Architects working agile

Methods and challenges

MARTIN WELLME
MASTER THESIS

Architects working agile
Methods and challenges

Martin Wellme

Master of Science in Network Services and Systems
School of Electrical Engineering
KTH, Royal Institute of Technology
Stockholm, Sweden 2019
Abstract

Enterprise Architecture (EA) is a discipline which is used for describing and designing an organisation's infrastructure and business processes. Agile methods are known for providing flexibility and adaptability in software development but can be applied to other areas as well. Nowadays, almost all aspects of a business should advance quickly which creates new challenges that did exist before and the agile way of working is very suitable these situations. This thesis looks into the challenges employees currently face when working with EA and how agile methods can be used to solve these issues.

To investigate this, 19 interviews have been done at an international manufacturer where its employees were asked about how they work, which agile practices they use and the challenges they face. The results of those interviews are presented statistically and compared to the literature review as well as two additional interviews done outside of the company in order to find agile methods that could be possible solutions to the company's challenges.

The interviews showed that the employees already work agile with most of them working iterative, incremental and implementing changes based on feedback from the business which are all prominent agile methods. A few challenges which can be solved through agile were found, one of those is better project prioritisation found in Kanban to solve the lack of resources. Another practice is to have forums, inspired by SAFe ART, between different roles to address the lack of coordination and contact between them. The location of the EA office was a challenge which could be solved through a non-agile way, by moving it away from the IT department and closer to the business or alternatively move it higher up within the IT organisation.
Sammanfattning


För att undersöka det här, har 19 intervjuer gjorts hos en internationell tillverkare där deras anställda blivit frågade om hur de arbetar, vilka agila metoder de använder och vilka utmaningar de möter. Resultatet av intervjuerna presenteras statistiskt och jämförs med litteraturstudien samt med två ytterligare intervjuer som har gjorts utanför företaget för att hitta agila metoder som kan vara möjliga lösningar till företagets problem.

Intervjuerna visade att de anställda redan arbetar agilt med de flesta av dem arbetar iterativt, inkrementellt and implementerar förändringar baserat på feedback från verksamheten som alla är uppmärksamma agila metoder. Några av utmaningarna kan lösas med hjälp av agila metoder, en av dem är en bättre prioritering av projekt som finns i Kanban för att lösa bristen på resurser. En annan metod är att ha forum, inspirerade av SAFe ART, mellan olika roller för att åtgärda bristen på koordination och kontakt mellan dem. Placeringen av EA-kontoret var en utmaning som kunde lösas på ett icke-agilt sätt, genom att flytta det bort från IT-avdelningen och närmare verksamheten eller alternativt flytta det högre upp inom IT-organisationen.
# Table of contents

1 Introduction .................................................................................................................. 3  
1.1 Background ............................................................................................................. 3  
1.2 Problem .................................................................................................................. 3  
1.3 Method and methodology ....................................................................................... 4  
1.4 The company .......................................................................................................... 5  
1.5 Delimitations .......................................................................................................... 5  
1.6 Outline ..................................................................................................................... 5  

2 Theoretical background ............................................................................................... 6  
2.1 Enterprise Architecture ......................................................................................... 6  
  2.1.1 The Zachman Framework .............................................................................. 6  
  2.1.2 The Open Group Architecture Framework ................................................. 7  
  2.1.3 Enterprise Architecture Management ......................................................... 9  
2.2 Agile development .................................................................................................. 9  
  2.2.1 Scrum ............................................................................................................ 10  
  2.2.2 Kanban ......................................................................................................... 10  
  2.2.3 Extreme Programming .................................................................................. 11  
  2.2.4 Scaled Agile Framework .............................................................................. 12  

3 Methodology ............................................................................................................... 14  
3.1 Research strategy ................................................................................................... 14  
3.2 Data collection ....................................................................................................... 14  

4 Literature review ....................................................................................................... 17  
4.1 Enterprise Architecture Challenges ....................................................................... 17  
4.2 Agile Architecture .................................................................................................. 22  

5 Results ....................................................................................................................... 26  
5.1 Interview questions ............................................................................................... 26  
5.2 Methods ................................................................................................................ 27  
  5.2.1 Model creation ............................................................................................. 28  
  5.2.2 Standardised framework ............................................................................. 29  
  5.2.3 Regular forums ............................................................................................ 30  
  5.2.4 Less common methods ................................................................................ 31  
5.3 Challenges ............................................................................................................ 32  
  5.3.1 Location of the Enterprise Architecture office ............................................ 32  
  5.3.2 Contact with the Enterprise Architecture office ........................................... 33  
  5.3.3 Resources ..................................................................................................... 34  
  5.3.4 Coordination ................................................................................................ 35  
  5.3.5 Less common challenges ............................................................................ 37  
5.4 Agile and architecture ............................................................................................. 38  
  5.4.1 Enterprise architects .................................................................................... 39  
  5.4.2 Business architects ...................................................................................... 40  
  5.4.3 Solution architects ....................................................................................... 44  
  5.4.4 Portfolio managers ....................................................................................... 46  

6 Discussion .................................................................................................................. 47  
6.1 Discussion of the results ......................................................................................... 47  
  6.1.1 Caused by agile ............................................................................................. 47  
  6.1.2 Solved by agile .............................................................................................. 48  
  6.1.3 Non-agile related ........................................................................................... 52  
6.2 Limitation ............................................................................................................... 53
6.3 Validity and reliability ................................................................. 54
6.4 Future research ........................................................................... 54
7 Conclusion ......................................................................................... 55
References .......................................................................................... 56
1 Introduction

The agile methods and the agile way of thinking have in last decades been a popular approach for working, especially within software development. In more recent years Enterprise Architecture (EA) have been utilised more frequently as a way to manage both the business side and the IT side of an organization[32]. Even though they might seem incompatible there are efforts to combine the two approaches and there is data that shows that they can coexist[5][6]. There are multiple ways these two can collaborate, one method is to implement agile practices and principles into the current management processes. Another way is to use EA to help the developers keep track of their dependencies and the impact of changing or removing them[26]. Architecture could also be used by a company as a way to know what systems or applications are available to them in order to prevent them from buying or developing already developed solutions[27]. Another way is to introduce architecture into the agile way of working by having the developers work more with the architecture themselves. This study will look into to the first approach, how the architects can work more agile.

1.1 Background

Agile focuses on flexibility, speed and being able to adapt to change quickly which means that it prioritises people and software over processes and documentation. It usually involves working in iterations with more frequent deliveries and trying to improve after each one iteration to reduce inefficiencies.[1][2]

Enterprise architecture consists of a collection of models, principles and methods which describes the organisational, business and IT structure of an enterprise as well as their interrelationships. This is then used to document, design and analyse the organisation's current and future state in order to improve its quality, reduce costs and provide a holistic overview.[3]

Enterprise Architecture Management (EAM) is a management discipline consisting of a set of management practices that are used to help design and develop the architectural plan of an organisation as well as to support and improve the existing architecture. The scope of EAM is not limited to only EA models or IT functionalities, it is a holistic approach to understand and control an organization's architecture.[4]

The research about how EA and agile can collaborate is limited but there are evidence that they can coexist. The concern seems to be how they can learn from each other and to what extent they can be combined since low-level models can be too costly to maintain and the architects need to know the boundaries of the architecture[27].

1.2 Problem

Even though there are evidence that agile and Enterprise Architecture Management can exist side-by-side they are two different approaches and realising how they can complement each other is key. EAM have a more of a top-down perspective, focusing a lot on long term investments and benefits while agile often focuses more on a lower level and on short term values. As a consequence planning and migration is not
always top priorities for agile teams which can make them lose sight of the organization's overall ambition and could therefore benefit from the structure that enterprise architecture brings.[17] On the other hand, EAM can sometimes suffer from IKIWISI syndrome (I'll Know It When I See It) and “ivory tower” syndrome. The first syndrome occurs when the stakeholders do not know what they need or what the problem is until later on in a project, partly because things can change during the course of a project and they are unable to provide proper requirements to the teams in advance.[18] The second syndrome occurs when the architects are isolated from the rest of the company and the employees[20] which can cause the model to be too complex and abstract to work in practice[19]. These issues among others can possibly be mitigated by importing parts of agile into the organisation's EAM.[17]

The issue is to know which parts from agile should be implemented, not everything can be used for EAM and the practices need to fit the company in question. Methods that works in theory might not be as successful when applied to the real world at an actual company with real humans and the complications they bring. Before determining which agile methods should be implemented, the current way of working needs to be determined. Some agile methods might already be in use and the new ones might need to be adapted to the current practices. It is unsuitable to just introduce new methods for the sake of it, they need to fit the existing conditions and need to serve a purpose. For that reason the problems the architects experience with their current way of working should also be identified. After the current situation is established, new options can be explored. Therefore the research questions are the following:

• How does the company currently work and which agile methods are already implemented in the company's EAM?
• What are the challenges with current way of working?
• How can these challenges be solve and can agile methods be used to solve them?

The purpose of the research is to investigate how architects can use agile methods in order to improve their own way of working. The goal of the research is to determine the challenges the architects face with their current EAM and suggest agile methods that can be a possible solution to those issues.

### 1.3 Method and methodology

The research methods and methodologies describe how the research will be carried out and decides the character of the research, from how data is collected to how a conclusion is drawn[35]. Further details will be described in Chapter 3.

