to be made easily, add layers to protect the system from disruptions, and use prototypes to test ideas early before committing to them.
  - Prototypes help teams to figure out the issues. one type of prototype is high-fidelity which is more detailed than the final product. they are designed so that parts can be used in the final system. The other type of prototype is low-fidelity which is simpler and is for quick test concepts and feedback.
- Speeding up delivery by making the development process faster and smoother and allowing different parts of the project to work at the same time, testing early, and using automated tools to check for problems.
  - Speeding up delivery by incorporating multiple perspectives, one issue is agile bubble anti-pattern. Early progress is fast but ignoring potential problems can lead to expensive fixes later.
  - Maximizing capacity; good architecture can help to divide the system into smaller parts so different teams can work on them in parallel which can speed up the development.
  - Early integration; Teams should focus not only on building new systems but also on existing ones which can help to use information from the early systems. teams should incorporate and test the new system with the existing one early enough so teams can fix issues before they become bigger issues later on.
  - Early and continuous testing; teams should start testing as soon as two connected components are ready because late-stage testing is expensive and time-consuming.
  - Automated deployment; using the same deployment tools for both testing and production environments can help to increase the speed for delivery in agile projects. Regular use of automated scripts ensures smoother and more reliable final deployments.

Table 2.2, shows the summary of various motivations for adopting agile EA:

Category	Description	References
Cost Reduction	Using agile EA to reduce overall costs, especially in global development and total cost of ownership	Hopkins et al. (2014), Uludag et al. (2022)
Faster Delivery	Agile enables quicker delivery by allowing teams to work in parallel and continuously integrate	Hopkins et al. (2014), Wessel et al. (2021)
Minimizing rework	Explaining the use of foresight and prototyping to minimize rework in agile EA. It covers how starting with simple solutions and gradually adding complexity, alongside testing ideas with prototypes, helps teams plan for potentially costly changes	Hopkins et al. (2014)

Table 2.2: Summary of Motivations for Adopting Agile EA

Challenges

Based on the research of Hauder et al. [16], and Wessel et al. [17], three challenges in adopting agile for EA can be identified. The first challenge is finding a balance between long-term and short-term approaches. Traditional EA management focuses on creating a stable, top-down architectural vision that supports long-term business goals. However, agile methods focus on being responsive and quickly adapting to changes in the business environment, new technology, and market demands. This difference makes it challenging for organizations to have a clear and long-term vision and be flexible enough to respond to instant needs at the same time. The tension between these two approaches brings a challenge because organizations should find a way to integrate long-term strategic goals with the need to respond rapidly to changes in the business environment. Motivation for organizations could be improving their abilities to respond immediately to the needs and changes in the market.

The Second challenge an organization can have in adopting agile for EA is the need for integration which means combining traditional EA and agile methods because each approach alone cannot address all EA management challenges. The difficult part is to find a way for these two approaches to work together smoothly and balance and align with the organization's goals. The motivation for addressing this challenge is that it allows organizations to increase flexibility next to having a long-term vision.

The third challenge is customization for each organization. Having a one-size-fits-all approach does not work in managing EA. Each organization has different and unique needs, goals, and contexts. It can be hard to find the right balance of flexibility and stability that can fit an organization's needs. However, the research by Akinpelu et al. [18] outlines a range of agile architecture frameworks that can be chosen according to the aims, goals, and needs of a company. This provides organi-

zations the opportunity to choose the framework that best aligns with their unique circumstances.

Referencing the work of Akinpelu et al. [18], There are some challenges in agile EA which are:

1. Communication: it is an important challenge, especially in Distributed Agile Development (DAD) projects where teams are not co-located. These projects can take about 2.5 times longer to communicate effectively compared to teams that work in the same location, making coordination and collaboration harder.

2. Interaction between architects and developers: it can be challenging because developers sometimes doubt architects and concentrate more on business needs than on following good architectural practices. This may cause conflicts and make it hard to implement agile architecture effectively. To solve this issue, they need to have good communication and experience. This point is also mentioned in the papers of Jourkovski et al and Duijs et al [19]. In the interviewees according to this study, both developers and architects mentioned the Scaled Agile Framework (SAFe) (which is "an organizational pattern for implementing agile approaches at an enterprise scale" [20]) positively. Some claimed that the SAFe was one way to bring developers and architects closer together, by providing developers with some architectural, more structured, ways of working. They do not want to apply the whole framework at once, but parts that are suitable for a specific team. [21]. In the research by Kleenhaus et al.[22], there are also communication challenges between enterprise architects (EAs) and agile teams (ATs). They often face some issues especially when it depends on mediators. While EAs are involved from the beginning of projects, they usually become less engaged while the project is going on so it causes communication gaps.

3. Balancing design and flexibility: There's a risk and chance of project failure if spending too little time on upfront architectural design and too much time on design can delay delivering value to customers and it can be harder to respond quickly to changes. Organizations should find the right balance between upfront design and flexibility.

4. Complexity and lack of Understanding: Large organizations often struggle to understand their complex structures. Sometimes only a few employees know about important information or only specific departments know it. It makes it hard for the entire organization to respond effectively to changes. Lack of transparency can make agile methods less effective.

The first two challenges are also mentioned by Askarianejad [23]. The writer also mentioned the reason that large companies struggle to adopt agile is because agile does not give a complete and long-term perspective of the whole system.

Based on the research of Buckl et al. [24], there are also some challenges in enterprise architecture management that agile can help to address. Enterprise architec-

ture management is a structured approach to improving an organization by aligning its business goals with IT systems. The challenges in this field are communication gaps, delivering results slowly, low stakeholder commitment, and difficulty adapting to changes. Agile practices can increase collaboration and shared understanding among stakeholders. Agile values create a supportive environment which can help to increase stakeholder involvement. As agile methods are flexible, they let enterprise architecture management quickly adapt to new technologies, market changes, and evolving business needs, and make EA more responsive. So integrating agile methods can help EA management to be more efficient.

Based on the research of Hustad et al. [25], they searched how knowledge is shared in large-scale projects. They used two main approaches which are documentation (explicit knowledge) and personal interactions (tactic knowledge). Different system development methods such as traditional, agile, or hybrid, affect how knowledge is shared. Traditional methods focus on formal documentation more and agile methods focus on personal interaction more. They found that the hybrid approach which is a combination of agile and stage-gate (waterfall) methods can be the best solution for large projects. However, even in agile projects, some level of documentation is needed but instead of creating detailed documents, easy-to-produce documents are recommended. Finding the right balance between penalization and formal documentation is challenging.

According to the study by Mederios et al [26], there is not enough research about how companies adopted agile EA and deal with the different challenges that it brings. The finding of this paper is that different teams choose different types of agile methods based on their needs and stakeholders embedding resources across teams to increase communications.

Table 2.3, shows the summary of different challenges of adopting agile for EA:

Category	Description	References
Long-term vs Short-term	Balancing stable long-term EA with the flexibility of agile to respond to business and technology changes	Hauder et al. (2014), Wessel et al. (2021)
Integration of Approaches	Combining traditional EA and agile for better flexibility and alignment with business goals	Akinpelu et al. (2021), Hauder et al. (2014)
Customization	Adapting agile EA to specific organizational needs and contexts	Akinpelu et al. (2021)
Communication	Challenges in Distributed Agile Development (DAD), lack of transparency, and conflicts in teams	Akinpelu et al. (2021), Uludag et al. (2022), Jourkovski et al. (2018)
Balancing Design & Flexibility	Finding the right balance between upfront design and adaptability	Buckl et al. (2011), Hustad et al. (2021)
Complexity & Lack of Understanding	Difficulty in large organizations managing complex structures due to lack of transparency	Askarinejad (2012), Kleenhaus et al. (2019)
Documentation vs Personal Interaction	Finding the right balance between formal documentation and personal interaction in large-scale projects.	Hustad et al. (2021)

Table 2.3: Summary of Challenges of Adopting Agile for EA

2.2.2 Required changes for adopting agile EA

According to the study conducted by Wessel et al. [17], some changes are needed for adopting agile methods in EA: 1. Redefining EA roles and processes: before having a large-scale agile transformation, EA roles need to be redefined to be aligned with the agile methods. It includes making EA more integrated with agile activities. 2. Introducing hybrid models: combining elements from different agile frameworks can help gaps that a single framework cannot handle. 3. Maintaining some waterfall features: having some structured elements like roadmaps and guidelines which the waterfall has, is necessary for continuing to reach the goals. 4. Governance mechanisms: setting up governance that has both agile and EA roles, like cross-team planning sessions and sharing the architecture vision, helps keep agile teams aligned with the organization's overall goals. 5. Management support and training: support and training are necessary to be sure that agile teams understand EA's vision and the alignment of it with agile practices.

Referencing the work of Guo et al. [27], alignment is commonly agreed as the most

important benefit EA brings to organizations to improve agility. However, few studies described what alignment includes and how to achieve a cost-efficient alignment without compromising the necessary quality. Also, it was mentioned in the study that some recent trends indicate that EA must be applied in an existing agile environment (small or large scale or mixed) and co-work with architectural styles like Service-Oriented Architecture (SOA) and microservice (it is a way of software structure which uses small and independent services [28]). SOA is a method of software development that uses services for building business applications. Each service delivers a separate business capability. Services communicate with each other over different platforms and languages [29]. In this paper, the focus is on organizational changes for agile adoption. The changes are team structures, governance, processes, culture, and leadership. Using these changes for the agile enterprise.

According to the study by Guo et al [30], the suggested use of EA in an agile and "business outcome-driven" way means that EA should not mainly be developed and used according to a pre-defined framework. Instead, EA should be developed and used for specific business purposes and using concrete deliverables so a more effective and efficient way of EA application can be enabled expected to be achieved by using EA (The What) and how to achieve these goals through EA solutions (The How). To address these issues, we analyzed the information provided by leading EA tool vendors available on their websites to get inspiration. The results showed that Use Cases (UCs) are used generally to motivate potential EA users by focusing on specific business issues. Then, EA solutions to address such business requirements or challenges are scoped and derived accordingly. We expect relevant findings could bring inspiration to agile EA engineering, change the EA's “heavy-weight” reputation, and improve the application of EA even among its sceptic.

According to Cammin et al [31], there are some requirements for agile enterprise architecture management. Some of these requirements are useful for understanding the required changes for adopting agile EA which are: continuous improvement that says early architecture work for ensuring quick improvements. Monitoring internal and external changes is another important requirement as it can help for quick reactions. using prototypes to help test and validate the architecture before full implementation.

Table 2.4 shows the summary of required changes for adopting agile EA:

Category	Description	References
Redefining Roles	Adjusting EA roles to integrate agile methods	Wessel et al. (2021), Uludag et al. (2022)
Hybrid Models	Combining agile frameworks with traditional waterfall models to address gaps	Wessel et al. (2021), Guo et al. (2021)
Governance Mechanisms	Introducing agile governance mechanisms to ensure alignment with organizational goals	Wessel et al. (2021), Alzoubi et al. (2023)
Training for Agile EA	Training teams on how to implement and understand agile EA methods	Alzoubi et al. (2023), Guo et al. (2021)

TABLE 2.4: Summary of Required Changes for Adopting Agile EA

2.2.3 Meeting agile team collaboration expectations

According to the study conducted by Reiter et al. [32], EAs often do not add value at the team level, leading to resistance to collaboration. Agile teams may find EA contributions irrelevant to their immediate needs, causing a disconnect and reduced engagement. One of the reasons is that EAs do have not enough technical knowledge which can make their work relevant to agile teams. so the architecture they provide isn't always useful. Another reason is having no regular feedback can cause limited collaboration and improvement. The last reason is indirect communication. If third parties do the communication, there is always a chance that information can be lost, dissatisfaction among agile teams, and reluctance to communicate directly with EAs. There are some tactics for improving collaboration between EAs and ATs in large-scale agile development. 1. Automated quality checks: To ensure agile teams follow architecture principles, automated checks are added to the development process. These automated checks arrange real-time feedback and check the standards as well. 2. Empowered communities of practices for architecture: EAs and agile teams are involved with each other and make architecture decisions together. Doing this tactic lets agile teams be involved in making decisions. 3. Agile architecture decision-making mode: Small decisions can be made quickly by teams but more important or broader decisions should be reviewed by the right experts to make everything run smoothly. 4. Conducting Architecture Spikes to assist agile teams: Architecture Spikes help agile teams tackle new or unknown technologies. EAs often show these spikes to provide guidance and technical support, ensuring teams have an architectural framework to work from, reducing risk, and improving understanding.

ATs have some expectations from EAs in large-scale agile development. According to Kleehaus et al. [22], modeling is one of their expectations ATs depend on different models provided by EAs, such as application landscape diagrams, business capability maps, and data models so ATs expect to have these models available,

high-quality, relevant, and detailed.