This research will primarily adopt a qualitative research method since the goal is to understand the behaviours of the participants and their opinions of their current way of working. An inductive approach is appropriate for this and for understanding the perspectives of the different employees. The data will be collected through interviews in order to get the individuals perspective on their methods and the challenges they face. To examine the data Narrative Analysis will be used to better understand the experiences of the employees[36]. Coding will also be utilised in order to convert the transcriptions into numerical information which will complement the qualitative
material with quantitative data. Statistics will be used to get a result from the new data and to evaluate that result.[35]

1.4 The company
The company the thesis is done at is an international manufacturer with over 30000 employees that has its headquarters in Europe but has production facilities all over the world. The data will be collected from a number of different departments and organisations at one of the company's larger offices which is located in Sweden. Both the production and the assembly of some the products are done at the office where different business organisations could be responsible for producing different parts of the same product. The different business organisations are fairly independent and the IT is for the most part centralised, only a few organisations have their own software development. Most of the business organisations have to go through the centralised IT department that is separate from the rest of the office, in a separate location which is a few minutes away by car. The Enterprise Architecture office is centralised as well at the IT department which is also where the solution architects are also located. The business architects are based at the different business organisations and the business architects interviewed were located at the departments for production and for research. In-depth knowledge of how the company's undying structures looks like, such as process structures, is limited to the information provided by interviewees.

1.5 Delimitations
The research is limited to the company described in the previous section with its centralised IT as well as EA office and decentralised business organisations. The employees chosen for the interviews are mainly architects because their way of working is the primary focus of the thesis. Other employees who work closely with the architects could be acceptable if their methods or challenges coincide with the architects. The participants perceptions of agile and architecture concepts may also vary from the definitions found in the literature review. The research follows those in the literature review unless otherwise stated which will be described to the interviewees if there are any uncertainties. Further delimitations about the chosen methodology of the research is described in Chapter 3.

1.6 Outline

Chapter 2 – Theoretical background
This chapter will describe the theoretical background of EA and agile development.

Chapter 3 – Methodology
This chapter will describe further how the study will be carried out.

Chapter 4 – Literature Review
This chapter will describe research that has been done which covers similar topics.

Chapter 5 – Results
This chapter will present the results of the literature review and the interviews.

Chapter 6 – Discussion
This chapter will discuss the results and the research as a whole.

Chapter 7 – Conclusion
This chapter will present the conclusions of the thesis.
2 Theoretical background

The two main areas of this study are enterprise architecture and agile development. Both disciplines have been around for a long time and there are several different ways to approach the concepts.

2.1 Enterprise Architecture

Enterprise Architecture (EA) started to take form during the 1980s with the Zachman framework created by John A. Zachman[4][45]. This introduced the idea of viewing the architecture from different angles[4] and stressed the importance of having a holistic view of it[45]. The discipline evolved during the following two decades when the landscape had become more complex thanks to new technologies and the architects realised they needed more than just modelling. This created a need for more planning, decision making and processes control.[4] During this period The Open Group Architecture Framework (TOGAF) was developed which focuses on the application landscape and is still one of the most known EA frameworks[45]. In more recent years the consensus have become that enterprise architecture is not just an IT discipline or function, it is more of a strategic and business function. Enterprise Architecture Management have therefore included more business strategy planning and implementation processes to be able to support the business better.[4] The EA models describes the current (as-is) or future (to-be) architecture of an enterprise. While the EA frameworks provide the enterprise with methods and templates for designing and evolving EA as well as meta-models and vocabularies for describing EA.[31]

2.1.1 The Zachman Framework

After his initial proposal in 1987 John Zachman released an updated version in 1992 which became the The Zachman Enterprise Architecture Framework that is still used today. Zachman's way of thinking and his principles have influenced many of the frameworks that followed[4],[45]

One of the most prevalent of those principles were to view the architecture from multiple perspectives and aspects, there six of each and can together be presented as a matrix (see Figure 1 below). The perspectives represents the rows in the matrix, they are: 1) Scope (Planner), 2) Enterprise Model (Owner), 3) System Model (Designer), 4) Technology Model (Builder), 5) Detailed Representations (Subcontractor) and 6) Functioning Enterprise (Enterprise). The lower the row the higher level of detail is involved. The different aspects of the architecture repeat the columns in the matrix, they are: 1) Data, 2) Function, 3) Network, 4) People, 5) Time and 6) Motivation. Together, the matrix describes the enterprise with each cell represent one dimension of the enterprise and the architecture is not complete unless every cell is provided.[4][45]
2.1.2 The Open Group Architecture Framework

The Open Group Architecture Framework (TOGAF) provides a number of methods and tools for producing, using and maintaining enterprise architecture in four different categories:[4][45][46]

1. **Business Architecture**, describes the processes the enterprise uses to reach its goals, such as strategy, governance, organisation and business processes.
2. **Data Architecture**, describes how the enterprise stores, organises and manage its physical and logical data.
3. **Application Architecture**, describes the applications, their interactions with other applications and their relationship with the business processes.
4. **Technology Architecture**, describes the enterprise's current and future infrastructure for both software and hardware capabilities.

TOGAF also provides an iterative process for developing architecture called Architecture Development Method (ADM) with different phases that can be done in whichever order the enterprise wants and can even be skipped or combined to fit its needs better (see Figure 2 below). There are 10 phases, eight of them represent the enterprise architecture life cycle phases, a phase for managing requirements and an initial phase which is not included in the cycle.[4][45][46] These phases are:[46]

0. **Preliminary phase**, preparatory work required for the creation of an architecture capability including framework and principle definition.
0. **Requirement Management**, a process for the management of the architecture requirements for the phases, handles changes and ensure their availability.
A. **Architecture Vision**, defines the vision, stakeholders, scope and principles of a proposed architecture development, then acquire the approval of them.

B. **Business architecture**, development of a business architecture which was defined in Architecture Vision.

C. **Information Systems Architecture**, development of an information systems architecture, which is the combination of data architecture and application architecture, that was defined in Architecture Vision.

D. **Technology Architecture**, development of a technology architecture which was defined in Architecture Vision.

E. **Opportunities & Solutions**, creates an architecture roadmap and an implementation planning based on the previous phases.

F. **Migration Planning**, creates an Implementation and Migration plan which describes how to reach the Target Architecture defined in Architecture Vision.

G. **Implementation Governance**, oversees the implementation to ensure it corresponds with the Target Architecture.

H. **Architecture Change Management**, process for change management of the new architecture, maintaining the life cycle and executing the governance.[46]

Figure 2: Visualisation of the TOGAF Architecture Development Method.[46]

TOGAF also defines Solution Architecture which is an architectural description of business operations or activities and how the IT system supports those activities, often specific to a single system or solution. It helps develop a solution based on the requirements by turning them into a solution vision, system specification and tasks. [46]
2.1.3 Enterprise Architecture Management

Enterprise Architecture Management (EAM) is the process of using the enterprise architecture to design situations to achieve certain goals or purposes set by the enterprise. Along with steering the enterprise's development as well as overseeing the architecture and adapting it to events that may occur.[18] It ensures that the business and IT follows the company's goals and strategy but not just for the current architecture, it also assists in developing the future architecture[17].

2.2 Agile development

Agile began in the 1990s as a reaction to the current situation for software development projects[39][40]. The IT industry was suffering from high failure rate caused by over-planning, bad communication and big deliveries. The project were not responding well to changes, the deliveries did not match the needs of the business and failed to keep within the budget or schedule. In response, new methods were developed to address these issues.[39] In 2001, a group of seventeen people met to discuss these new lightweight methods which resulted in the “Manifesto for Agile Software Development”. The group included representatives from Scrum, Extreme Programming as well as Feature-Driven Development among others and they called themselves “The Agile Alliance”. [41] The manifesto describes what the group values when it comes to software development[1]:

“Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan”[1]

The group still values the things to the right of “over” but they value the things on the left side more.[1] These values guide the various agile methods towards flexibility [43] and the group published twelve principles which should be followed as well: “

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”[42]

One of the drawbacks of agile is that other processes in the organisation that are not related to the software development need to change as well in order to get the most of the agile methods. These other processes could for example be concerned with demand management or strategic purchasing which need to conform to the agile way. [30]

2.2.1 Scrum

Scrum is the most common agile method and is based on iterative development[39] which focuses on dividing a project into smaller and more manageable parts called sprints instead of following a long-term schedule. Each iteration is usually a couple of weeks long and is almost like its own project with its own planning, design, development and testing but with a much smaller scope. This enables the teams to adapt better to incoming changes and they are able to produce incremental deliveries of a product[43] which they can receive feedback on that can be used in the future[40].

There are normally three roles within Scrum: the (scrum or development) team, the scrum master and the product owner. The product owner represents the need of the business or stakeholders and is responsible over the prioritisation of project's requirements. The team is responsible for the actual work of a sprint, such as development and testing of the deliveries The scrum master's responsibility is to ensure that the team follows the scrum process and that any obstacle the team faces is handled. There are two backlogs for each sprint, the product backlog and the sprint backlog. The project's requirements are put in the product backlog and is prioritised by the product owners. The sprint backlog contains the tasks that should be performed in the current sprint. At the start of a sprint, there is a sprint panning meeting where the three roles review the items in the product backlog to decide what should be in the new sprint backlog.[43] At the end of each sprint all roles meet for a sprint review where the result of the sprint is demonstrated as well as a retrospective where they evaluate the sprint in order to find improvements for the next sprint. There are also daily meetings throughout the sprint to get continuous status updates on the progress [40] and any problem can be brought up[43],[39]

2.2.2 Kanban

Kanban can be used by itself or together with other agile methods. It is a method for managing workload and change through the use of continuous deliveries without overloading the teams. This is done by limiting the number of tasks the team can take one at once and preventing the team from committing to new tasks when the limit is reach[44], unless an existing task is put on hold. An other feature is the Kanban board which used to visualise the workloads by showing the status of the current, finished and planned tasks, issues the team is facing as well as any free resources the team has. [39] Kanban puts an emphasis on fast deliveries just like Scrum but there are some differences between the two methods which can be seen in Table 1 below[44].
Table 1: Differences between Kanban and Scrum[44].

<table>
<thead>
<tr>
<th>Kanban</th>
<th>Scrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>No prescribed roles</td>
<td>Pre-defined roles of Scrum master,</td>
</tr>
<tr>
<td></td>
<td>Product owner and team member</td>
</tr>
<tr>
<td>Continuous Delivery</td>
<td>Timeboxed sprints</td>
</tr>
<tr>
<td>Work is pulled through the system</td>
<td>Work is pulled through the system in</td>
</tr>
<tr>
<td>(single piece flow)</td>
<td>batches (the sprint backlog)</td>
</tr>
<tr>
<td>Changes can be made at any time</td>
<td>No changes allowed mid-sprint</td>
</tr>
<tr>
<td>Cycle time</td>
<td>Velocity</td>
</tr>
<tr>
<td>More appropriate in operational environments</td>
<td>More appropriate in situations where work can</td>
</tr>
<tr>
<td>with a high degree of variability in priority</td>
<td>be prioritised in batches that can be left alone</td>
</tr>
</tbody>
</table>

2.2.3 Extreme Programming

Extreme Programming (XP) is an iteration based method that focusses on simple but fast deliveries by incorporating customer feedback and assuming that requirements may change in the future[39]. XP incorporates already established concepts but takes it to an extreme level. There are 12 extreme concepts:[43]

1. Test driven development, a method which focuses on writing the test cases directly after the design of a functionality and before it is developed.
2. Pair programming, having two developers work on the same task using the same computer which results in better code and less bugs but requires practice.
3. Refactoring, regularly reviewing the code in order to improve it and even remove parts of it for the long-term benefits.
4. Simplicity, focus on delivering the simplest solution possible as long as it satisfies the customer's needs because the requirements can change quickly.
5. Planning game, the customers prioritises the requirements then meets with the team at the start of every iteration to decide what should be implemented.
6. Small releases, the team should deliver a useful and working system each iteration so the customer can change priorities without the system breaking.
7. Continuous integration, integrating new code into the product and using automated testing in order to have a full product build every day.
8. Continuous testing, to make sure the daily builds are working properly.
9. Collective code ownership, every member of the team should understand the entire code and be able to work on any part of it, unless the team is very big.
10. Sustainable pace, not overworking the developers by limiting the amount of overtime thus keeping the team productive and effective for the entire project.
11. Coding standards, project specific guidelines for quality and maintainability.
12. On-site customer, the teams need to have close contact with the customers in order to be able to adapt to their feedback and change of requirements.[39][43]
2.2.4 Scaled Agile Framework

Scaled Agile Framework (SAFe) combines Lean, a methodology for creating value while reducing waste[53], and Agile concepts with architecture by providing holistic and architectural views of how the concepts can be applied. It incorporates practices from other agile methods but scales them up to be more suitable for larger enterprises who usually develop bigger systems and already have an established architecture. SAFe is designed around four core values and nine principles.[47] The four core values are:

1. **Alignment**, to align the teams and management towards a common goal through communication and everyone understanding the strategy.
2. **Built-in Quality**, to ensure customer satisfaction, value delivery and high quality in every aspect of the solution.
3. **Transparency**, to build trust which promotes performance, innovation, improvement and decentralised decisions. “You can't manage a secret”.
4. **Program Execution**, to ensure development and value delivery as well as advocating change. Through the entire process, from concept to release.[38]

The nine principles are: “

1. **Take an economic view.**
2. **Apply systems thinking.**
3. **Assume variability; preserve options.**
4. **Build incrementally with fast, integrated learning cycles.**
5. **Base milestones on an objective evaluation of working systems.**
6. **Visualize and limit work in process, reduce batch sizes, and manage queue lengths.**
7. **Apply cadence, synchronize with cross-domain planning.**
8. **Unlock the intrinsic motivation of knowledge workers.**
9. **Decentralize decision-making.**[38]

To aid the alignment, SAFe introduces an organisational structure called the Agile Release Train (ART) which consists of several agile teams and other stakeholders. ART aligns the teams to work towards a common and to develop as well as deliver the solutions together. The collaboration exists across the normal organisational silos by incorporating people from different parts of the company who usually do not work together without management involvement. This speeds up the process, creates a more cross-functional organisation and encourages communication outside your own team. [50]

SAFe even recognise the need for architecture and that insufficient investment into architecture will lead to a slow down in development[50]. Their approach for this is a collection of practices and values called Agile Architecture which enables the architecture to evolve over time through two main concepts, Emergent design and Intentional architecture. The former is having the architecture emerge over time as the system is developed and focuses on adapting to the current needs or functionality. The latter focuses more on the bigger picture by planning the architecture. Together the two concepts form the Architectural Runway which represents the company's technical infrastructure required for future development. It is important to find the balance between emergent design and intentional architecture, they need to complement each other in order to have both fast control and a holistic view.[34]
An example of how the architecture is used is with the Portfolio SAFe configuration which coordinates the Agile development with the enterprise strategy by organising it around one or more value streams and their solutions. Each value stream has their own funding for developing their solutions and they utilise Kanban to ensure that demands are aligned with the value streams and ARTs. The capture and reflection of business capabilities can only be done through the coordination of the value streams which is handled by the enterprise architects who also provide strategic guidance to all the value streams. In this configuration the development and architecture tasks are in the same backlog called the Portfolio backlog and are prioritised together.[33][50]
3  Methodology

The study started with a literature review and the questions for the interviews were based on the results of it. The interviews were done with employees of the company where they were asked about the methods they use and the challenges they face. The data from those interviews were then converted to make it easier to analyse.

3.1  Research strategy

The first phase of the research consisted of a literature review about the challenges of Enterprise Architecture Management and the practices of Agile Architecture. The data was obtained by searching for full-text research and conference paper in various databases such as Google Scholar\(^1\), IEEEXplore\(^2\), Springer\(^3\) and KTH Primo\(^4\) using relevant keywords and phrases. The main phrase that were searched for was “Agile Enterprise Architecture Management” and more results were achieved by placing the quotation marks in different positions. In order to find more general information, individual or multiple words from the main phrase were used, such as “Enterprise Architecture”, “agile architecture” or just “agile”. Additional data were gathered by examining the references of previous sources and papers that have cited them.

3.2  Data collection

In order to determine how current practices looked like, interviews were conducted with different types of architects. The interviews were semi-structured with mostly open-ended questions in order to steer them towards more of a discussion rather than a survey. This was to get a more accurate representation of the current situation of the company without forcing the answers or the direction of the interview. It would also make the interviewee talk for freely and provide additional information they might not have given in a stricter environment. Everyone should be able answer the questions but at the same time they needed to be on a high enough level that they would produce a useful result. The interviews were compared to the outcome of the literature review, which practices observed in the research do the company actually use and which challenges do they face. The literature review were be used to examine if there were any additional agile practices which could benefit the company or addressed its issues. A few interviews were also conducted outside of the company which were used as references for the discussion. These external interviews followed the same format as the other interviews and provided new perspectives on the methods used by the architects and the challenges they faced.

---

\(^1\) Google Scholar: https://scholar.google.se/

\(^2\) IEEEXplore: https://ieeexplore.ieee.org/

\(^3\) Springer: https://www.springer.com/

\(^4\) KTH Primo: https://www.kth.se/en/kthb
There is an issue of what kind of data is required for this type of study and therefore which methods are needed. Interviews would provide qualitative data which would be useful but might not be ideal for an end result. It could be hard to get sufficient amount of data, drawing measurable conclusions could be difficult and you could easily to miss your target with only qualitative data. To complement the qualitative data with quantitative data is a way to reduce these drawbacks and it is more suitable for a measurable result. To achieve this, the result of the interviews were converted into quantitative data by categorise it into four categories:

1) A certain method was used or a certain challenge existed.
2) The method was partially used or the challenge only existed to a certain degree.
3) The method was not used or the challenge did not exist.
4) The method or challenge was not mentioned, or there were no clear answer.

This new data were used to present the common methods and challenges as well as their frequency. The frequency was determined by grouping up the first two categories to represent that a certain method or a challenge existed, and the last two categories were grouped up to represent that the method or challenge did not exist. Interviews where the method or challenge was not mentioned were considered the same as if the interviewee said it did not exist. This was because some methods and challenges would not be relevant for every role or employee, and would therefore not be mentioned during those interviews. If a challenge or method was important enough, it was assumed that the interviewee would have mentioned it. Finding a solution to a challenge that does not exist was unnecessary and the consequences of the solution to that challenge could have an actual negative impact. An undetected challenge would still exist after the study and could be detected by the company afterwards. An unidentified method would not be too severe either, as it could possibly be identified as a potential solution if it was beneficial enough. It was therefore preferred to potentially risk that methods or challenges go unnoticed than to overestimate something that was not relevant or did not exist.

Every interview were recorded in order to reduce the need for continuously writing down what were being said instead of listening to the interviewee, notes were still made for follow-up questions. The voice recordings were then transcribed and paraphrased, if needed also translated into English. Transcriptions are easier to go through compared to voice recordings when looking for information but the recordings can still be used if anything needs to be clarified. Both the company and the employees were anonymous in the study in order to ensure the privacy of the interviewees. It was completely voluntary to participate in the interviews and they were asked for their consent to record the interview. The voice recordings were not be shared with anyone and only author of the thesis had accessible to them.
All the transcriptions and recordings were used for compiling every method and challenge mentioned in each interview. Those mentioned by multiple interviewees were summed up and organised according to the four categories mentioned above as well as by the role of the interviewee. This generated the quantitative data which were used to determine the frequency of the most common methods and challenges.

The first interviewees were taken from a list of potential interviewees recommended by the head of an architecture group at one of the business organisations who thought the people on the list would fit the study. The rest of the interviewees were recruited by asking people on location if they were available or asking during an interview if they could recommend someone else for the study. The target interviewees were architects of different types: enterprise, business and solution architects. Other roles could provide different perspectives although they are not the focus of this study.
4 Literature review

The research papers, books and online articles were found by searching for keywords in various databases and by looking through the references in previous finding as detailed in Chapter 3. Most of the findings focuses on either enterprise architecture or agile methods, only a few of them examines the combination of the two approaches and even fewer were based on empirical research. Many of the resources found discussed how to make agile development more structured which is the reverse of the thesis' aim and will not be in the scope of the thesis. The findings of the literature review will form the foundation for the questions that will be asked during the interviews.