Availability is another expectation; ATs want EAs to help with consultation when it is needed. There are some challenges in some organizations with EA availability because of limited capacity.

Another expectation is communication; most of the communication between these two teams is indirect through solution architectures (SAs) or domain architectures (DAs). ATs prefer direct and frequent communication with EAs. Involvement is also important for ATs because ATs sometimes are not included in the early planning of architecture and they can make changes later but ATs would like to be involved from the beginning.

Although EAs support ATs through architecture principles, technical guidance, and consulting on architectural issues, ATs need more practical help from EAs such as helping with tool selection and architecture implementation. According to Reiter et al. [32], EAs said they don't have enough time to support and work closely with agile teams fully. ATs want to give and get feedback with EAs. The feedback was informal like meetings or chats but ATs want EAs to listen to their feedback and make necessary changes. Some teams would like to have regular feedback sessions.

According to the study by Uludag et al [33], they focused on establishing architecture principles and guidelines in the context of enterprise architecture management (EAM), focusing on collaboration between EAs and ATs. The suggested approach highlights community-driven decision-making, which integrates top-down governance with effect from teams through normative (rules-based) and mimetic (copying best practices) pressures. In this approach, the emphasis is on being more flexible, trust-based governance, and empowering ATs to contribute to decision-making. The community involves EAs to achieve global, company-wide optimum and ATs with a bottom-up perspective.

Here are the steps from the collaborative approach to establishing principles and guidelines:

- **Derive drivers:**
  - EAs analyze the organization's wider goals to identify architecture drivers.
  - ATs contribute real-world insights from their day-to-day work, such as challenges faced during implementation and innovations.
- **Determine principles and guidelines:**
  - EAs draft principles based on the drivers identified, selecting which are relevant.
  - ATs contribute their perspective, assuring that principles take into account practical implementation issues.

- **Specify and classify:** Principles are set to ensure they are clear. They must be useful for the target audience. If a principle is not clear, it should be broken down into specific guidelines and Key Performance Indicators (KPIs) to measure satisfaction.
- **Vote and accept:** The community of EAs and ATs checks the principles and guidelines. Both groups have equal rights to talk about their opinions, except when guidelines are driven by legal requirements. This can encourage collaboration, and proposals must be presented convincingly by both EAs and ATs.
- **Apply principles and guidelines:** When the principles and guidelines are accepted then ATs are responsible for implementing the principles and guidelines. EAs assist ATs during implementation, and ATs provide feedback on how well the guidelines work in practice.
- **Manage compliance:** ATs have flexibility in adhering to the guidelines but if they choose to ignore a guideline, they should say the reasons. A web application helps track compliance, providing visibility into which guidelines are applied to which teams.
- **Handle changes:** EAs and ATs regularly collect feedback on the principles and guidelines and based on this feedback, the community can make changes to existing guidelines or propose entirely new ones if it is necessary. This feedback loop ensures that the guidelines remain relevant and adaptable to changing needs.

The steps of the collaborative approach are shown in figure 2.3.

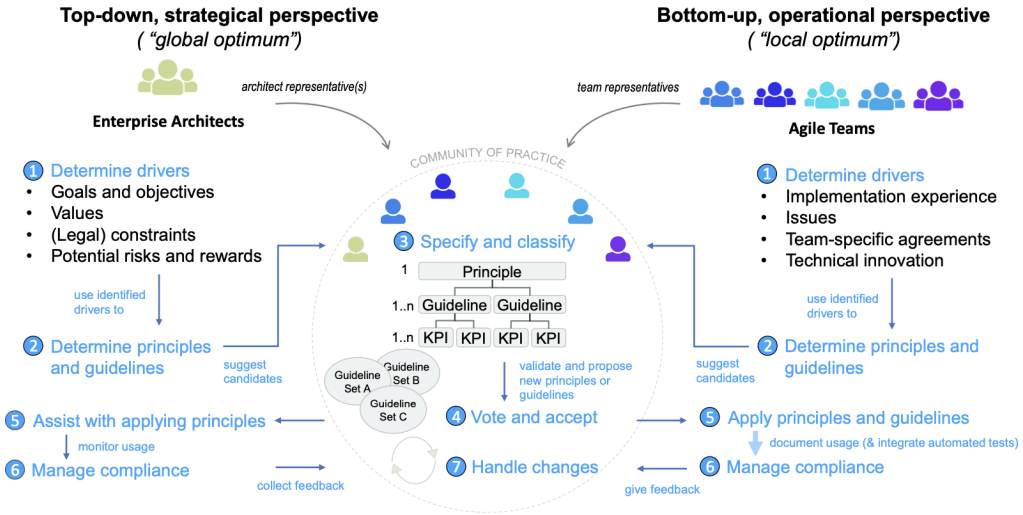

FIGURE 2.3: Overview of the collaborative approach to establishing principles and guidelines [33]

In the "apply principles and guidelines" step, to encourage teams to follow the principles, a "belt system" has been introduced. Teams start with a white belt and can

work to reach the black belt by following the principles and guidelines. This system applies normative pressures which means that teams are encouraged to follow the rules to receive rewards, such as belt promotions. The belt ranking also became public which created mimetic pressures. Teams with higher belts are like role models so other teams can learn from them.

The collaborative approach has some benefits. It encourages collaboration between EAs and ATs, which can ensure that architecture principles are a combination of top-bottom and bottom-up approaches. Community-driven, which lets both architects and agile teams take part in decisions, helping everyone feel more involved and supported by the voting process. Transparency and influence, which the belt ranking system helped to add visibility and inspire teams to follow best practices.

2.2.4 Methods and frameworks used in agile enterprise architecture

In the study [34], a practical agile EA modeling method has been introduced to improve the way organizations model their EA. The method was explained in detail and showed the way it aligns with various agile principles. Key factors for successfully implementing this agile method were: getting support from the organization's top management, setting clear roles for projects, guiding projects through business questions, and ensuring strong communication with stakeholders. The agile EA method can be seen in figure 2.4.

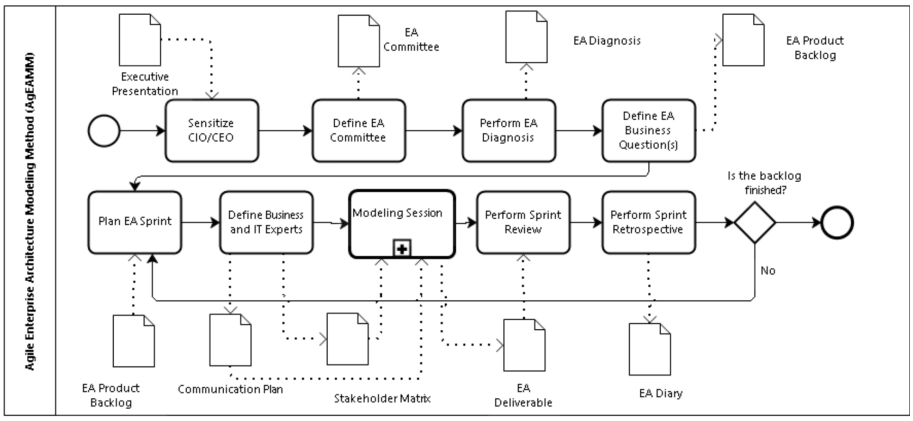

FIGURE 2.4: Agile EA modeling method designed in study by [34]

In the work of Hanschke et al. [10], agile enterprise architecture is achieved by integrating frameworks like Scrum and The Open Group Architecture Framework (TOGAF) to try to combine the flexibility of agile methods with the structured approach of enterprise architecture management.

TOGAF is one of the enterprise architecture frameworks that helps to improve business efficiency and helps businesses define their goals and align them with architecture objectives around enterprise software development [35]. The Architecture Development Method (ADM) is the core of TOGAF. It is a step-by-step approach

to developing an EA on different levels of detail. EA has three different levels which are:

- Enterprise strategic level: Focuses on overall business strategy and long-term goals.
- Segment architecture level: Concerns specific segments or departments of the enterprise.
- Capability architecture level: Involves the implementation of specific capabilities and technical solutions.

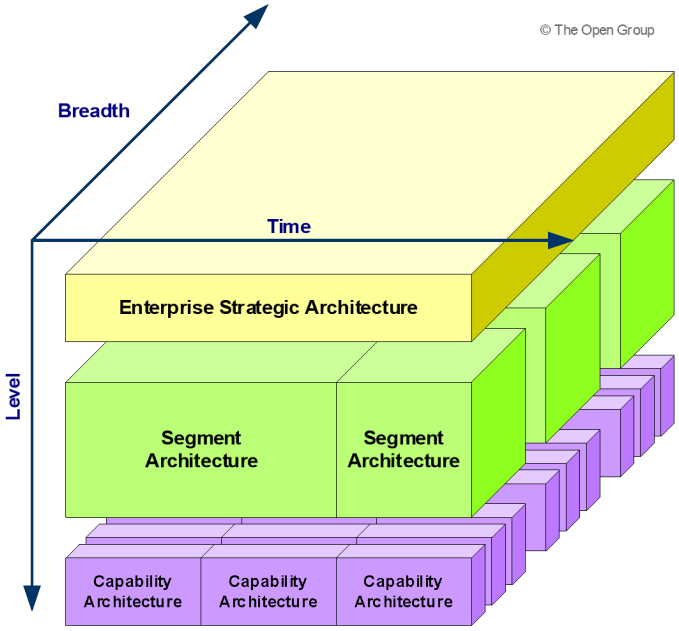

FIGURE 2.5: Classification Model for Architecture Landscapes [36]

The ADM has ten phases: eight main phases labeled A to H, and two additional phases which are the Preliminary phase; which is done once, and the Requirements Management phase, which is ongoing [10]. The different phases of the TOGAF framework are:

- **Preliminary Phase**: Preparation and initiation activities necessary to set up the architecture capability, including customization of TOGAF and definition of architecture principles, usually only carried out once.
- **A. Architecture Vision**: Initial phase of an architecture development cycle: definition of scope, identification of stakeholders, creation of the architecture vision, and obtaining stakeholder approval to proceed.
- **B. Business Architecture**: Development of business architecture to support the agreed architecture vision.

- **C. Information Systems Architecture**: Development of information systems architecture to support the agreed architecture vision (i.e., data and application architecture).
- **D. Technology Architecture**: Development of technology architecture to support the agreed architecture vision.
- **E: Opportunities & Solutions**: Initial implementation planning and identification of ways to deliver the architecture described in the last phases.
- **F. Migration Planning**: Detailed implementation and migration plan to move from baseline to target architectures (transition architectures).
- **G. Implementation Governance**: Architectural oversight of the implementation and governance function.
- **H. Architecture Change Management**: Procedures for managing changes to the architecture.
- **Requirements Management**: Continuous process of managing architecture requirements throughout the ADM, influencing activities and results in each of the eight main phases.

Agile Software Development (ASD) has become used in recent years. Its iterative and incremental approach [37]. It allows for quick feedback loops, enhancing collaboration with stakeholders, and making it easier to address changing requirements [38]. Scrum is a popular ASD method that can provide a clear framework. In Scrum, work is organized into sprints lasting from one to four weeks. At the end of each Sprint, a product increment is delivered. Scrum uses feedback and retrospectives to improve continually [10].

Integration of TOGAF and Scrum addresses the challenges of making architecture development more dynamic and adaptable while still aligning with long-term business goals. Scrum and TOGAF are widely used and accepted in real-world environments so they chose these two frameworks because of their popularity and practical relevance. EA frameworks like TOGFAF and Zachmen aim to make the complex task of enterprise management (EAM) easier [39]. These frameworks help design and set up EAM processes, making it easier to create architecture models efficiently and in the appropriate level of detail [40].
Scrum can be applied to the creation and management of architectures. For example, the Architecture Development Team (ADT) uses Scrum to break down the development of the EA into manageable tasks or the ADT works in sprints, so they focus on smaller parts of the architecture such as the Business, Information Systems, and Technology Architecture layers, rather than trying to design the whole EA in one go. This lets the architecture be built iteratively and flexibly, adapting to changing business needs, instead of a long-term process.