4.1 Enterprise Architecture Challenges

The challenges for EAM and how agile method, especially from Scrum can help solve them is describe in a conference paper from 2011 by Buckl et al. [15]. In this paper the challenges identified were (1) aligning EA efforts with the stakeholders' interest in a shared terminology, (2) ensuring early and periodically delivery EA products, (3) ensuring commitment and involvement of all parties and (4) adapting to an environment and goals that constantly changes. One of the agile idea suggested to help against the first challenge was the pull-principle which is when the product owner prioritises the product backlog and the development team then decides how much of the functionality will be delivered during the sprint[16]. The usage of this principle would give the stakeholders and EA management a shared vision which would help the creator of the EA models align them with the interests of the stakeholders.[15] The application of the pull-principle was included in the survey by the Technical University in Münich[11] where it was the fourth less used agile principle and only 33% of the participants agreed that it was a part of their EAM practices while 51% said it was not (14% said neither and 4% did not respond).

In another paper[12] from the Technical University in München they investigates the challenges within EAM. The most prominent issue found were ad hoc EAM demands which are one-off spur of the moment demands that are unexpected and usually fall outside standard practices[13][14]. Other top challenges includes unclear business goals and difficulties finding experienced enterprise architects. Challenges that occurred frequently but simultaneously were not an issue for other participants were analyse to determine if organisational factors could have influenced theses challenges. The top challenges were that the “enterprise environment changes too quickly”, “EAM team focuses primarily on IT” and “reluctant information providers”. The organisational factors which influenced those were the industry sector of the company, size of the company and the modelled state of the enterprise architecture respectively. The top ten challenges found in the paper can be seen in Table 2 below.[12]
Table 2: Top ten EAM challenge in practice[12].

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Agree</th>
<th>Neither</th>
<th>Disagree</th>
<th>No response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad hoc EAM demands</td>
<td>89</td>
<td>6</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>UNCLEAR BUSINESS GOALS</td>
<td>84</td>
<td>11</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Hard to find experienced enterprise architects</td>
<td>82</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>EA demands unclear for EAM team</td>
<td>74</td>
<td>13</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>Enterprise environment changes too quickly</td>
<td>70</td>
<td>9</td>
<td>19</td>
<td>4</td>
</tr>
<tr>
<td>Conflicting interests among stakeholders</td>
<td>69</td>
<td>15</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>EAM team focuses primarily on IT</td>
<td>67</td>
<td>9</td>
<td>23</td>
<td>3</td>
</tr>
<tr>
<td>Reluctant information providers</td>
<td>62</td>
<td>14</td>
<td>20</td>
<td>6</td>
</tr>
<tr>
<td>Unavailable stakeholders</td>
<td>49</td>
<td>26</td>
<td>21</td>
<td>6</td>
</tr>
<tr>
<td>Late valuation of EAM through stakeholders</td>
<td>47</td>
<td>26</td>
<td>19</td>
<td>10</td>
</tr>
</tbody>
</table>

Other common challenges were described and listed by a consultant who have worked for many different organisations[6]. No organisation had every problem on the list but some had the majority of the issues. These were the common challenges he had observed:

1. There isn't an enterprise architecture effort.
2. Skewed focus.
3. Project teams don't know the enterprise architecture exists.
4. Project teams don't follow the enterprise architecture.
5. Project teams don't work with the enterprise architects.
6. Outdated architecture.
7. Narrowly focused architecture models.
8. Dysfunctional “charge back” schemes.
9. A “do all this extra work because it's good for the company” attitude.”[6]

In a literature review study by Lucke et al. they categorise enterprise architecture problems into five categories: Management, Semantic problems, Insufficient resources, Complexity and Representation. They also categorise them into two groups on a more abstract level, understanding & management of EA and modelling of complex systems. There are 14 types of problem which they divided into the five categories with some overlap which is presented in Figure 3 below.[21]
Insufficient management commitment can make it more difficult to persuade the management the value of EA efforts and related to this is the lack of central architecture authority. They do not have control over the decisions that are made which affects the information system architecture because of the management bypasses the review boards and the lack of investment that is needed for proper architecture. Inadequate EA governance in EA projects sometimes cause them to lose focus on their goals when there are no common principles or guidelines for the EA development process. The problem with stakeholders comes from not enough involvement from them but also by having too many stakeholders or them having too different needs and opinions which both makes it harder for them to come to an agreement over architectural definitions. The coordination can occur when there are multiple organisational units, departments or architects from different EA layers working together. With EA involving many different groups of people there are some issues with communication between them, such as not speaking the same “language” when some people are being too technical and others are too business focused. Not understanding requirements of the business can occur when the architects are not the people who will use the architecture and even if they create a great architecture it might not fit the business. There can also be a lack of shared understanding if there is not a shared vision or shared vocabulary within the EA process as well people only having knowledge of how changes affects their own system but not how it will affect other systems. Another issue is the lack of experienced architects, because of the complexity of the EA field and the EA frameworks, the demand of competent people is quite high. The complexity of the systems in an organisation with the large amount of different applications and the dependencies to each other also makes it more
difficult to maintain consistency and traceability. The **rapidly changing conditions** in technology and business makes it unwise to perform a large architectural effort in one delivery because by the time it is finished the conditions have changed and the architecture is out-of-date. The **scoping of architectural descriptions** is also an issue because of the holistic approach of EA and insufficient scoping can lead to too broad of a scope or over-detailing. The **EA frameworks** are usually too complex but still not too abstract or not described enough in certain areas and the frameworks are not able to adapt to the concerns of the specific organisations. Part of the **knowledge management** problem is the lack of documentation which in turn leads to difficulties keeping track of the decisions being made and why they were made. There is also **insufficient tool support** for keeping track of the architectural goals and objectives among other things.[21]

Other challenges for enterprise architecture emerges during the development of the architecture capabilities, one of these is finding the balance between excessive control and insufficient impact. Applying too much control during the initial stages of the architecture implementation process is a common mistake or applying the same amount of control across all processes within the organisation. The organisation also needs to find the balance between reactive and proactive decisions as well as between having a strategic view and a tactical view. For the first dilemma the organisation need to have the right priorities and the right resources. While for the second one they need to determine for how long their plans should extend, in general an organisation starts with a tactical view which evolves over time towards a more strategic view. Another thing to take into consideration is if the architectural delivery should be holistic and serve the greater good which can be more financial and time consuming or if they should benefit a specific project.[4] A study [27] in which both architects and developers were interviewed found that both groups value this holistic view. The problem of over-architecting occurs when the architectural models tries to include too much detail on the lower levels where the circumstances changes so much and so often that it becomes too expensive to keep the models up-to-date. The architects need to know where to put the boundaries of the models in order to prevent creating unnecessary models, the developers preferred using no models over using bad or outdated models. The same study found that there were often a problem with communication between the roles, sometimes unbeknown to one of the parties. Possible reasons for this was the large distance between the groups as well as insufficient trust or knowledge of the work of the other group. There were also a lack of reusability when company bought or developed solution which was already available to them. A common reason was the lack of knowledge of already developed systems which was partly caused by not having a holistic view and not using the architecture well enough. Although, even when existing solutions were known the companies sometimes chose to order or develop something new anyway. There were a number of reason for this. It is often takes less time and effort to buy something brand new than to find a suitable solution and integrate it. Further more it is usually easier to create a new system than to adapt an existing one to the new requirements even if latter is cheaper. They often prioritised time to market over saving money.[27] Additional challenges for enterprise architecture models may occur when modelling large systems and the cost could be too high as the models increase in size as well as the cost for collecting the data required for those models[51]. Other issues enterprise architecture need to address, in order to be successful, are the companies' tendencies of spreading their resources too thin and their inability to prioritise[52].
There is also the question of how the organisation's enterprise architecture should be structured, centralised or more decentralised. The decision is greatly affected by how the organisation is structured in general but other organisational factors influence the decision as well. For example the goals and the level of maturity the organisation's EAM as well as corporate politics or other relation. A centralised structure is better suited for a more centralised organisation that have their IT at a single location and where the centralised enterprise architecture office who reports to the Chief Enterprise Architect (CEA) can provide architecture services. A decentralised structure fits organisations that have different business units which operate quite independently from each other and have their own architecture resources. Another structure is the Centre of Excellence structure (CoE) where the resources are organised based on their area of expertise and their services are shared with the rest of the organisation. There is also a hybrid structure which is a mixture of decentralised and CoE where some parts can be centralised or shared and other specific to a certain business organisation.

Centralise architecture groups have more influence in the organisation but lacks control over decisions regarding the information architecture. There are two main reasons for this, business managers can carry out projects without involving them and not following the architecture plan or they can decide not to invest in the architecture. Some believe the enterprise architecture should be driven by the business and not from the IT department since it is a business function and not purely an IT function even though IT is an important part of it. Nonetheless, the EA effort is driven IT department in many companies, partly because they usually initiate the effort and because the business does not have the knowledge carry it out.

![Figure 5: A decentralised enterprise architecture model][4].
Normally the Chief Enterprise Architect (CEA) talks to the Chief Information Officer (CIO) directly or to a manager within IT who is in charge of the IT strategy or architecture[24]. The CIO is in turn often close to senior management of the company and is in most companies the main stakeholder of the enterprise architecture, even though other roles should have some involvement in it[28]. A CIO of a company is responsible for any issue regarding the company's IT and to maintain the quality of the overall IT system[25]. Enterprise architecture can provide the stakeholders with decision support for those issues by helping them plan, design and communicate the issues[26]. Nonetheless, some EA processes fail partly because the top-level management, which include the CIO, does not understand or support the processes. Some of the reasons comes from the EA's side where they have not communicated or engaged enough with the business people and the knowledge about the EA process have therefore been lacking. These issues can be reduced by having the stakeholders participate and collaborate more, the CIO can be seen as the main stakeholder for EA because of its responsibility for the IT system as well as for business and IT alignment.[23] A method to increase collaboration, especially between independent organisations, is having an architecture forum where different architecture organisations within the business can come together to discuss and collaborate on common issues voluntarily, such as standards or standardisation[4].

4.2 Agile Architecture

Agile Enterprise Architecture Management (Agile EAM) is a way to mitigate some of the issues of the individual approaches by combining agile with EAM. The goal is to make the architects work in a more agile way with more streamlined processes with incremental deliveries in order to get faster feedback.[10][37] Another agile principle which can useful takes inspiration from Toyota's lean development and the principle of impediment reduction which is removing any obstacle of the team, similar to one of the responsibility of the scrum master. This addresses the issue agile can have when it comes to incorrect scope and management.[15] Other research have suggested that the use of agile methods are needed for reducing the effort of creating and using architectural knowledge[7] or to receive faster feedback by having incremental architecture deliveries[22].

It is however rare to find quantitative research examining the combination of agile and architecture, it is for the most part supported by the opinions of experts[10] or analysed by practitioners. These practitioners use their experiences to describe the topics and many of them agree that the two can coexist, but it differs on how and to what extent[5][8][9][10]. One of those practitioners is Scott Ambler who have written multiple articles about the subject and in one of his articles he lists what an effective enterprise architecture needs in an agile environment. It should be:[5]

1. Business driven, not driven by the IT department but IT is still important. It should be owned by the business which can be difficult since they are not always that knowledgeable of EA and might view it as an IT function.
2. Evolutionary, the architecture should be developed iterative and incrementally to better respond to feedback as well as steer it better.
3. Collaborative, the architects need to work closely with the users of the architecture, both on the IT and business side, maybe even work in the team.
4. **Focused on producing valuable artefacts**, the EA team should not focus on the artefacts they want create but on what the users want.

5. **An explicit part of development**, just as the architects need to be involved with the developers, the teams themselves should work closely with the enterprise architects. This will help them use the architecture more effectively and they can even assist with its design.[5]

In an article on his website this is expanded on with an agile approach to enterprise architecture. It argues that enterprise architecture is needed to make sure that applications do not affect other systems and that they are utilizing the existing infrastructure effectively. Otherwise it can be costly and there is a risk of having multiple products which are similar or duplicated even if individual projects might be successful it can be an issue on a larger scale. However, the enterprise architecture needs to be agile as well by having more streamlined activities to better support other parts of the organisation and work more according to their customers. This version of agile enterprise architecture incorporates practices, principles and values from agile, these include the following:[6]

1. **Focus on people, not technology or techniques**, it is the actual people who use the architecture and they are more important for their success than the tools or techniques they use. The architecture is useless unless someone can utilise it. The architects need to work closely with the customers and design together as well as be the individuals who other people go to get things done, the models also need to be given to the customers so they can be used.

2. **Keep it simple**, when creating models it not necessary for them to be absolutely perfect, it is better to focus on just making them “good enough”. It takes too long to try to achieve perfection that it will be out-of-date when it is completed, it is better to create an up-to-date model of the current situation.

3. **Work iteratively and incrementally**, to work iteratively the architects can detect early if the model is not suitable or if they are stuck. Working incrementally helps architects to create models that are just “good enough” by not delivering complete models. The customers can get the models quicker, give feedback and if something is missing they can continue working on them.

4. **Roll up your sleeves**, documentation and modelling should not be the architects main objectives. They should prioritise working with the architecture in the actual projects in order to detect earlier if the ideas work, improve the teams' understanding of the architecture and get feedback quicker.

5. **Look at the whole picture**, do not use just one type of model, look from different perspectives to improve their understanding of the architecture.[6]

They also list the potential issues with an agile approach to enterprise architecture:[6]

- There is no means of ensuring that the approach is followed, having enterprise architects being a part of the teams mitigates this somewhat.
- The approach relies on the responsibility of the people in the organisation.
- The approach needs to continuously work towards becoming simpler.
- The modelling and documentation needs to comply with an agile approach.[6]

Other practitioners agree with Ambler that agile and enterprise architecture can coexist but that the organisations need to align traditional top-down enterprise architecture with bottom-up architecture which is more towards agile. There can be
some friction between the two approaches if people from either side do not see the
good architectural practices brings flexibility and stability to an
organisation which helps it change as well as innovate. To be agile means more than just having iterative, responsive and interactive processes, the organisation also needs to develop systems which can adapt to changing requirements and conditions. Enterprise architecture helps with that by identifying what the business needs and how technology or business changes could improve the organisation. There are two important capabilities of an organisation’s architecture to consider when discussing agility, the execution system which concerns maintaining the current business and the innovation system which is the ability to change or innovate. There are also five aspects of agility which architecture can benefit from:[18]

- **Making changes**, able to easily make changes though modularity or layering as well as having clear interfaces and few dependencies between systems.
- **Deploying changes**, to deploy the changes quickly while retaining quality and manageability. Changes that need a lot of management effort is not very agile.
- **Dealing with the effects of changes**, handling the effects of the changes and minimise the impact if something goes wrong, which also increase robustness.
- **Integrating**, having systems that are integrated with its environment which makes it easier to implement changes or use it in other environments.
- **Decoupling**, having less dependencies will make it easier and quicker to make changes, develop new solutions and having independent components.

Another study have looked into the integration of TOGAF and Scrum where they conducted semi-structured interviews with experts from a number of different companies. Similarly to the practitioners they found that there are some disagreements between the two approaches when thinking about the strategy while using Scrum at the same time. The teams usually lose track of the overall plan and tend to only focus on the individual projects rather than organisation's goals. On the other side, the EA sometimes suffers from Ivory tower which create a gap between the architects and the teams. Problems are therefore discovered late and the teams have to wait for architectural decisions before they can implement solutions which makes them view the architects as a hindrance that creates unnecessary documentation. The approach proposed focused on implementing changes faster rather than spending time on documentation of the current architecture. The architects should also collaborate more closely with the developer teams and to prevent ivory tower the teams should be able to define their own solutions but they should still take the IT landscape into
consideration. The approach uses the multiple levelled design of TOGAF and the goal is that the architecture should enable agility across all levels. The TOGAF ADM and architecture changes should be done in Scrum projects with little to no modification of the Scrum methods. Therefore the projects will have their own architecture versions of Product Owners and Scrum masters with the architects acting as the teams.[17]

The most relevant work is a master thesis[48] that used a questionnaire to investigate what the current challenges for EA development are and how it might benefit from agile approaches. The research found that the most common challenges were communicating with stakeholders, unclear roles and responsibility of the stakeholders, the scoping of EA development and EA products not being useful. The most common agile approaches used in the EA development were incorporating new requirements during the development, involvement from the business in the EA development and having an incremental approach. The study found the correlation between the use of three specific agile practices and the absence of certain EA development challenges. One of those practices was the ability to deal with changing requirements, which had a positive influence on both stakeholder communication and rapidly changing conditions. Another method was the use of reflection which addressed issues regarding outdated models and not having a shared understanding. The last correlating practice was focusing on the essential, which was a rare occurrence in the survey, and it had an impact on several different EA challenges. Those challenges were rapidly changing conditions, EA governance, architectural scope, understanding requirements and knowledge documentation & presentation. The research also found that there were no significant correlation between any of challenges and working incremental or iterative, both of which are prominent agile practices.[48]

Another relevant work is a study done by the Technical University in Münich which uses a survey to identify which agile principles have been incorporated into the EA management and which EAM challenges they are meant counteract. The paper provides an empirical foundation for the application of the agile principles but is purely quantitative and do not cover the implementation of these principles or the impact of them. Nor does it concern the correlation between the challenges within EAM and the usage of these agile practices or have any qualitative data. The authors found that both incremental and iterative approaches were popular and that over 90% of the participants operates cross-functional, while prioritising time over quality was not that popular (26% agreed, 40% disagreed). The top 15 practices used were, in order (and with how many percent agreed they used it): 1) Operates cross-functional (95%), 2) Incremental (89%), 3) Iterative (81%), 4) Perform tasks in a self-organised manner (77%), 5) Specialised to perform various tasks (76%), 6) Incorporation of reflections & retrospectives (73%), 7) EAM team incorporates feedback (72%), 8) Leader acts as servant for the team (72%), 9) Leader fosters team's self-organization (68%), 10) As simple and accessible as possible (67%), 11) Usable for stakeholders (67%), 12) Common language (66%), 13) Foster learning by experiments (65%), 14) Early delivery (65%), 15) Members know their colleagues' duties (64%).[11]
5 Results

A total of 19 employees were interviewed with an average time of 50 minutes, every interview was done in person and one on one (with one exception where a co-worker of the interviewee was present but did not talk). The interviewees had one of four different roles: 4 were enterprise architects, 9 were business architects, 4 were solution architects and 2 employees were portfolio managers. The portfolio managers' way of working differs from the architects to such a degree that they will be excluded from the method part of the results. However, they will still be included in the challenge part since they have experienced many of the same challenges and problems as the rest of the interviewees.

![Role distribution of the interviewees.](image)

Figure 6: Role distribution of the interviewees.

The interviewees were asked about their way of working, which agile methods they use and the different challenges or problems they experience in their work. The result will be divided into four parts, Interview questions, Methods, Challenges and Agile architecture. The first part will cover the questions that were drawn from the Literature review in Chapter 4. The following two parts will present the common topics that were mention in the majority of the interviews, then topics that were not as popular but still common. At the end of each topic a table will statistically present how many of the interviewees used a specific method or experienced a particular challenge. The forth part will be a summary of the interviewees thoughts on agile, architecture or the combination of the two concepts.

5.1 Interview questions

The Literature review were used to find possible agile methods that could be used by the architects in their EAM as well as challenges they may face which can be asked during the interviews. The questions will still be open-ended to avoid influencing the
interviewees in any way. Some of these questions will not be included in the result because they are not mentioned enough in the interviews to be recognised as a common topic or not mentioned at all. Many of the methods and challenges could be found in multiple sources but only one source will be mentioned. From the research, the following examples of possible methods were assembled:

- From [5]: working iterative and incremental in order to receive faster feedback as well as working closely with the business.
- From [6]: modelling or designing together with the business or customers as well as the valuation of time over quality or completion (time to market).
- From [8]: having smaller or earlier architectural deliveries instead of delivering everything at once.
- From [18]: being able to make changes easily and having few dependencies between systems.
- From [17]: working together with the teams.
- From [11]: being cross-functional, perform changes based on feedback, the use of reflection and having knowledge of what other people are doing.
- From [4]: having regular architectural forums.