As mentioned, there are three different levels of architecture, In this study [10], the Enterprise Strategic Level and Segment Architecture Level are grouped into the

same category because both of them focus on higher-level, long-term architectural planning and strategy. In contrast, the Capability Architecture Level is put in a separate category because it addresses the practical implementation of specific technical solutions and architecture changes.
On the Enterprise Strategic or Segment Architecture Level a framework architecture is created and its refinement and implementation in vertical cuts through all architecture layers, as proposed by Scrum, are enabled. The ADT is responsible for developing the EA. It takes care of phases A to D and the Requirements Management phase of the TOGAF ADM. On the Capability Architecture Level, implementation teams use Scrum to make architecture changes. The teams can make their own decisions but they should follow the overall EA to be sure everything fits together. Pipelining is also important in this case which means, the ADT and the Portfolio Management Team (PMT) work one step ahead of the implementation teams, to be sure that upcoming projects are organized in advance. If something in the implementation is delayed or doesn't work as expected, the process can be slow, but communication between teams helps keep everything moving. The overview of integration is shown in figure 2.6.

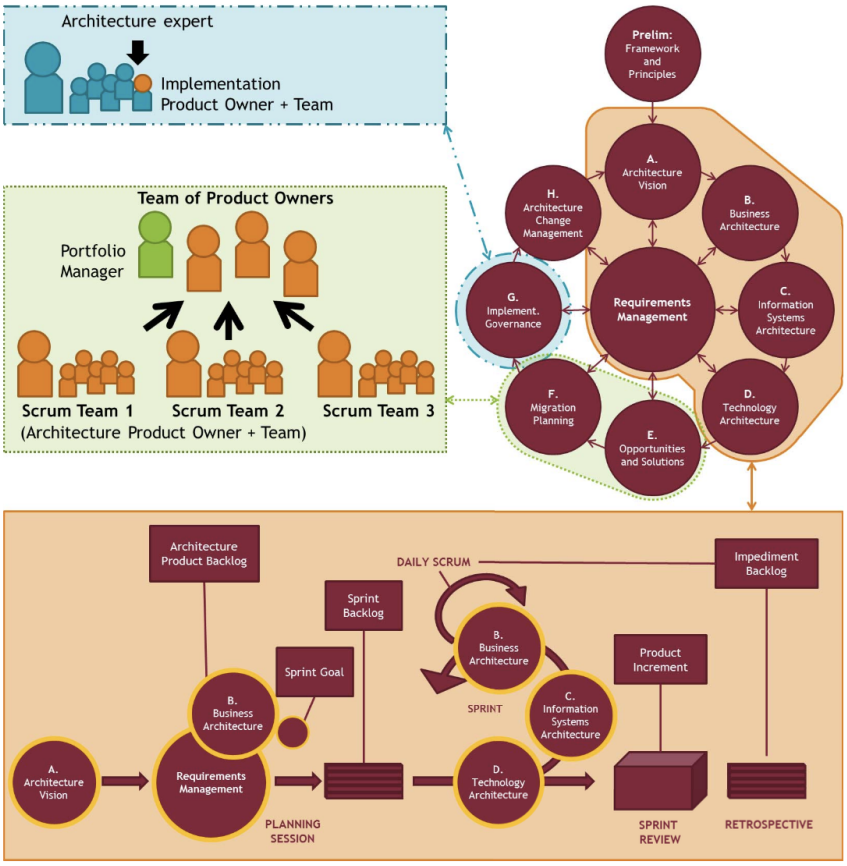

FIGURE 2.6: Overview of the integration. [10]

According to the study conducted by Wessel et al. [17], organizations face challenges when they apply agile methodologies across large-scale operations. Agile methods work very well for managing individual projects but they face significant challenges when applied across an entire company. This study analyzed three large service companies that implemented different Agile-Scaling Frameworks (ASFs) including SAFe (Scaled Agile Framework), LeSS (Large Scale Scrum), and the Spotify model. There are short explanations of these ASFs in table 2.5.

Table 2.5: Definition of agile-scaling frameworks

Agile-Scaling Frameworks	**Explanation**
LeSS	The LeSS framework seeks to apply the principles and ideals of scrum in a large-scale enterprise context as simply as possible through defined rules and guides [41].
Spotify model	The Spotify model is a people-driven, autonomous approach for scaling agile that emphasizes the importance of culture and network. It has helped organizations increase innovation and productivity by focusing on autonomy, communication, accountability, and quality. Spotify is not a framework [42].
SAFe	The framework includes structured guidance on roles and responsibilities, how to plan and manage the work, and values to uphold. SAFe promotes alignment, collaboration, and delivery across large numbers of agile teams [20].

These frameworks could not provide an ideal solution for managing agile processes alone. None of these frameworks alone could offer a complete solution for managing agile processes while maintaining the structured oversight required by EA. These companies faced some issues aligning their agile processes with the company's architectural goals, so they adopted hybrid approaches, combining elements from multiple frameworks to manage dependencies and improve team coordination but these hybrid models also did not provide acceptable mechanisms for managing EA effectively. Companies wanted to get rid of their traditional waterfall-oriented approach. However, all companies experienced that some waterfall-related characteristics of EA stayed important in an agile context so business managers and enterprise architects work together to create a clear vision and roadmap for the company's architecture also EAs create standards and long-term plans for how the system should be built. While ASFs are useful for helping different teams work together and manage tasks, they don't supply governance for keeping the entire organization's architecture consistent. To address this issue, the study suggested a conceptual model for future research on governance for keeping the entire organization's architecture consistent. This model combines the flexibility of Agile methods with the structured oversight

that EA requires. A proposed conceptual model is shown in 2.7
Here are the explanations of different parts of this model:

- ASF Characteristics: They are the specific elements and processes of the Agile-Scaling Framework being used and how they contribute to addressing agility at scale.
- Scope of Application: This defines the extent to which Agile methods are applied across the entire organization, covering both core business functions and supporting operations.
- Attention to EA: This emphasizes the need for consistent focus on Enterprise Architecture during the Agile transformation, ensuring that long-term goals and architectural coherence are maintained alongside agile flexibility.
- Process and Service Innovation: This construct evaluates the impact that Agile governance has on innovation within the organization, particularly in terms of process improvements and the development of new services.

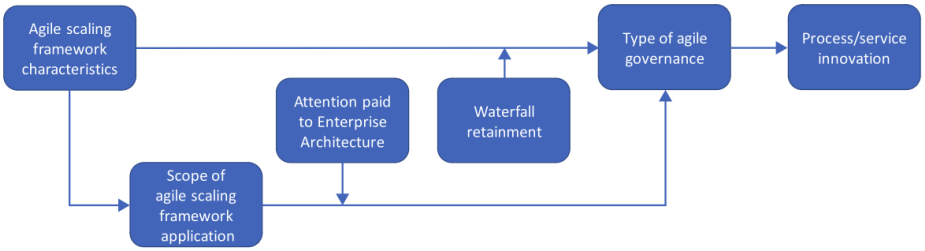

FIGURE 2.7: A framework for future research: A proposed conceptual model by [17]

The suggested conceptual model highlights that governance mechanisms are important that can balance the flexibility of agile next to structured EA. It suggests a governance structure that provides a clear long-term plan while allowing for the iterative and flexible development approaches by agile at the team level so organizations can be sure that while individual teams work with agility, the overall systems development of the company stays consistent and aligned with the company's strategic goals. For the future search, this model can be tested.

According to the study conducted by Kornyshova et al. [43], the Situational Enterprise Architecture (SEA) approach provides a flexible, scalable, and lightweight way to implement EA by allowing organizations to select and apply only the relevant EA components. SEA is based on Situational Method Engineering (SME) principles and separates large EA frameworks like TOGAF into smaller, reusable components. These components can be applied to the organization's context and needs. Three use cases are tailored to EA, which is to create a customized EA method for specific organizational needs. Progressive integration means, beginning with important

components and slowly adding more. Enhancing existing EA means filling the gaps in an existing EA framework by adding missing components.

The SEA approach can answer several EA challenges:

- Customize the EA method family into an adapted method. In this case, a particular EA approach is elaborated according to the needs of a given company.
- Provide progressive integration of the EA method components. This is a more lightweight usage of the EA approach, as only the main EA components are identified, and their implementation is done one by one.
- Select one or a set of adapted components to enhance an existing EA. In this case, an EA framework is already implemented but it does not work well the goal is to complete the lacking EA components.

In the study of Uludag et al. [44], They mentioned that the research in Large-scale Agile Development (LSAD) is empirical and based on real-world observations, not theoretical. The studies mostly talk about the lessons learned through case studies and there are not enough contributions to develop solutions or frameworks.

In the study of Guo et al. [30], they mention as the traditional EA follows strict rules and frameworks, EA should be more agile and focus on business goals. EA should solve specific business problems and deliver clear useful results for the company. There are some problems in this case: 1. What types of business problems EA should solve? 2. How to solve those problems using EA. This study looked at how popular EA tools are used by different companies. They found that they often use Use Cases (UCs) to address business issues/requirements and to derive EA solutions. UCs are a simple, straightforward, and powerful way to express the functional requirements/behavior of a system. They observe how 3 UCs are leveraged by 6 leading EA tools and as a result, it is possible to use UC as an approach to defining business expectations/issues on the one hand and to derive EA solutions on the other hand.

2.3 Systematic literature review conclusion

The first two insights from this research are finding motivation for the integration of agile and EA and also recognizing the challenges and trying to address them. Some of these challenges are communication, balancing design and flexibility, complexity, and lack of understanding. Also in this integration, some changes are required to adopt agile EA such as redefining roles and training for agile EA.

Another insight from this research is checking the interaction between different roles especially agile teams with EAs. The study shows that if they have effective collaboration with fixed rules and regular feedback, then they can build an architecture that is flexible and follows a consistent plan.

Moreover, integrating some elements of agile frameworks like TOGAF and Scrum mentioned in paper [10] offers an approach to balancing the flexibility of agile methods with the structured discipline of enterprise architecture. TOGAF's phased, strategic planning combined with Scrum's iterative, adaptive processes allows for architecture to develop gradually while staying aligned with long-term goals. This approach keeps the projects flexible and organized and makes them a good fit for fast changes. Another study mentions that sometimes only one single method does not work and it is better to use different Agile-Scaling Frameworks (hybrid approach). None of these frameworks alone could offer a complete solution for managing agile processes while maintaining the structured oversight required by EA. Therefore, elements from multiple frameworks are integrated to manage dependencies and improve team coordination.

The findings from the literature review can make the structure and steps of the new method to help technology-native companies apply the agile principles and decrease their challenges. To summarize the findings, the following design requirements from the literature review side can be customized for a modern EA method tailored to technology-native companies:

1. Maintain some waterfall features next to agile principles
2. Automated quality check
3. Real-time monitoring
4. Have a modeling support
5. Better communication
6. Adaptive governance for agile and EA integration

2.4 Further background information

2.4.1 Cloud-native application & software architecture

Cloud-native applications are software programs that consist of multiple small, interdependent services called microservices. Traditional software programs have monolithic applications with a single block structure including all the required functionalities. By using the cloud-native approach, software developers break the functionalities into smaller microservices. This makes applications more agile as these microservices work independently and take minimal computing resources to run [45].

There are different ways to build software which is known as software architecture. There are different types of software architectures such as layered, microservices, and monolithic [46]. This section focuses on monolithic and microservice architectures and their differences.

In monolithic architectures, all processes are coupled and run as a single service so if one part needs more resources, the entire architecture must be scaled. Adding or improving a monolithic application's features is becoming more complex because of the large and complex code. This limits experimentation and also implementation of new ideas. Monolithic architectures add risk for application availability because many dependent and tightly coupled processes increase the impact of a single process failure [28]. On the other hand, a microservices architecture builds an application as independent components that run each application process as a service. These services communicate via an interface using lightweight APIs. Services are built for business capabilities and each service executes a single function. As they run independently, each service can be updated, deployed, and scaled to address the demand for specific functions of an application [28].

2.4.2 Automated governance

Automated governance suggests using technology to implement rules and policies in IT operations automatically. Automated governance helps companies with a consistent approach to risk management, compliance, and security. It decreases the need for manual intervention, enhances scalability, and enables best practices within managed environments. Automated governance promotes a balance between agility and control by supplying assurance and responsibility while enabling innovation and rapid deployment [47].

2.4.3 DevOps

"DevOps is the combination of cultural philosophies, practices, and tools that improves an organization's ability to deliver applications and services at high velocity: improving products faster than organizations using traditional software development and infrastructure management processes. This speed allows organizations to enhance customer satisfaction and compete more effectively in the market" [48].

Some practices of DevOps are: continuous integration, continuous delivery, and microservices.

- Continuous integration: Developers combine their code into one place. Afterward, the system automatically checks for mistakes and tests the code. This can help to find problems early enough, improve the quality of the software, and release updates faster [48].

- Continuous delivery: changing codes are automatically built, tested, and prepared for a production release faster. It helps to keep continuing integration by deploying code changes to the testing or production environment after the building. With this, developers can have a ready-to-deploy version of software [48].

4 Treatment design

This chapter explains the designed method in detail. Based on the requirements from the interviews and insights from the literature reviews, the method is designed. First, the high-level method is explained and further, the activities within each step are also explained.

4.1 Design process

In chapters 2 and 3, the design requirements have been mentioned based on the interviews and literature review. Table 4.1 shows which part of the design addresses these requirements. A detailed explanation of the entire design and its steps is provided in the following sections.

TABLE 4.1: A table showing which part of the design addresses which requirement, and on which problem investigation it was based on.

No.	Design Requirement	Based on	Design Choices
1	Shift to federated governance with flexible guardrails	Interviews & literature review	- Whole method
2	Automate compliance and governance processes to reduce manual work	Interviews & literature review	- Automated governance and compliance
3	Use tools like capability maps and value streams for better communication and alignment	Interviews & literature review	- Capability enhancement assessment
4	Manage microservices effectively and reuse existing services	Interviews & literature review	- Check microservices availability - Reuse existing microservices
5	Integrate EA tools with DevOps and CI/CD pipelines	Interviews	- Integrate into CI/CD pipeline for new microservices
6	Enhance visibility into systems and software components by using SBOM	Interviews	- SBOM availability check

7	Focus on product-centric approaches rather than application-centric ones	Interviews	- Updated business goals
8	Optimize cloud cost and resource management	Interviews & literature review	- Evaluate the total cost of ownership
9	Automate integration and monitoring processes for faster decision-making	Interviews & literature review	- Real-time monitoring - Integration check
10	Support collaboration through teams	Interviews & literature reviews	- Shown by stakeholders throughout the entire low-level method

Based on the interviews and literature review communication in technology-native companies is important. It is not a separate step within this designed method. However, in the low-level method which will be explained in section 4.4, the stakeholders that are responsible for each activity have been shown and their roles have been explained.

4.2 Explanation of the method

Using the design requirements in the previous chapters based on the interviews and the insights from the literature review, a method has been designed to align EA practices with agile methodologies to benefit technology-native companies. It includes a high-level method that shows the main steps. Each step has been explained by specific activities in a low-level method. The developed method has been inspired by TOGAF ADM. It is a simpler version of it with fewer steps. The content of these steps is more specific to use for companies that are more agile and do self-development.

This method includes more than just agile principles. While agile methods emphasize speed, iterations, and action, they can cause some risks such as security vulnerabilities. For example, the company may quickly decide to buy an application but later the company figures out the application is not exactly what they needed and does not cover all their requirements. To decrease these kinds of risks, it is important to have some waterfall characteristics to do some check marks and go through some specific steps to be sure that the implementation is something right.

4.3 High level method

As already mentioned this high-level method is inspired by TOGAF ADM. It has 6 steps which are foundation setup, defining goals, current situation assessment, decision-making, design & planning, and monitoring. The high-level method can be seen in figure 4.1.

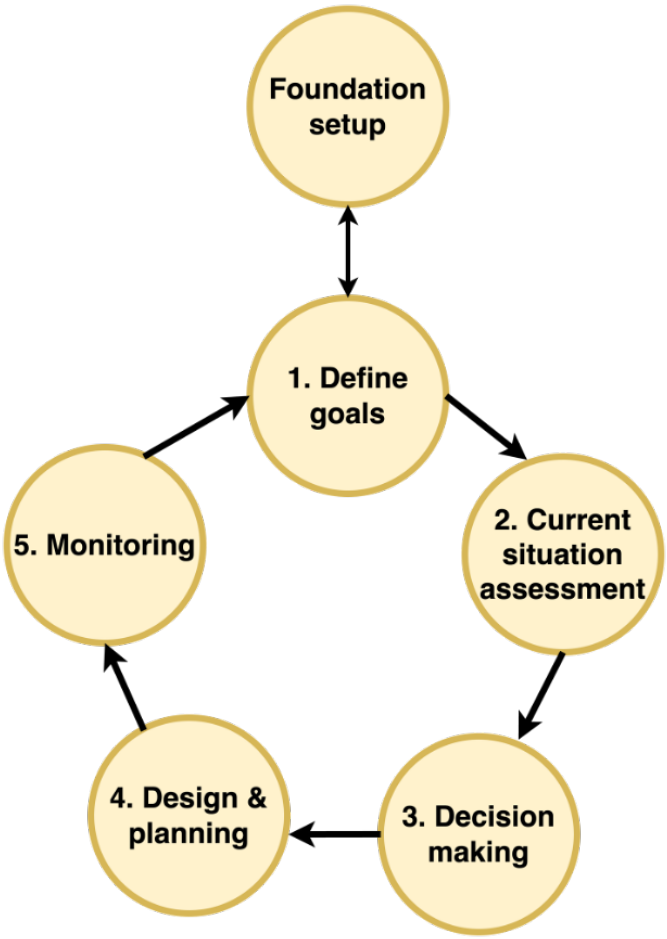

FIGURE 4.1: The designed agile enterprise architecture high-level method for technology-native organizations

Table 4.2 shows the mapping of TOGAF ADM steps to the designed high-level method. The TOGAF ADM steps are explained in detail in Chapter 2. In the following section, a brief explanation of each step in the designed high-level agile EA method can be found.

Table 4.2: Mapping the method with TOGAF ADM

TOGAF	**Designed method**
Preliminary	Foundation setup
Architecture vision	Define goals
Business, Information systems & Technology architecture	Current situation
Opportunities & solutions	Decision making
Migration planning	Design & planning
Implementation governance	Monitoring

4.3.1 Foundation setup

This step aligns with the preliminary step of TOGAF ADM. In the preliminary phase of TOGAF, we need to establish governance and compliance. In the foundation setup step, governance & compliance will be checked. As the method is for dynamic and tech-driven companies, agility is crucial. Interviewers also focused on having governance and compliance checks automatically. Automating these processes can reduce the need for manual work and enhance scalability.

4.3.2 Define goals

This step aligns with phase A (architecture vision) of TOGAF ADM. This step aims to validate business principles, goals, and drivers. In the define goals step, also the goals will be set and checked after setting the governance & compliance.

4.3.3 Current situation assessment

This step aligns with phases B, C, and D (business architecture, information systems architectures, and technology architecture) of TOGAF ADM. The activity of these steps is to develop architectures at three levels: business, information systems, and technology. At each level, both the baseline (as-is situation) and the target (to-be situation) are set to give a clear understanding of the existing and desired conditions. After the states are defined, the differences and the gaps are analyzed to show what needs to be changed. In the current situation step capability assessment, application assessment, TCO evaluation, dependency map checking, integration checking, and managing risks will be checked to determine the current situation and help to make decisions for future steps.

4.3.4 Decision making

This step aligns with phase E (opportunities & solutions) of TOGAF ADM. The objective of this step is to assess and choose the best implementation options based on the assessment. For example, building an application or buying one.

4.3.5 Design & planning

This step aligns with phase F (migration planning) of TOGAF ADM. This step is quite different depending on the decision of building or buying an application. The details of each path will be explained in the coming section.

4.3.6 Monitoring

This step aligns with phase G (implementation governance) of TOGAF ADM. This part is about real-time monitoring and managing the projects after the implementation. It helps to ensure that the implementation fits the approved design, business goals, and all requirements.

4.4 Low-level method

Each step of the high-level method includes various activities within the designed low-level method. The low-level method is illustrated in a flowchart, as shown in figures 4.2 & 4.3. As the low-level method is extensive and has lots of activities, it was separated into two figures for a clearer overview. Figure 4.3 illustrates the second part of the method and "X" is the connection point between the first part to the second part of the low-level method. Each activity has been explained in this section. For every activity, its input, process, and output have been identified. Input-process-output (IPO) is a tool used to define a workflow, the flow of information, or activities within a system. Input can have data, information, or resources that enter the system. Processes can have activities or operations executed on the inputs and output can have results, products, or outcomes that are produced by the processes [57]. Data within a flowchart shows the input and output of each activity as well as describes resources used or generated [58]. The stakeholders that are responsible for each activity are shown in the flowchart and will be explained in this section as well.

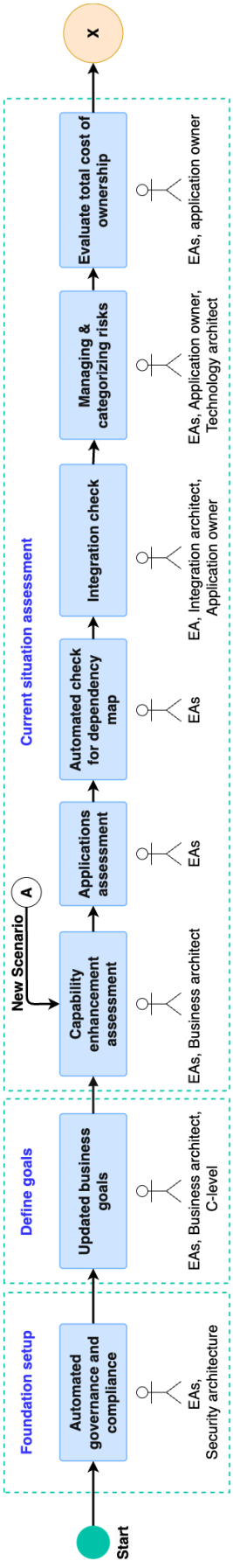

FIGURE 4.2: The first part of the designed low-level method for technology-native organizations

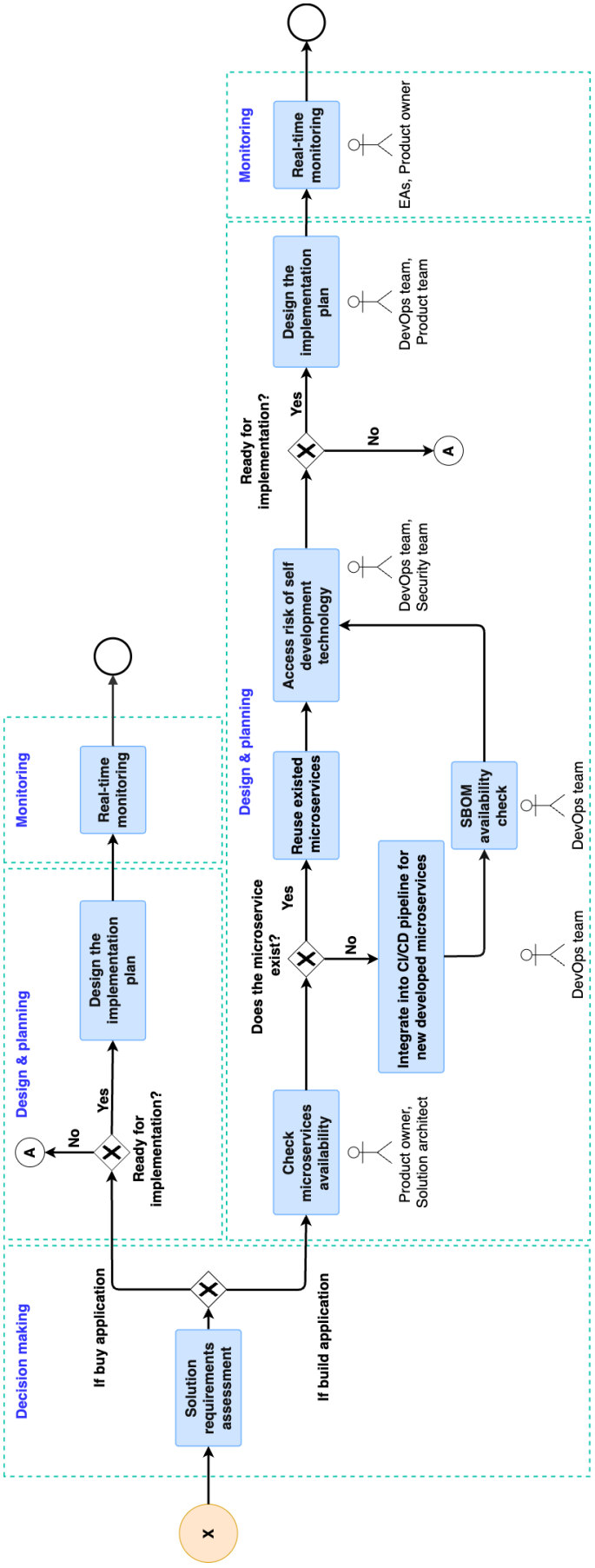

FIGURE 4.3: The second part of the designed low-level method for technology-native organizations

4.4.1 Automated governance and compliance

The process within the method begins with setting up automated governance and compliance which is the only activity within the first step of the high-level method. This means creating a system to ensure the organization follows security, regulatory, and operational rules. Some tools can be used to automate tasks in this activity. Automating governance and compliance helps the organization follow the rules, decrease risks, and stay organized without doing things manually. Automated governance can include security compliance checks and data protection to fulfill regulations and keep information safe. Stakeholders involved in this activity are EAs and security architecture.