From the research, the following examples of possible challenges were gathered:

- From [18]: ivory tower syndrome.
- From [15]: aligning with stakeholders' interests, adapting to change, ensuring early and periodically EA deliveries.
- From [12]: ad-hoc demands, unclear goals, requirement or conditions changing too fast and too much IT focus.
- From [6]: out of date models and troubles reaching the teams.
- From [21]: lack of authority, being bypassed by the management, not enough stakeholder involvement, lack of communication or coordination between layers, not understanding business requirements, no knowledge of how changes affect other systems, no shared architectural vision or vocabulary, complex environment, insufficient tool support and lack of resources.
- From [27]: applying too much control, over-architecting and lack of reusability, models not used by customers/users.
- From [4]: having centralised or decentralised enterprise architecture.

Some challenges were not found in the literature review but during the interviews, these were: lack of EA contact, no CIO, EA office at the IT department, too many projects at the same time, stovepipes and having different methods for the same tasks.

5.2 Methods

There were some topics that were talked about in the majority of the interviews which will be outlined in separate sections followed by a section with topics that where still common but not to the same extent as the previous ones. At the end of each section a table will present the amount responses for the methods for each role in four categories. ● will represent that the interviewees use the method, ◆ means that the interviewees use the method to some degree, ○ indicates that that the interviewees do not use the method and - shows that the method was not mentioned during those interviews. The total will show the percentage of responses for all roles by merging ● with ◆ and ○ with -.
5.2.1 Model creation

Even though the different architects have separate assignments and tasks they work in similar way for creating models, which seems inspired by agile. Most people work iterative and incremental by having starting small and expanding on it. It differed however how they initially start on the models. Half of the enterprise architects said that they starts from the top and break it down by for example starting to look at the application landscape as a whole then break away an area of it such as Enterprise Resource Planning (ERP) and form an overview of that area. Then they break ERP down to different parts and look into those parts further e.g. administrative ERP and operative ERP. Those part are then broken down to even smaller parts like the HR and Finance for which they investigate how the current situation is and how it will look like in the future, from both an industrial perspective and from the commercial perspective. One of the enterprise architects and the majority of business architects had an almost reverse approach where they first identified the most important or relevant areas and starts from there. Compiling information or guidelines within those areas then successively adding things to the model by either inspecting the same area further or by moving on to the next area. Most of the solution architects had a similar approach where they start at a relevant area and iterates within it. One difference is that they keep it on a high level and do not go into too much detail so it is easier for the projects to stay within in the models. They also felt it is unnecessary to add too many details about areas they do not know everything about because they will probably end up having to change them anyway.

Another common method is perform changes based on feedback. All of the enterprise architects as well as the majority of the business architects and solution architects said they worked this way. Most of the them said they do it through having early or smaller deliveries to the business, users or stakeholders who can provide feedback on it which the architects can use to improve their next delivery. Some only had these deliveries during the initial stages of an assignment while others had continuous demonstrations through out the projects. Many of the models are delivered in smaller batches to avoid having a large delivery at the end of a project which could turn out to be wrong or outdated.

Related to delivering in smaller parts is how much they value time to market over quality or how much they prioritise delivering something quickly over delivering it more complete. Their opinions of this is pretty split even within the roles. Half of the enterprise architects said they work more towards the former approach where they deliver a model more quickly and improve it afterwards or they produce an early foundation which they can change or build upon later. Whereas one of their colleague generally tried to finish a part so it can be presented and they can receive feedback on it. While the last one was more in between where the deliveries rarely are extremely complete but they should be sufficiently complete (depending on what they are supposed to be used for) and they should not lead to wrong decisions. The approach among the business architects differs somewhat as well. One said they occasionally deliver models before they are complete. Someone else said that if there is a deadline they do not want to miss they can deliver with some limitations but since their systems are a bit disconnected they can usually correct the limitations afterwards with change requests. Other business architects who works with machine deliveries do not
have the same priority because those project usually follows some kind of Waterfall model where the specification and deadlines are mostly defined beforehand. One business architect's opinion was that you should not be too fast, you can of course experiment and have prototypes but when the actual development start you should be pretty certain that it is correct otherwise it will be expensive. The solution architects were more in agreement that they can deliver without necessarily being completely done or having every detail but the customers need to get their value and it is not accepted that it will cause any stop in production.

In addition to the deliveries and feedback over half of the architects are working together with the business or their customers, often for creating models. Usually through workshops or meetings with people from the business in order to learn how the current situation looks like, what their needs are and what plans they have for the future. During those collaborations the majority also create the actual models and designs the plans together with the business and not just gathering information.

Table 3: Methods: Model creation

<table>
<thead>
<tr>
<th>Methods</th>
<th>EA (4 in total)</th>
<th>BA (9 in total)</th>
<th>SA (4 in total)</th>
<th>Total (17 in total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model creation</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● &amp; ○ ○ &amp; -</td>
</tr>
<tr>
<td>Iterative</td>
<td>4 0 0 0</td>
<td>6 0 0 3</td>
<td>2 0 0 2</td>
<td>12 (71%) 5 (29%)</td>
</tr>
<tr>
<td>Incremental</td>
<td>2 1 0 1</td>
<td>5 1 0 3</td>
<td>1 0 0 3</td>
<td>10 (59%) 7 (41%)</td>
</tr>
<tr>
<td>Starting small</td>
<td>4 0 0 0</td>
<td>5 0 0 4</td>
<td>0 0 1 3</td>
<td>9 (53%) 8 (47%)</td>
</tr>
<tr>
<td>Early/smaller deliveries</td>
<td>2 0 0 2</td>
<td>5 0 0 4</td>
<td>2 0 0 2</td>
<td>9 (53%) 8 (47%)</td>
</tr>
<tr>
<td>Changes from feedback</td>
<td>4 0 0 0</td>
<td>6 1 0 2</td>
<td>3 0 0 1</td>
<td>14 (82%) 3 (18%)</td>
</tr>
<tr>
<td>Work with the business</td>
<td>0 1 1 2</td>
<td>6 0 0 3</td>
<td>3 0 0 1</td>
<td>10 (59%) 7 (41%)</td>
</tr>
<tr>
<td>Create with the business</td>
<td>0 0 0 4</td>
<td>6 0 0 3</td>
<td>2 1 0 1</td>
<td>9 (53%) 8 (47%)</td>
</tr>
<tr>
<td>Prioritise time to market</td>
<td>2 1 1 0</td>
<td>1 1 3 4</td>
<td>1 1 0 2</td>
<td>7 (41%) 10 (59%)</td>
</tr>
</tbody>
</table>

5.2.2 Standardised framework

None of the enterprise architects said they follow any standardised framework such as TOGAF. One of them thought that there is no real need to follow any framework while another though it would be better if they did, especially when it comes to model usage and it would be beneficial when hiring new employees. However, both agreed that it is hard to find a framework that would suit the company and even if they chose one it would still need to be modified to fit the company better.

Among the business architects Astrakan were the most popular to follow for process modelling and was even used by one of the solution architects. However, one business architect had chosen not to use it because they are collaborating with a company from another country and they need a standard they can easily agree on. ArchiMate together with TOGAF was chosen which are closer to being international standards than Astrakan.
Half of the solution architects said that they have just recently started using SAFe and they have not gotten that far with it yet but they are positive towards using it more. The others followed a standardised method but it is either not that well defined or the conditions are rarely suitable to follow it properly. Some of them have also received a collection of guidelines from the enterprise architects which describes how the solutions should be designed and rules regarding security, information handling etcetera as a support for the architects. These are meant to help the developers to know what to keep within without restricting them too much so they are still able to be independent. It also helps the teams know what is available to them, defines what an application should be and what functionalities should be in it so they do not develop something that already exists in another system. It is the solution architects responsibility to make sure the developers follow these guidelines but it is still quite new and has just recently been sent out to them so not everyone have received it yet.

Table 4: Methods: Standardised frameworks

<table>
<thead>
<tr>
<th>Methods</th>
<th>EA (4 in total)</th>
<th>BA (9 in total)</th>
<th>SA (4 in total)</th>
<th>Total (17 in total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard frameworks</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● &amp; ○ ○ &amp; -</td>
</tr>
<tr>
<td>Follows a framework</td>
<td>0 0 3 1</td>
<td>4 1 0 4</td>
<td>2 2 0 0</td>
<td>9 (53%) 8 (47%)</td>
</tr>
<tr>
<td>Uses Astrakan</td>
<td>0 0 0 4</td>
<td>3 0 1 5</td>
<td>1 0 0 3</td>
<td>4 (24%) 13 (76%)</td>
</tr>
<tr>
<td>Uses SAFe</td>
<td>0 0 0 4</td>
<td>0 1 0 8</td>
<td>2 0 0 2</td>
<td>3 (18%) 14 (72%)</td>
</tr>
</tbody>
</table>

5.2.3 Regular forums

A common occurrence among the architects is participating in regular forums with other architects, generally every week, where they can bring up to various architectural questions or issues that they can discuss and solve together. Most prevalent is the forum between the enterprise architects and lead solution architects from the different business areas. The leads can bring up any question or concern that affects their department and the other leads together with the enterprise architects can help them. The forum can also be used to raise issues that affects multiple departments which they can be solved together and come up with a common solution instead of the individual departments developing separate solutions. The developer teams can raise their questions to their respective solution architects who relays them along with their own questions to the lead solution architects who can ask them during the next forum. This was mentioned by all the enterprise architects and all of the solutions architects, both lead and non-lead. One of the non-lead also mentioned being part of a forum with the enterprise architects but is not any longer since they rotate members in order to give competence to more and new people. The other non-lead mentioned an internal forum for the solution architects within the same business area together with their lead solution architects.