Table 4.3: Automated governance and compliance

Step:	*Automated governance and compliance*
Input	Policies and rules that are set by internal people and have external regulation
Process	Defining policies, implementing automation, and monitoring compliance
Output	Automated compliance report/dashboard generated by a system

The input and output of this activity are in the table 4.3. The information of this input can be the policies and rules set by the security team and they try to align with compliance requirements within the organizations and external regulations can be from regulatory bodies such as GDPR ("It is a crucial component of EU privacy law that helps to manage personal data" [59]) and ISO 27001 ("an international standard for addressing information security" [60]). EAs design governance frameworks and choose automation tools. The information of the output can be compliance dashboards and reports developed by automated governance tools that will be used by security teams and EAs for tracking.

4.4.2 Updated business goals

The next step is to define and update business goals. This activity is also the only activity in defining the goal step of the high-level method. Goals should support agility within cloud-native organizations, and help the organization perform better. Stakeholders that are involved in this activity are EAs, business architects, and C-level executives.

The input, process, and output of this activity can be seen in Table 4.4. The information of this input can be the feedback of stakeholders from different teams and customers feedback and according to this feedback, EAs, C-level executives, and business architecture decide to update the business goals. The information of the output can be the updated business goal document created by business architecture and C-level.

TABLE 4.4: Updated business goals

Step:	*Updated business goals*
Input	Stakeholders feedback
Process	Analyzing business needs, aligning goals with objectives
Output	Updated business goals document create by stakeholders within the company

4.4.3 Capability enhancement assessment

This is the first activity related to the current situation assessment step. A business capability is an organization's fundamental building block, describing its needs to achieve its mission and strategic goals. Business capability focuses on what the organization does rather than how it does it [61]. During this activity, the organization assesses its existing capabilities to decide whether it can support the updated business goals. This contains identifying gaps and areas for improvement. For this, a business capability map can be useful. A business capability map is a visual expression that shows the different capabilities of an organization and the way they support the business objectives. Business capability maps help organizations analyze, prioritize, and align their capabilities with strategic goals and make better plans and decisions. One of the key values of the business capability map is gap identification. It allows organizations to find the gap in their capabilities and shows which part they need to invest in new technology, people, or processes to reach their goals [62]. The stakeholders that are involved in this activity are EAs and business architects.

TABLE 4.5: Capability enhancement assessment

Step:	*Capability enhancement assessment*
Input	Current business capability map
Process	Internal teams analyze existing capabilities using tools and use data from business goals
Output	Information about the gaps and improvement priorities based on the existed map

The input and output of this activity can be seen in Table 4.5. EAs and business architects assess the current business capabilities, analyze the situation, and give recommendations for enhancement. The output can be used by EAs to check the gaps and improve the priorities.

4.4.4 Applications assessment

In this activity, different aspects of each application will be assessed including its hosting type, to be sure that all the applications are in the cloud as technology-

native organizations are cloud-natives. The application lifecycle will also be checked to be sure that none of the applications are at the end of their life. For example, this can happen when the technology becomes obsolete or when the security risks increase. This activity helps gather more information and also helps to figure out which applications should be focused on during the following activities.

Table 4.6: Application assessment

Step:	*Application assessment*
Input	List of all the applications within the company
Process	Use proper tools to check detailed information about applications (e.g., hosting type)
Output	Detailed information about applications that can help to determine where the company should focus

EAs check the list of applications and gather more information about each application which can help make further decisions.

4.4.5 Automated check for dependency map

This is the third activity related to the current situation assessment step. Dependency mapping includes making visual models in the form of maps or diagrams that show the relationships and dependencies between different components or elements in a system or project. The purpose of it is to improve understanding by providing a visualization of how different parts of a system rely on each other. It helps stakeholders make better decisions, manage risks, and make sure that the development and operation of projects are efficient [63].

Dependency map can improve collaboration and decision-making. Some advantages of dependency mapping are:

1. Enhanced visibility and understanding; it provides a visual representation of complex relationships which helps stakeholders get a deeper understanding of a system's dependencies and the visibility enables better decision-making and solving problems [63].

2. Improved decision-making; by using a dependency map, organizations can make better decisions. It shows how different changes affect connected elements to help risk assessment and choosing proper actions [63].

3. Effective risk management; A dependency map helps organizations identify crucial points of failure and possible vulnerabilities which helps organizations manage risks and come up with strategies to mitigate risks [63].

Using a tool to help check this dependency map is useful especially for agile organizations to have a faster analysis and faster decision making. The stakeholders for this activity are EAs who can use tools to visualize and analyze these dependencies.

Table 4.7: Automated check for dependency map

Step:	*Automated check for dependency map*
Input	Information of different systems and applications that are connected and interact within the organization
Process	Use automated tools to analyze dependencies, identify relationships between applications, and visualize them in a tool to find the potential risks or other issues
Output	Dependency map that shows a clear overview of system interconnections

The input and output of this activity are in Table 4.7. The information from this input can be analyzed by EAs to understand the current situation and the output can be also used by EAs to gain more insights.

4.4.6 Integration check

This is the fourth activity related to the current situation assessment step. Integration architecture (IA) is a main part of the IT landscape of organizations. It facilitates the integration of different IT components and helps to trace data flows between applications. Application programming interfaces (APIs) are used to connect different applications. IA enables managing applications and data in IT architecture by providing and using interfaces. Integration between applications can help to smooth the processes and applications must be able to work in different processes.

IA has some advantages including reduction of costs and improved customer satisfaction. The cost of operating different systems is expensive because of their maintenance, upgrades, and coordination. IA connects systems and prepares a clear overview. As the updates affect several systems at the same time, it avoids costly errors. Doing less manual workload can help organizations to focus more on their customers [64].

Integration check can be also useful when an organization decides to replace an existing application with a new one. Before replacing, the organization must check the interactions of the application that is going to be replaced with other existing applications. It can help to identify all the dependencies and ensure that the replacement does not cause any issues and reduce risks. The stakeholders that are involved in this activity are EAs, integration architects, and application owners.

Table 4.8: Integration check

Step:	*Integration check*
Input	A report that includes information about the applications, their functionalities, and dependencies
Process	Checking and analyzing the integration between different applications. Later it can be used to check if new applications can function correctly within the existing application without causing trouble and recommendations for changes
Output	A diagram that shows how different systems interact and the potential challenges

The input and output of this activity can be found in Table 4.8. For the input, the application owners are responsible for checking how their specific applications integrate and interact with other applications in the organization. IAs supply information about applications and information about the integrations and connections among applications. EAs help to do the integration check when the organization needs to replace any application and the IAs focus on being sure that all the applications in the organization connect properly and work well together. The output helps stakeholders make smart decisions and make future improvements.

4.4.7 Managing & categorizing risks

This is the fifth activity related to the current situation assessment step. Risk is an uncertain event or situation that has a positive or negative impact on the project its objectives [65]. When organizations make changes to their EA, such as adding new business goals, moving applications to the cloud, or replacing existing applications, they can face various risks such as security issues and high costs. Managing & categorizing risks can help the organization identify potential risks and try to prevent them from happening or mitigate them. Different risks exist such as AI risk and obsolescence aggregated risks. The stakeholders that are involved in this activity are EAs, application owners, and technology architects.

Table 4.9: Managing & categorizing risks

Step:	*Managing and categorizing risks*
Input	List of risks related to different products within companies
Process	Identify and analyze the potential risks
Output	Come up with solutions based on the type of the risk.

The input and output of this activity can be seen in Table 4.9. For the input, application owners, and technology architects check the risks such as technology risks, IT risks, and data security risks. The output can be used by EAs, application owners, and technology architects to make appropriate decisions based on the risks.

4.4.8 Evaluate total cost of ownership

This is the last activity related to the current situation assessment step. It is crucial to check TCO before the company builds or buys an application. According to Hopkins et al. the cost of developing a system is less expensive than maintaining a system over time as maintenance includes fixing issues, updating, and adapting to changes. So using the total cost of ownership (TCO) can be useful for this situation as it covers the cost of building a system and the cost of maintaining and running it [15].

The stakeholders that are involved in this activity are EAs and the application owners. The application owners assess the costs that are related to application ownership and maintenance. EAs analyze long-term financial feasibility. The stakeholders use the TCO report to check costs related to license, maintenance, and support.

Table 4.10: Evaluate total cost of ownership

Step:	*Evaluate total cost of ownership*
Input	Cost data from vendor side or internal teams that check license cost, support, and maintenance costs
Process	Analysis of different costs by stakeholders
Output	TCO report preferably created by using a tool

The input and output of this activity can be seen in Table 4.10. For the input application owners collect data on license cost, support, and maintenance cost. For the output, EAs create a TCO report to have better decision-making.

4.4.9 Solution requirements assessment

This is the only activity related to the decision-making step. Based on the previous step and the information gathered, such as risk management and integration check, organizations decide whether to buy or build an application that aligns with their needs and requirements. Organizations should decide if they want to build or buy applications at this step. First, the applications that exist in the market also should be checked to see if they satisfy the requirements of the company. If yes, they can continue with buying application otherwise, they should build their own application.

The stakeholders that are involved in this activity are EAs, business analysts, and product owners. EAs ensure that the solution aligns with the organization's IT

strategy and architecture framework. Business analysts collect and check requirements and product owners prioritize requirements based on business goals.

Table 4.11: Solution requirements assessment

Step:	*Solution requirements assessment*
Input	Requirements and insights from all the previous steps (which are collected by people and systems)
Process	Analyzing information and data and check existed applications
Output	List of functional and non-functional requirements which help for solution (develop or buy)

Since the last two activities of the low-level method, which are designing the implementation plan and real-time monitoring, are the same for both building and buying an application, explaining the process for building an application will also cover the steps in buying.

4.4.10 Check microservices availability

This is the first activity related to the design & planning step if the organization decides to build an application. This step can be useful for organizations to be more agile. It is important to check the list of available microservices before starting and developing new ones. As already mentioned in the previous chapter, according to one of the interviews, it is smart to be able to reuse existing components instead of developing new ones with the same functionality. This approach can improve the efficiency.

After this step, the organization will realize that if the microservice they need already exists in the list of microservices or not. If it does, the organization can reuse the existing microservice which can help with saving time and resources and continue with the next step. If the microservice does not exist, the organization should develop a new microservice according to its requirements and integrate it into CI/CD pipelines. The stakeholders that are involved in this activity are product owners and solution architects. For the input product owners and solution architects check the applications' list and have an overview of the applications/microservices and their dependencies. The output is the decision to develop or reuse a microservice and if the organization needs to develop a new one, the DevOps team will start doing it. The input and output of this activity can be seen in Table 4.12. For the input product owners and solution architects check the applications' list and dependency map to have an overview of the applications and their dependencies. The output which is a document, helps to decide for developing new microservices or reusing existing ones.

TABLE 4.12: Check microservices availability

Step:	*Check microservices availability*
Input	List of microservices
Process	Check the list of microserives
Output	Make a decision based on the list,

4.4.11 Integrate into CI/CD pipeline for new microservices

This is the second activity related to the design & planning step if the organization decides to build an application. If in the previous step, the organization decides that a new microservice needs to be developed, the next step is to integrate it into the CI/CD pipeline. This step focuses on automating the process of testing and deploying new microservices by integrating them into the continuous integration and continuous deployment (CI/CD) pipeline. It ensures that new microservices are delivered quickly and efficiently. After the DevOps team develops a new microservice, it should go through several steps before it becomes fully operational. The CI/CD pipeline is used to automate these steps to decrease manual work which is useful for organizations to be more agile. The stakeholders involved in this activity are the DevOps team.

TABLE 4.13: Integrate into CI/CD pipeline for new microservices

Step:	*Integrate into CI/CD pipeline for new microservices*
Input	Developed application by the DevOps team
Process	Automate testing and deployment of microservices by using CI/CD pipelines
Output	Deployed microservices

The input and output of this activity can be seen in Table 4.13. For the input, the DevOps team provides code and deployment requirements and after that, they set up the integration and automation within CI/CD pipelines.

4.4.12 SBOM availability check

This is the third activity related to the design & planning step if the organization decides to build an application. "A Software Bill of Materials (SBOM) illustrates all components within a software application, including open-source libraries, third-party dependencies, licenses, and vulnerabilities. It is essential for managing dependencies, ensuring legal compliance, mitigating security risks, and supplying organizational transparency" [66]. SBOMs are essential because recently, modern enterprises rely on multiple applications including open-source or third-party components. Hackers can take advantage of weaknesses in software components. Organizations that do not track software components may face security attacks, compliance

issues, and system failures. Lots of organizations know about the importance and benefits of SBOMs but there are many challenges in adopting and implementing SBOMs as a standard practice [66]. SBOMs have a couple of benefits such as [66]:

- Enhanced cybersecurity: SBOMs provide visibility into software components and their associated vulnerabilities. By addressing these vulnerabilities, organizations can boost their cybersecurity and reduce the risk of cyberattacks.
- Streamlined software development and maintenance: SBOMs help to understand software dependencies better, and make it easier to track changes, updates, and compatibility issues. This simplifies software development, maintenance, and the integration of new features.

As already mentioned, checking SBOMs availability after integrating into the CI/CD pipeline for the new microservices can be useful for mitigating security risks, and dependency tracking. It can also help to manage and solve issues faster. The stakeholders involved in this activity are the DevOps team.

TABLE 4.14: SBOM availability check

Step:	*SBOM availability check*
Input	Software components, dependency information, open-source licenses
Process	Identify software components
Output	list of components related to each microservices

The input and output of this activity can be found in Table 4.14. The input for this step is provided by the DevOps team, which provides information about the software components, including their dependencies and open-source licenses.

4.4.13 Access risk of self-development technology

This is the fourth activity related to the design & planning step if the organization decides to build an application. This step is the step after reusing the existing microservices or SBOM availability check. In this method, risks have been managed and categorized in the previous steps from a high level. However, in this activity, the focus is on analyzing the potential risks that happen when organizations develop new microservices or applications. Organizations should consider the challenges and risks if they want to have a successful and long-term implementation. The risk related to the SBOM can be checked in this step. It is also important to pay attention to the hosting type of the new microservice. The stakeholders involved in this activity are the DevOps team and the security team.

Table 4.15: Access risk of self-development technology

Step:	*Access risk of self-development technology*
Input	List of potential risks (such as open-source or third-party component risk)
Process	Analyze risks
Output	Address risks and monitoring them

The input and output of this activity are listed in Table 4.15. The input is a list of potential risks that have been gathered by the DevOps team and security team. They try to be sure that all potential challenges are considered. the DevOps team manages technical and operational risks and the security team ensures the application is secure and compliant.

4.4.14 Design the implementation plan

This is the fifth activity related to the design & planning step whether the organization decides to build or buy an application. In both situations (building or buying an application), a decision must be made before this step, which is "Is the organization ready for implementation?" If the organization is ready, it can work on designing the implementation plan otherwise the organization should have a new scenario and go back to the previous step which is capability enhancement assessment to check the business capabilities again and come up with other plans. It is important to check when the new application will be rolled out during this step.

The stakeholders that are involved in this activity are the DevOps team and the product team that helps to roll out a new microservice or an application and do the detailed work.

Table 4.16: Design the implementation plan

Step:	*Design the implementation plan*
Input	Essential information and resources to develop a clear implementation roadmap
Process	Different phases and a structured plan for implementation
Output	Ready to use microservices

The input and output of this activity are listed in Table 4.16, the product team is responsible for gathering essential information and resources to develop a clear implementation roadmap.

4.4.15 Real-time monitoring

This is the activity related to the monitoring step whether the organization decides to build or buy an application. This is the last step within the whole activities of this method. Real-time monitoring uses tools and applications to check and record how everything is working. It can be used to improve security and find problems quickly enough. Stronger network security, noticing issues early, optimized network performance, and faster response time are some of the benefits that real-time monitoring has [67]. The stakeholders involved in this activity are the EAs and the product teams.

Table 4.17: Real-time monitoring

Step:	*Real-time monitoring*
Input	Data from different sources that provide insights
Process	Monitoring and analyzing data within dashboards and finding improvements
Output	Dashboard with various insights

The input and output of this activity can be found in Table 4.17. EAs analyze the monitoring data to find issues and suggest improvements. Product teams check the monitoring to prioritize development progress and new features. The output of this activity helps EAs and product owners monitor performance, identify potential risks, and plan necessary improvements.

6 Validation

This chapter focuses on the treatment validation and the goal of this chapter is to answer the third research question of this study which is: "To what extent does the designed EA method help technology-native organizations to work better in agile environments?"

6.1 Validation approach & method

There are different methods for validation such as expert opinion, technical action research, and statistical difference-making experiments. For this study expert opinion has been chosen. Expert opinion is one of the ways to validate an artifact which is a method in this study. In this method, experts imagine how the method would work in practice and what effects it would have. If the expected effects do not meet the requirements, the method should be redesigned. One of the positive things about this method is that negative opinions can be useful for improvements as it can identify and remove bad designs early enough [12].

The validation questions were developed based on the Unified Theory of Acceptance and Use of Technology (UTAUT) framework to provide a reliable validation process. According to Venkatesh et al. [83], UTAUT helps to indicate whether a new technology will be successful or not. UTAUT has different factors that can be used for the validation part. For this validation, the questions are based on two factors of UTAUT which are: performance expectancy and effort expectancy. In the context of this study, they mean:

1. Performance expectancy: The perceived usefulness and benefits of the designed method in helping users in their activities.
2. Effort expectancy: How easy the designed method is to use and operate by users.

Google Forms has been used to prepare the questionnaire and function as a tool for participants to answer the questions.

6.2 Validation process

The invitation to participate in the validation process of the designed method was sent among SAP LeanIX employees which resulted in a total of six participants who took part in the validation of the method and its implementation process. The validation process went as follows:

1. **Presentation**: Stakeholders first attended a presentation in which the researcher explained the topic, method, and the way it was implemented. The presentation has been explained in more detail in appendix C.

2. **Questionnaire completion**: In the end, the participants were asked to fill out a questionnaire based on their opinions on performance expectancy, effort expectancy, and add suggestions for improving the method or its implementation, based on the questions in section 6.3.

6.3 Validation questions

The validation questionnaire consists of two sections: demographic questions and questions related to the method and its implementation.
1. Demographic questions: Before the main questions about the method and its implementation, the questionnaire begins with a section on demographic questions to gather background information about participants. These are the demographic questions that have been used for the validation part to find out how much the participants are familiar with this topic. These two questions used a scale from 1 to 5. Table 6.1 shows questions:

TABLE 6.1: The demographic questions used in the validation process

No.	Question
1	How familiar are you with Enterprise Architecture (EA)?
2	How familiar are you with software development companies in terms of Enterprise Architecture?

2. Method & implementation questions: The second section includes questions related to assessing the method and its implementation. Here is the list that includes seven questions related to the method. The first five questions are Likert scale questions and the last two are open-ended questions. Likert scale is a rating scale which is used to measure opinions, attitudes, and behaviors. It includes questions that have five or seven answer choices, ranging from strong agreement to strong disagreement and participants can choose the option that is close to their opinions. As the Likert scale gives a wide range of possible answers, it is a good way to realize how strongly people agree or disagree about a topic [84]. The participants answered these questions assuming that the method would be used independently and not only within the SAP LenaIX workspace. Questions related to the method and its implementation can be found in table 6.2.

TABLE 6.2: The method & implementation questions used in the validation process

No.	Question	Options
1	How understandable and easy to follow are the steps of the proposed modern EA method?	1. Very easy 2. Easy 3. Neutral 4. Difficult 5. Very difficult
2	How well does the method incorporate agile principles for technology-native companies?	1. Extremely not well 2. Somewhat not well 3. Neutral 4. Somewhat well 5. Extremely well
3	This method can be used independently by the user without requiring specific resources (such as a person or documentation)	1. Strongly disagree 2. Disagree 3. Neutral 4. Agree 5. Strongly agree
4	How useful do you think this method would be for EA practice in technology-native companies?	1. Not useful 2. slightly useful 3.Somewhat useful 4. Very useful 5. Extremely useful
5	How useful do you think the implementation of this method would be in SAP LeanIX?	1. Not useful 2. Slightly useful 3. Somewhat useful 4. Very useful 5. Extremely useful
6	What improvements would you suggest for the method?	open-ended question
7	What improvements would you suggest for the implementation of this method?	open-ended question

6.4 Validation result

6.4.1 Demographic questions results

Q1. *How familiar are you with Enterprise Architecture (EA)?*

The results of this question can be found in Figure 6.1. Here, it can be seen that most participants (83.3%) rated their familiarity with EA at the highest level which is 5, and only a small percentage (16.7%) has moderate familiarity in this field which they rated as 4.

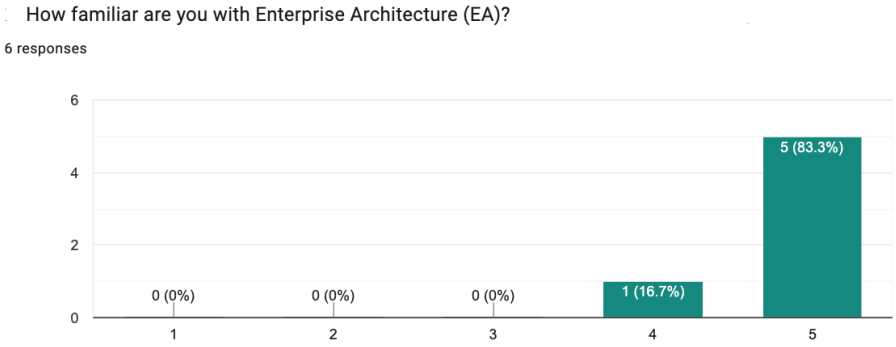

FIGURE 6.1: Familiarity with enterprise architecture

Q2. *How familiar are you with software development companies in terms of Enterprise Architecture?*

The results of this question can be found in Figure 6.2. Here it can be seen that half of the participants rated their familiarity with software development companies in terms of EA as 4 which means they are highly familiar with EA in software development companies, 16.7% of the participants rated as 5 which means they are very highly familiar, while the rest of the participants have modern or low familiarity with this subject.

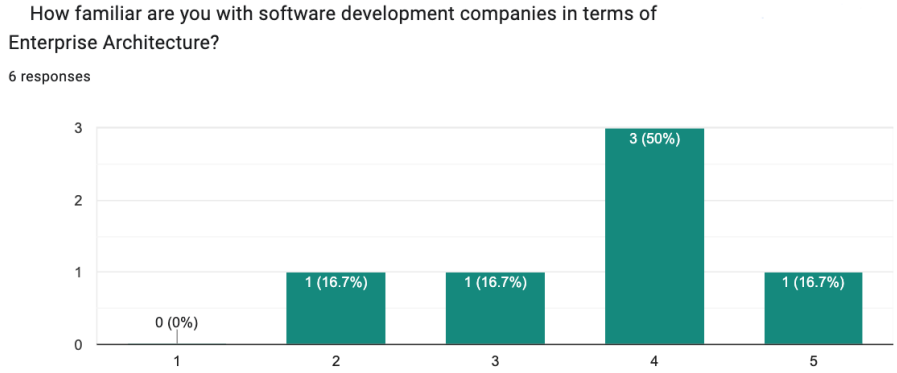

FIGURE 6.2: Familiarity with software development companies in terms of enterprise architecture

6.4.2 Likert scale questions results

Q1. *How understandable and easy to follow are the steps of the proposed modern EA method?*

The results of this question are presented in Figure 6.3. Here it can be seen that 66.7% of participants found the designed method very easy to understand and follow. 16.7% rated that the method is easy, and 16.7% of participants voted for neutral.

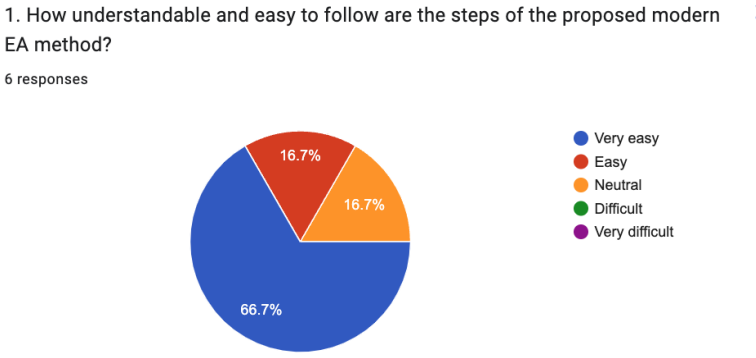

FIGURE 6.3: First question result

Q2. *How well does the method incorporate agile principles for technology-native companies?*

Based on the result in figure 6.4, 83.3% of participants found that the method somewhat incorporated well with agile principles, while 16.7% of participants rated it as extremely well.