The situation is the opposite for the business architects, none of them have any forum with the enterprise architects and only one has any regular forum involving solution architects or other business architects. In this weekly forum people from the IT department can come for advice or ask questions to business architects, solution architects and representatives from the business capabilities. According to the
architect this has been shown to be very successful and discussions are taking place at neighbouring sections about implementing similar forums. Some of the interviewed business and solution architects have showed interest in having a forum with representatives from different business organisations in order to weaken stovepipes or improve reusability and coordination between them. It can also be useful for discussing solutions for functionalities that are similar for multiple organisations, such as invoices which do not differ that much between the different businesses and they can use the same system for it. Half of the enterprise architects mentioned that they host a larger forum a couple of times a year where they invite architects, developers and people from the business to talk about what they are currently working with but also to discuss interesting trends or topics with each other.

Most of the enterprise architects interviewed and some of the business architects were positive towards having a forum between them, similar to the one involving the lead solution architects. There used to exist a similar forum but not any longer. According to one enterprise architect it can easily be started again but it requires that the businesses are able to identify business problems (both long term and short term) and having clear business plans. Some business organisations that are almost there where there will not be such big problems compared to those organisations that are not as transparent.

Table 5: Methods: Regular forums

<table>
<thead>
<tr>
<th>Methods</th>
<th>EA (4 in total)</th>
<th>BA (9 in total)</th>
<th>SA (4 in total)</th>
<th>Total (17 in total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular forums</td>
<td>● ○ ○</td>
<td>● ○ ○</td>
<td>● ○ ○</td>
<td>● &amp; ○ ○</td>
</tr>
<tr>
<td>Regular forums with EA</td>
<td>4 0 0 0</td>
<td>2 2 0 0</td>
<td>8 (47%)</td>
<td>9 (53%)</td>
</tr>
<tr>
<td>Regular forums with BA</td>
<td>0 2 2 0</td>
<td>0 0 0 4</td>
<td>3 (18%)</td>
<td>14 (82%)</td>
</tr>
<tr>
<td>Regular forums with SA</td>
<td>4 0 0 0</td>
<td>3 0 0 1</td>
<td>8 (47%)</td>
<td>9 (53%)</td>
</tr>
<tr>
<td>Want EA and BA forum</td>
<td>3 1 0 0</td>
<td>0 0 0 4</td>
<td>8 (47%)</td>
<td>9 (53%)</td>
</tr>
<tr>
<td>Want organisation forum</td>
<td>0 0 0 4</td>
<td>2 0 0 7</td>
<td>4 (24%)</td>
<td>13 (76%)</td>
</tr>
</tbody>
</table>

5.2.4 Less common methods

Some methods were only mentioned in some of the interviews but not to the same degree as the previous topics.

Reflections

The use of reflections were only mentioned by a few of the business and solution architects but they are not always utilised. It often up to the people involved if it should be done or if they should be used when starting a new project. Almost all of them want it to be improved in some way, either by having a better system support for it or change the process to incorporate it better.

Cross-functional

Cross-functionality is also only mentioned by two of the architect roles, enterprise and business. In the former group most of them have worked for many years within
different fields and have therefore very broad knowledge. They do not really have a niche area, they are more comb shaped with multiple tops and they usually work in smaller groups to help each other which makes it easy to handle distinct areas. While a business architect describes their situation as more of a T shape where they still have broad knowledge but have people who are more experienced in certain areas who can take on those tasks and if there is a lack in a certain area they try to find someone who can fill that gap. As long as they work together they can divide the tasks among themselves easily, it is only when people try to work separately there might be some issues.

<table>
<thead>
<tr>
<th>Methods</th>
<th>EA (4 in total)</th>
<th>BA (9 in total)</th>
<th>SA (4 in total)</th>
<th>Total (17 in total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less common</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● &amp; ○ ○ &amp; -</td>
</tr>
<tr>
<td>Uses reflection</td>
<td>0 0 0 4</td>
<td>2 1 0 6</td>
<td>2 1 0 1</td>
<td>6 (35%) 11 (65%)</td>
</tr>
<tr>
<td>Needs better reflections</td>
<td>0 0 0 4</td>
<td>3 0 0 6</td>
<td>2 0 0 2</td>
<td>5 (29%) 12 (71%)</td>
</tr>
<tr>
<td>Cross-functional</td>
<td>3 0 0 1</td>
<td>1 0 0 8</td>
<td>0 0 0 4</td>
<td>4 (24%) 13 (76%)</td>
</tr>
</tbody>
</table>

5.3 Challenges

There were some topics that were talked about in the majority of the interviews which will be outlined in separate sections followed by a section with topics that were still common but not to the same extent as the previous ones. At the end of each section a table will present the amount responses for the methods for each architect role in those groups in four categories. ● will represent that the interviewees have experienced the challenge, ○ means that the interviewees have experienced the challenge to some degree, ○ indicates that that the interviewees have not experienced the challenge and - shows that the challenge was not mentioned during those interviews. The total will show the percentage of responses for all roles by merging ● with ○ and ○ with -, the responses from the Portfolio managers will be included in the total but will not have a separate group in the table.

5.3.1 Location of the Enterprise Architecture office

One of the most talked about subject were the location of the enterprise architecture office. It is currently located at the IT department which is itself at separate from the rest of the company and in a different building. Almost all enterprise and business architects think that it should not be located in the IT department and should be more towards the rest of the business since enterprise architecture is not an IT only function but for the company as a whole. IT is still important and some argues that if the enterprise architecture office moves closer to the business they will risk losing contact with the IT department. Many of them think that the EA office should be more centralised, perhaps under a CIO and closer to the CEO, and the rest of the top management. This would give them more say in things which a few of the architects think they do not have at the moment since they are on such a low level in the organisation and it is what other companies have done with sucess. The company does not have a CIO at the moment, the closest they have is the head of IT or the CDO
(Chief Digital Officer) which are not entirely the same thing as a CIO. Nonetheless, the EA office is not directly under the head of IT and they are a couple of levels below which could affect how much say they have in things. They used to have a similar but centralised functionality but it did not really take off and ended up doing mostly selective measures. It got a lot of criticism for not being sufficiently connected to the business and for not working on issues the business wanted. It was later removed and recreated at the IT department as the enterprise architecture office but some of its old responsibilities were not carried over and it got more IT priorities.

Table 7: Challenges: The location of the Enterprise Architecture office

<table>
<thead>
<tr>
<th>Challenges</th>
<th>EA (4 in total)</th>
<th>BA (9 in total)</th>
<th>SA (4 in total)</th>
<th>Total, with PMs (19 in total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EA office location</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
<tr>
<td>Not at the IT department</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Should be under a CIO</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

5.3.2 Contact with the Enterprise Architecture office

Almost all of the business architects felt they had too little contact with the enterprise architecture office, some even said it is non-existent and that they have co-workers who do not even know that there is an EA office at the company. Some are also not sure of what the EA office does, what its deliveries are or even what its purpose is. This was news for one of the enterprise architects who was not aware that the business architects wanted more contact with them. The solution architects do not share their business counterparts, all but one think they have a good relationship with the enterprise architects. The last one thinks they have too little contact with them and only have it through the lead solution architects but the architect also thinks that they should not need to meet the EA office if they keep within the guidelines set by them.

Almost all of the enterprise architects thought that they have issues reaching all they way down to the teams or to the business. Two of them feel it is hard to get insight into what the teams are working on without being a part of the teams and they do not have the time to be involved with all of the teams. One of them also felt they have difficulties reaching all the way to the teams with their deliveries and to know if their guidelines are being followed but they have the lead solution architects to help them with this. The third architect experienced similar issues but with the business where they have difficulties communicating the models to them and that some follow the models while others do not.

A couple of years ago it was mandatory for projects at the IT department to meet with the enterprise architects in order to go to production to make sure the solution would fit the rest of the company’s IT landscape. Those meetings were a good way for them to meet each other, have a dialogue and for the teams to ask for advice. These meetings are not mandatory any longer and now it is up to the teams themselves to booked a meeting if they need advice about the design of their solutions so they are not utilise as much. Many architects think those meetings are beneficial and are underutilised and one business architect sometimes insists the teams should book those meetings because it is such a good practice. The collection of guidelines
described in 5.2.2 mitigates some of those losses but it puts more responsibility on the solution architects since they have to make sure the team follows the guidelines. Some of the solution architects have also said that they lack the resources to follow them properly because the people with the competence to develop according to them are too busy with other things and they have to do it in another way if they want it developed in time.

A popular opinion is that the enterprise architecture office suffers from “ivory tower” which is supported by half of the enterprise and business architects as well as both of the portfolio managers. Many believe that they are too far away from the business, they lack knowledge of it and does not how it works. Some are not as certain and think that there are only warnings of ivory tower or that they are just too isolated.