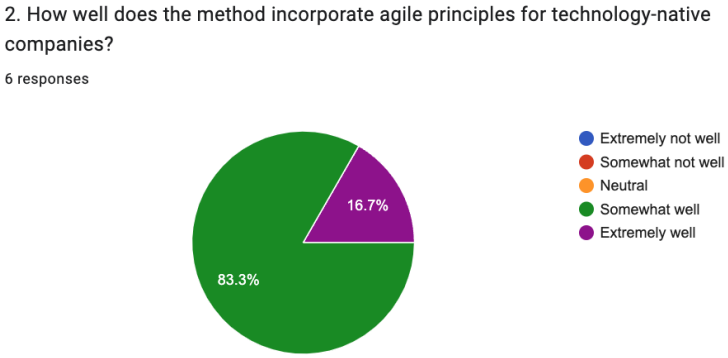

FIGURE 6.4: Second question result

Q3. *This method can be used independently by the user without requiring specific resources (such as a person or documentation)*

The results of this question are presented in Figure 6.5. 83.3% of participants agree that the method can be used independently. However, 16.7% of participants strongly agree that the method can be used independently without requiring specific resources.

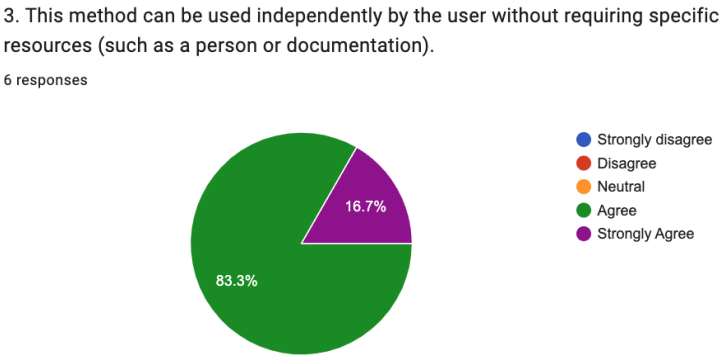

FIGURE 6.5: Third question result

Q4. *How useful do you think this method would be for EA practice in technology-native companies?*

The results of this question can be found in Figure 6.6. Here it can be seen that half of the participants think that this method is very useful for EA practice in technology-native companies and the other half think the method is extremely useful for these companies.

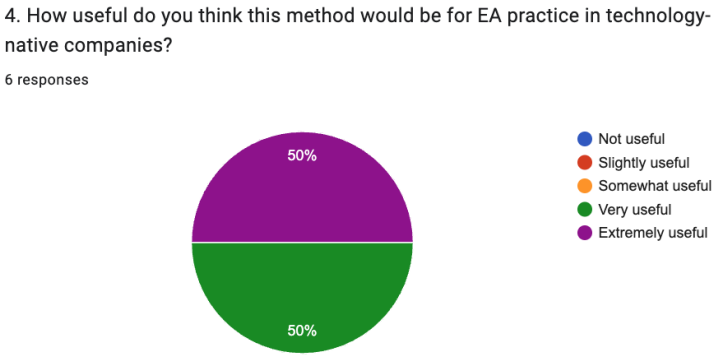

FIGURE 6.6: Forth question result

Q5. *How useful do you think implementation of this method would be in SAP LeanIX?*

Figure 6.7 shows the results of question 5. Here it can be seen that 66.7% of participants think the implementation of the designed method is extremely useful in SAP Leanix while 33.3% of participants think that the implementation is very useful. This question refers to the general use of the method in transformation projects within SAP LeanIX. Based on the results, it shows that most of the participants think that this method is extremely useful and valuable and the rest think it is very useful. It can help to improve processes in SAP LeanIX.

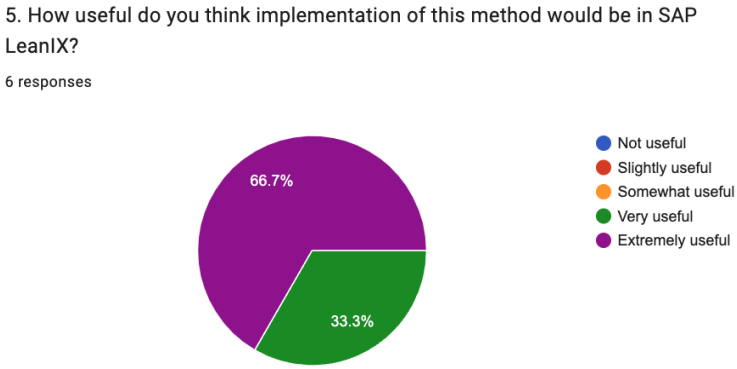

FIGURE 6.7: Fifth question result

6.4.3 Open-ended questions results

Q6. *What improvements would you suggest for the method?*

Most participants suggested considering AI in this method. Based on their feedback, AI potential is a huge topic in banking, high-tech, and other related fields. They recommended adding AI-related insights into the demo could make it more relevant, especially in areas such as build or buy decisions, compliance, and SBOM topics.

Another suggested improvement was to add more detail on how some of the steps could be performed. For example, in the "reuse existing microservices" step, looking for an already existing microservice requires knowing its features/functions vs what is being asked for.

Lastly, there was another improvement suggestion to focus more on collaboration within the designed method.

Q7 *What improvements would you suggest for the implementation of this method?*

Some participants found the implementation of the method useful. They noted that the implementation was well-structured, had good precision, and effectively demonstrated the framework. There is also some feedback to help improve the implementation which was:

- Ensure that all integrations are included so the implementation of the method is as fast and as automated as possible.
- Focus more on facilitating automated data discovery, which helps easily develop a baseline.

6.5 Discussion of results

The results of the demographic questions indicate that the participants of the validation process were highly familiar with EA. However, their familiarity with software

development companies within the context of EA was varied. Some participants were highly familiar with EA within software development companies, while other participants only had moderate familiarity. From these results, it can be said that the participants of the validation process are well-qualified to evaluate the designed method and its implementation.

The Likert scale questions provided valuable insights into the usability and effectiveness of the designed EA method. The majority of participants (66.7%) found the method very easy to follow and use, with an additional 16.7% of participants rating it as easy. This shows that the method is accessible and user-friendly for those with a background in EA.

The results of Q2 regarding the incorporation of agile principles showed that 83.3% of participants felt that the method was well-aligned with agile methodologies, while 16.7% rated it as extremely well incorporated. This shows that the method successfully integrates agile principles, and is adaptable for technology-native companies.

The results of Q3 regarding the independence of the designed method showed that most participants (83.3%) agreed that the method can be used without requiring specific external resources, while 16.7% strongly agreed.

Also, the perceived usefulness of the designed method was strongly positive. Half of the participants considered the method very useful, while the other half found it extremely useful. Similarly, the implementation within SAP LeanIX was also rated as extremely useful (66.7%) and very useful (33.3%). These results show that the designed method could be useful in real-world applications.

However, the results of the open-ended questions show that there is still room for improvement within the designed method and its implementation.

7 Conclusion

This chapter aims to answer the main research question, discuss the findings, and highlight the limitations and suggestions for future work.

7.1 Research questions & conclusion

In this section, the aim is to answer the main research question which is:

How can the enterprise architecture process be improved for technology-native organizations by designing an enterprise architecture method that satisfies the requirements of agility, scalability, and governance to help stakeholders create an adaptable and well-managed architectural framework?

This main question investigated a new EA method to fulfill the requirements of agility, scalability, and governance for technology-native companies. The goal was to design an EA method that helps technology-native companies build and maintain an agile, scalable, and well-governed EA at the same time. By doing this, the stakeholders can have a flexible framework to help them make better decisions, increase efficiency, and align with their business goals. The following sub-questions were formulated to address the main research question. The following part describes the results of each sub-question.

SQ1 What is the state of the art on agile principles in enterprise architecture?

The state of the art on agile principles in EA has been addressed from two perspectives. The first is the theoretical perspective. Integrating agile methodologies with EA is useful for organizations to be flexible. However, there are some challenges with integrating EA with agile methodologies. Key challenges include balancing long-term EA strategy with short-term agility, ensuring collaboration and communication between different teams, and maintaining governance while allowing innovation. There are some motivations for integrating agile with EA which are: reducing overall cost, quicker delivery, and minimizing rework. There are some required changes for the integration of agile and EA which is: the combination of agile frameworks with traditional waterfall models and based on these insights, there were only a few methods that were designed for agile EA.

From the practical side, the interviews results shows that the current state of the art lacks the automation to reduce manual work.Also, some companies keep developing new microservices or applications without checking if the existing applications are already there that have the same functionality, the result is having lots of microservices that have the same functionality. Using SBOM is useful for these types of companies as they use open-source libraries and components so they can be aware of the risks of these libraries.

SQ2 How can a modern EA method be designed to support technology-driven organizations?

A list of design requirements were made with the insights gained during the problem investigation. This list can be seen in Table 4.1 and was used to design a high-level modern EA method to support technology-driven organizations. The resulting high-level method is shown in Figure 4.1. The steps within the high-level method were detailed out resulting in a low-level method which is shown in Figures 4.2 and 4.3.

SQ3 To what extent does the designed EA method help technology-native organizations to work better in agile environments?

To answer this question, both expert validation and reflections on the implementation were used. The validation involved six experts from SAP LeanIX who evaluated the designed method and its implementation within the SAP LeanIX workspace. The focus was on usability and ease of use of the designed method and its implementation. The findings of the validation have been discussed in Section 6.5. The results show that the method provides notable benefits for technology-native organizations.

Experts who helped with the validation found this method useful and easy to use. They also gave some suggestions for improving this method such as using AI for decision making. Their improvement could enhance the efficiency and usability of this method later. The findings of SQ3 for validation were not limited to the use of SAP LeanIX and they can be generalized.

After the implementation, a reflection of the designed method has been done and can be used as part of the validation. During the reflection, which can be found in Section 5.5, it has been found that the designed method missed an important step. The missing step was the application assessment, and after realizing that it was missing, it has been added to the designed method.

Overall, this research successfully developed a new approach for modern EA for technology-native organizations, integrating agile principles into traditional EA. The study began with a problem investigation from a theoretical perspective (literature review) and practical perspective (interviews) which focused on analyzing the existing EA frameworks, the challenges, motivation, and requirements to integrate with agile principles which helped to gain better insights into this topic. With these insights, the method was designed also by inspiring TOGAF ADM. Within this method, different agile principles have been used to make the method useful for technology-native companies such as automation, reusing existing microservices, and integrating into CI/CD pipelines. The method was implemented within the SAP LeanIX tool which means following the design method's steps within this tool to show how this method could work in a real-world EA setting. To test the usability and usefulness of this method and its implementation, they have been evaluated by

using expert opinions. Based on the result of the validation, most participants found the method useful for technology-native companies and they thought the method somewhat well incorporated agile principles for technology-native companies.

7.2 Discussion

The designed method was implemented using the SAP LeanIX workspace but it does not depend on any specific tool and it is generalized. There are various tools available for all the steps within this method and the SAP LeanIX is an example of how this method can be applied. So different tools can be used for the implementation. The main focus of this method is to integrate the agile principle with EA so technology-native companies can use it, independently of the tool they choose to use.

The designed method only has one iteration but since agile methods have multiple iterations adding more iterations in different steps of the designed method would improve it. As the time was limited, there was no time to add more iterations.

Another area that can be improved is communication. Based on the literature review, communication between different teams is crucial. However, this method does not focus enough on this aspect. The reason is that there was a lot of theoretical discussion on communication but less focus on the practice side. Most studies in section 2.2 focus on how agile principles can integrate with EA in theory, but a few methods show how this can work in real-world scenarios. According to this, the designed method focuses on the more practical side and pays less attention to communication but still the stakeholders that communicate with each other for different activities are shown in the low-level method and also explained in Chapter 4.

As already mentioned in the validation result, AI can enhance the designed method. AI can be used in different steps within this method. For example, it can be used in updated business goals. It can analyze customer feedback and market trends to help companies figure out which part they can improve. By tracking customers' behavior and checking the reviews or surveys, AI can find the main common issues and customers' needs. According to this, AI can suggest ways for the company to make improvements. It is important to test AI before applying it to ensure that it works properly and correctly. The testing can take time as it requires accurate planning to see how AI can integrate with different steps of this designed method and to see if it can add any value and be useful for this method.

SAP LeanIX has various use cases to help organizations manage their EA within their workspace. Based on this study, SAP LeanIX could provide a new use case tailored for technology-native companies to help them with using the workspace. The use case could focus on better support for agile principles in EA to help the organizations make quicker and better decisions, and more automation to adapt to fast changes. By having this use case, SAP LeanIX can provide better support for technology-native companies.

7.3 Limitations

This study has provided useful insights for agile EA, but several limitations need to be noted. The following part explains the limitations:

- The main focus of this method is intended if companies want to replace applications and there are two options which are build or buy applications.
- There were a limited number of participants to validate this method. It would be better if more participants from different teams could attend for the validation.
- All participants for the validation were from SAP LeanIX company.
- SAP LeanIX was the only environment considered for implementing the designed method.
- There was a time limitation for the implementation of this method. As SAP LeanIX offers more advanced features, the implementation could improved by adding extra features or exploring other use cases.

7.4 Future research

The following future research has been suggested for future improvement:

- For validating the method, it is important to conduct various case studies within different technology-native organizations to assess different aspects of the method and determine the challenges these organizations face when using it. The variation should focus on the following aspects: 1. the company type which affects how the method is used and find out the focus is on which part of the method. 2. application portfolio complexity which can be companies with few applications to companies with complex environments, and 3. experience with EA which means how the company is experienced with EA to assess different aspects of the method and find the challenges these organizations face when using this method.
- Based on the validation result, an improvement for the designed method could be the integration of it with AI. As discussed in section 7.2, AI can help to provide better insights. It can be used in areas such as build vs. buy decisions, compliance, and SBOM management.
- The method was only implemented using the SAP LeanIX tool. For future work, it is important to explore how the method can be implemented with other tools and how the implementation be effective without the SAP LeanIX tool.

Bibliography

[1] R. Perez-Castillo, F. Ruiz, M. Piattini, and C. Ebert, "Enterprise architecture," *IEEE Software*, vol. 36, no. 4, pp. 12–19, 2019.

[2] D. Greefhorst and H. Proper, "The role of enterprise architecture," 01 2011.

[3] LeanIX, "What is enterprise architecture?," 2024.

[4] H. Guo, J. Li, and S. Gao, "Understanding challenges of applying enterprise architecture in public sectors: A technology acceptance perspective," in *2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW)*, pp. 38–43, 2019.

[5] LeanIX, "Welcome to the user documentation for leanix eam," 2025.

[6] LeanIX, "About us," 2024. Accessed: 2024-11-17.

[7] E. Siachou, D. Vrontis, and E. Trichina, "Can traditional organizations be digitally transformed by themselves? the moderating role of absorptive capacity and strategic interdependence," *Journal of Business Research*, vol. 124, pp. 408–421, 2021.

[8] U. Schultze and W. J. Orlikowski, "Metaphors of virtuality: shaping an emergent reality," *Information and Organization*, vol. 11, no. 1, pp. 45–77, 2001.

[9] Asana, "Agile methodology: What it is, how it works, and why it matters," 2024.

[10] S. Hanschke, J. Ernsting, and H. Kuchen, "Integrating agile software development and enterprise architecture management," in *2015 48th Hawaii International Conference on System Sciences*, 2015.

[11] IASA - The Business Technology Architect Organization, "Where to start with enterprise architecture?," 2025. Accessed: 8 Mar. 2025.

[12] R. J. Wieringa, *Design Science Methodology for Information Systems and Software Engineering*. Springer, 2014.

[13] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele Univ.*, vol. 33, 08 2004.

[14] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, Association for Computing Machinery, 2014.

[15] R. A. Hopkins and S. Harcombe, "Chapter 12 – agile architecting: Enabling the delivery of complex agile systems development projects," 2014.

[16] M. Hauder, S. Roth, C. Schulz, and F. Matthes, "Agile enterprise architecture management: An analysis on the application of agile principles," in *Proceedings of the 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations*, pp. 138–145, IEEE, 2014.

[17] R. Wessel, P. Kroon, and H. J. de Vries, "Scaling agile company-wide: The organizational challenge of combining agile-scaling frameworks and enterprise architecture in service companies," *IEEE Transactions on Engineering Management*, vol. PP, pp. 1–14, 12 2021.

[18] T. Akinpelu, R. van Eck, and T. Zuva, "Agile architecture frameworks: Challenges and open issues," in *Proceedings of the 2021 IEEE International Conference on Software Architecture (ICSA)*, pp. 11–20, IEEE, 2021.

[19] R. Duijs, P. Ravesteyn, and M. van Steenbergen, "Adaptation of enterprise architecture efforts to an agile environment," in *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS)*, pp. 427–434, SciTePress, 2018.

[20] Atlassian, "What is safe?," 2024. Accessed: September 5, 2024.

[21] M. Canat, N. Catala, A. Jourkovski, S. Petrov, M. Wellme, and L. Robert, "Enterprise architecture and agile development: Friends or foes?," pp. 176–183, 10 2018.

[22] Ö. Uludag, M. Kleehaus, N. Reiter, and F. Matthes, "What to expect from enterprise architects in large-scale agile development? a multiple-case study," 08 2019.

[23] Z. Askarinejad, "Challenges and weaknesses of agile method in enterprise architecture," *International Journal of Computer Science & Engineering Survey*, vol. 3, pp. 37–45, 12 2012.

[24] S. Buckl, F. Matthes, I. Monahov, S. Roth, C. Schulz, and C. M. Schweda, "Towards an agile design of the enterprise architecture management function," in *2011 IEEE 15th International Enterprise Distributed Object Computing Conference Workshops*, pp. 322–329, 2011.

[25] M. Nakayama, E. Hustad, and N. Sutcliffe, "Agility and system documentation in large-scale enterprise system projects: a knowledge management perspective," 02 2021.

[26] P. Medeiros, A. Santana, M. Lima, H. Moura, and M. M. da Silva, "An agile approach for modeling enterprise architectures," in *Proceedings of the 23rd Americas Conference on Information Systems (AMCIS 2017)*, 2017.

[27] H. Guo, D. Smite, J. Li, and S. Gao, *Enterprise Architecture and Agility: A Systematic Mapping Study*, pp. 296–305. 07 2021.

[28] Amazon Web Services, "Microservices on aws," 2025.

[29] A. W. Services, "What is service-oriented architecture (soa)?," 2025.

[30] H. Guo, J. Li, S. Gao, and D. Smite, "Agile enterprise architecture by leveraging use cases," pp. 501–509, 04 2021.

[31] P. Cammin, L. Heilig, and S. Voss, "Assessing requirements for agile enterprise architecture management: A multiple-case study," 01 2021.

[32] Ö. Uludag, N. Reiter, and F. Matthes, *Improving the Collaboration Between Enterprise Architects and Agile Teams: A Multiple-Case Study*, pp. 347–366. 01 2021.

[33] Ö. Uludag, S. Nägele, and M. Hauder, "Establishing architecture guidelines in large-scale agile development through institutional pressures," 08 2019.

[34] P. Medeiros, A. Santana, M. Lima, H. Moura, and M. M. da Silva, "An agile approach for modeling enterprise architectures," *Academia.edu*, 2021.

[35] LeanIX, "Togaf: The open group architecture framework," 2024. Accessed: 2024-09-05.

[36] T. O. Group, "The togaf standard - microservices architecture," 2021. Accessed: September 5, 2024.

[37] S. Dyck and T. A. Majchrzak, "Identifying common characteristics in fundamental, integrated, and agile software development methodologies," in *Proceedings of the 45th Hawaii International Conference on System Sciences (HICSS)*, pp. 5299–5308, IEEE Computer Society, 2012.

[38] K. Conboy, "Agility from first principles: Reconstructing the concept of agility in information systems development," *Information Systems Research (ISR)*, vol. 20, no. 3, pp. 329–354, 2009.

[39] F. Ahlemann, E. Stettiner, M. Messerschmidt, and C. Legner, eds., *Strategic Enterprise Architecture Management: Challenges, Best Practices, and Future Developments*. Berlin, Heidelberg: Springer, 2012.

[40] M. Op't Land, E. Proper, M. Waage, J. Cloo, and C. Steghuis, *Enterprise Architecture: Creating Value by Informed Governance*. Berlin, Heidelberg: Springer, 2009.

[41] Atlassian, "What is large-scale scrum (less)?," 2024. Accessed: September 5, 2024.

[42] Atlassian, "What is the spotify agile model?," 2024. Accessed: September 5, 2024.

[43] E. Kornyshova and R. Deneckère, "A proposal of a situational approach for enterprise architecture frameworks: Application to togaf," in *Procedia Computer Science*, vol. 207, pp. 3493–3500, Elsevier, 2022. 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022).

[44] Ö. Uludag, P. Philipp, A. Putta, M. Paasivaara, C. Lassenius, and F. Matthes, "Revealing the state of the art of large-scale agile development research: A systematic mapping study," 07 2022.

[45] Amazon Web Services, Inc., "What is cloud native?," 2024.

[46] S. Khan, "Layered, microservices, and modular monolithic," 2024.

[47] A. W. Services, "Automated governance - aws well-architected devops guidance," 2024.

[48] Amazon Web Services, "What is devops?," 2025.

[49] . C. Resources, "Structured vs. unstructured interviews: Key differences & benefits," 2025.

[50] ProductPlan, "Technical debt," 2025.

[51] P. Consulting, "Capability map," 2025.

[52] Asana, "Value streams," 2025.

[53] A. W. Services, "What is devsecops?," 2025.

[54] Testim, "What is test automation?," 2025.

[55] Amazon Web Services, "What is a service mesh?," 2024.

[56] Google Cloud, "What is kubernetes?," 2024.

[57] Adobe, "Learn about the input-output model," 2023.

[58] Lucidchart, "Flowchart symbols and their meanings," 2024.

[59] Wikipedia contributors, "General Data Protection Regulation — Wikipedia, The Free Encyclopedia," 2025.

[60] Wikipedia contributors, "ISO/IEC 27001," 2025.

[61] LeanIX, "Business capability," 2025.

[62] LeanIX, "Business capability map: Examples and templates," 2025.

[63] Miro, "What is dependency mapping?," 2025.

[64] LeanIX, "Integration architecture," 2025.

[65] P. P. M. Institute, "Project risks: Causes, effects, and management," 2025.

[66] LeanIX, "Software bill of materials (sbom)," 2024.

[67] Business.com, "What is real-time monitoring and why is it important?," 2024. Accessed: January 19, 2025.

[68] LeanIX, "Leanix meta-model documentation," 2025.

[69] LeanIX, "LeanIX Reports Documentation," 2025.

[70] LeanIX, "Report Views Documentation," 2025. Accessed: 2025-02-15.

[71] Amazon Web Services, "What is Governance, Risk, and Compliance (GRC)?," 2025.

[72] LeanIX, "Landscape report documentation," 2025.

[73] LeanIX, "Functional fit description," 2025. Accessed: February 4, 2025.

[74] LeanIX, "6r documentation," 2025.

[75] LeanIX, "Leanix diagrams documentation," 2024.

[76] LeanIX, "Circle map report documentation," 2025.

[77] LeanIX, "What is technology obsolescence?," 2024.

[78] LeanIX, "Cost management documentation," 2025.

[79] LeanIX, "Creating fact sheets," 2024.

[80] GitLab, "The ultimate guide to sboms," October 2022.

[81] LeanIX, "Working with transformations," 2024.

[82] LeanIX, "Collaboration," 2024. Accessed: February 6, 2025.

[83] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User acceptance of information technology: Toward a unified view," *MIS Quarterly*, 2003.

[84] S. McLeod, "Likert scale: Definition, examples, & analysis," 2023.

[85] D. Goerzig and T. Bauernhansl, "Enterprise architectures for the digital transformation in small and medium-sized enterprises," *Procedia CIRP*, vol. 67, pp. 540–545, 01 2018.

[86] Wikipedia contributors, "Digital transformation – Wikipedia, The free encyclopedia," 2023. [Online; accessed 9-September-2024].

[87] A. Gill, "Agile enterprise architecture modelling: Evaluating the applicability and integration of six modelling standards," *Information and Software Technology*, vol. 67, 07 2015.

[88] Ö. Uludag and F. Matthes, "Large-scale agile development patterns for enterprise and solution architects," pp. 1–22, 07 2020.

[89] Ö. Uludag, N.-M. Harders, and F. Matthes, "Documenting recurring concerns and patterns in large-scale agile development," 07 2019.

[90] Y. I. Alzoubi and A. Mishra, "Enterprise architecture contribution in distributed agile software development," *Journal of Software: Evolution and Process*, vol. 35, no. 8, p. e2443, 2023.

[91] T. Dreesen, P. Diegmann, and C. Rosenkranz, "The impact of modes, styles, and congruence of control on agile teams: Insights from a multiple case study," in *Proceedings of the 53rd Hawaii International Conference on System Sciences*, (Grand Wailea, Hawaii), HICSS, 2020.

[92] Y. Alzoubi, A. Gill, and B. Moulton, "A measurement model to analyze the effect of agile enterprise architecture on geographically distributed agile development," *Journal of Software Engineering Research and Development*, vol. 6, 12 2018.

[93] Y. Alzoubi, A. Gill, and A. Al-Ani, "Distributed agile development communication: An agile architecture driven framework," *Journal of Software*, vol. 10, pp. 681–694, 06 2015.