<table>
<thead>
<tr>
<th>Challenges</th>
<th>EA (4 in total)</th>
<th>BA (9 in total)</th>
<th>SA (4 in total)</th>
<th>Total, with PMs (19 in total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact with EA</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
</tr>
<tr>
<td>Lack of EA contact</td>
<td>0 0 0 4</td>
<td>7 1 0 1</td>
<td>1 0 3 0</td>
<td>9 (47%) 10 (53%)</td>
</tr>
<tr>
<td>Unclear what EA does</td>
<td>0 0 0 4</td>
<td>2 2 0 5</td>
<td>0 0 0 4</td>
<td>5 (26%) 14 (74%)</td>
</tr>
<tr>
<td>Issues reaching down</td>
<td>3 0 0 1</td>
<td>0 0 0 9</td>
<td>0 0 0 4</td>
<td>3 (16%) 16 (84%)</td>
</tr>
<tr>
<td>EA meetings not utilised</td>
<td>2 0 0 2</td>
<td>2 0 0 7</td>
<td>1 1 0 2</td>
<td>6 (32%) 13 (68%)</td>
</tr>
<tr>
<td>Guideline collection mitigates loss of meeting</td>
<td>2 0 0 2</td>
<td>0 0 0 9</td>
<td>1 2 1 0</td>
<td>4 (21%) 15 (79%)</td>
</tr>
<tr>
<td>EA ivory tower</td>
<td>2 1 1 0</td>
<td>4 2 0 3</td>
<td>0 0 0 4</td>
<td>11 (58%) 8 (42%)</td>
</tr>
</tbody>
</table>

5.3.3 Resources

Almost everyone expressed that there is a lack of resources, the only exception were the enterprise architects who think they might only need a few more if any and some of the business architects where it depended on the task. Almost half of them think they have too many active projects at the same time that they have to switch between which wastes time and they can not focus on a specific assignment. Both a solution and an enterprise architect have said that this is one of the reasons to why it can take a couple of months before they can produce a first delivery instead of just a couple of weeks if they were able to focus on a single project. If the delivery turns out to be incorrect and they need to change or redo it they might have to spend half a year to deliver a correct prototype. A couple of the interviewees have also said that they can only focus on one area at the time because they are too few and it is the same people who work in all areas. For example a team that works both with production and development, if there is a production issue the development is put on hold until that issue is solved. Another interviewee explained a similar situation where change requests and pre-studies are handled by the same people so when a project is finished it takes a while before the next can start. Many have said the pressure is high at the moment which is partly because the company is doing well and they are also going through a digitalisation, both of which leads to business organisations investing more into IT. Some business architects have conveyed a need for better prioritisation of
projects by not starting new projects if the resources or competences are too busy. When you find that you do not have enough resource to fulfil the requirements for a new project you put it on hold until another project is finished and the resources are free. This would reduce the pressure on the employees and reduce the risk of them getting burned out. Only adding more people will not always be enough to solve the issue because they might just get more assignments since they have more resources now and the issue will remain. It also takes time and effort to instruct new employees. Most of the enterprise architects prioritises their own tasks with one saying they prioritises together with their bosses while none of the solution architects mention it. For the business architects only a few of them prioritises their own assignments because the business do that for them, one solution architect said their lead solution architect does it and some business architects act as their own Product Owners for some assignments.

Table 9: Challenges: Resources

<table>
<thead>
<tr>
<th>Challenges</th>
<th>EA (4 in total)</th>
<th>BA (9 in total)</th>
<th>SA (4 in total)</th>
<th>Total, with PMs (19 in total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● ○ ○ ○ + -</td>
</tr>
<tr>
<td>Lack of resources</td>
<td>1 0 3 0</td>
<td>5 2 1 1</td>
<td>4 0 0 0</td>
<td>14 (74%) 5 (26%)</td>
</tr>
<tr>
<td>Too many projects</td>
<td>1 1 0 2</td>
<td>4 0 0 5</td>
<td>1 0 1 2</td>
<td>9 (47%) 10 (53%)</td>
</tr>
<tr>
<td>Lack of prioritisation</td>
<td>0 0 1 3</td>
<td>3 0 0 6</td>
<td>0 0 0 4</td>
<td>3 (16%) 16 (84%)</td>
</tr>
<tr>
<td>Prioritises own tasks</td>
<td>2 1 0 1</td>
<td>2 2 1 4</td>
<td>0 0 2 2</td>
<td>7 (37%) 12 (63%)</td>
</tr>
</tbody>
</table>

5.3.4 Coordination

Many have brought up that there is a problem with lack of coordination between different parties at the company. Primarily between different business organisations which the majority of both business and solution architects think need improvements as well as a few of the enterprise architects. Most of them think they need to have better discussions with each other, which could be possible through forums that were described in 5.2.3 or by inviting people from other businesses to the meetings that could be relevant to them. A few of the interviewees have said that there is a dependency on the individuals and knowing the right people which makes it harder for new employees. One architect said that the only way for them know if changes in other organisations will affect them is to know someone from that organisation who can tell them that the change is on its way so they can prepare for it. The company have strong informal networks and if you are a part of them you know who knows what which makes it faster to solve and discuss things together. However, it becomes more difficult if you are not part of those networks or you have to talk to someone who is not. It is both a strength and a weakness, it makes for a smooth way of working that does not require a lot of bureaucratic decisions but it creates a dependency on the individuals. Many architects believes the company have organisational “silos” or “stovepipes” in some places. The organisations are very focused on themselves without too much involvement from people from other organisations and they keep themselves within their own silo or stovepipe. When the organisations solves their own problems without knowing what the other organisations are doing or have done, it results in a lot of solutions and systems with similar functionalities or capabilities.
All of the solution architects and the majority of the enterprise and business architects experience this issue. It can lead to a more complex IT landscape as well as more stovepipes which could result in increased costs for the IT department and make it harder to manage everything. They are working on counteracting it by having capability models to see if a capability already exists in a system or by having more discussions between organisations. One of the solution architects works with arranging meetings with the different departments that are doing similar tasks, such as assembling different component, in order to coordinate them. However, the departments very rarely meet without involvement from the architects.

According to some of the architects the organisations are sometimes unwilling to use an existing solution even if they know about it, partly because they think they are unique and therefore need a unique solution. One solution architect explained that at the start of the projects the business wants their own solution and does not want to reuse someone else's because they think they are unique. However, after they show the business what is available to them they usually comply but with some small differences which can be added to the existing system. A solution architect explained that the business sometimes does not want to use someone else's solution because they want to have control over the system and want to be able to do changes to it without having to involve another department or organisation. When it comes to reusability one enterprise architect explained that it is sometimes an intentional strategy by the company to buy solutions for processes that could be considered “standard processes” which does not differ that much compared to other companies. With this strategy it is a lot easier to determine the cost of things in contrast to developing your own where you do not always know how much it will cost.

Another reason to why they want different systems is that the different business organisations have different ways of working for the same tasks. These distinct processes for doing the same tasks could be a consequence of the organisations being too individual or because the conditions are different. One example of this was when the company tried to create a common process for goods reception which they thought would not differ too much between the different type of goods. They discovered that the 12 different production units worked in 12 separate ways but they agreed on doing it the same way and defined it using parts from the different methods. When they did a follow up six months later, they discovered that they had seven or eight different methods. It was still progress but to be able reuse effortlessly the business need to agree on a common method. Otherwise they would always get new requirements which would be hard to adapt to if they have several different ways of working. Interviewees from three of the roles have said that to prevent the department from choosing their own system for the same process, a decision needs to come the top about what system is going to be used for that process. Some of the architects are already working with trying to standardise the way of working or coordinate the businesses in order to have the same solution for the same needs. All of the enterprise architects think there are problems with the coordination between the developer teams. There are two main issues, describe by half of the architects each, the first one is the coordination of the IT landscape which is related to having multiple similar solutions. This is partly caused by the teams not solving things the same way and choosing different technical components in their solutions which also makes it harder for people to move between the teams. Another reason is the impression that the product owners goes directly to the development team with a product instead of
checking with the EA office first who can assign it to a more appropriate team. The current process might be quicker but it affects the IT landscape and, in the long run, slow down change implementation. The other issue is synchronising the teams and the sprints, especially for larger projects which involves multiple teams. They need to communicate better with each other to synchronise their sprints and be able to make demands before the sprint planning. When the teams are misaligned it is difficult for them to demand something if the other team is in the middle of a sprint, then the demands might have to wait until the next sprint. They also need to align their deliveries with the other teams in order to fit them in the same solution and to keep within the guidelines set by the architects.

Table 10: Challenges: Coordination

<table>
<thead>
<tr>
<th>Challenges</th>
<th>EA</th>
<th>BA</th>
<th>SA</th>
<th>Total, with PMs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(4 in total)</td>
<td>(9 in total)</td>
<td>(4 in total)</td>
<td>(19 in total)</td>
</tr>
<tr>
<td>Coordination</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● + ○ ○ + -</td>
</tr>
<tr>
<td>Coordinating businesses</td>
<td>1 1 0 2</td>
<td>5 0 0 4</td>
<td>3 0 1 0</td>
<td>10 (53%)  9 (47%)</td>
</tr>
<tr>
<td>Individual dependency</td>
<td>0 1 0 3</td>
<td>2 0 0 7</td>
<td>0 0 0 4</td>
<td>3 (16%)  16 (84%)</td>
</tr>
<tr>
<td>Stovepipes</td>
<td>2 0 0 2</td>
<td>3 0 0 6</td>
<td>2 0 0 2</td>
<td>7 (37%)  12 (63%)</td>
</tr>
<tr>
<td>Similar solutions/system</td>
<td>3 0 0 1</td>
<td>4 0 0 5</td>
<td>4 0 0 0</td>
<td>12 (63%)  7 (37%)</td>
</tr>
<tr>
<td>Business not reusing</td>
<td>2 0 0 2</td>
<td>4 0 0 5</td>
<td>3 1 0 0</td>
<td>10 (53%)  9 (47%)</td>
</tr>
<tr>
<td>“Unique” businesses</td>
<td>1 0 0 3</td>
<td>3 0 0 6</td>
<td>0 1 0 3</td>
<td>5 (26%)  14 (74%)</td>
</tr>
<tr>
<td>Different methods</td>
<td>0 0 0 4</td>
<td>3 0 0 6</td>
<td>1 0 0 3</td>
<td>4 (21%)  15 (79%)</td>
</tr>
<tr>
<td>Change from the top</td>
<td>0 0 0 4</td>
<td>1 1 0 7</td>
<td>1 0 0 3</td>
<td>4 (21%)  15 (79%)</td>
</tr>
<tr>
<td>Coordinating teams</td>
<td>4 0 0 0</td>
<td>0 0 0 9</td>
<td>0 0 0 4</td>
<td>4 (21%)  15 (79%)</td>
</tr>
</tbody>
</table>

5.3.5 Less common challenges

Some challenges were only mentioned in some of the interviews but not to the same degree as the previous topics.

“Ad-hoc” EA assignments

A few of the business architects think that the enterprise architects are not working with enough “enterprise architecture work” and are working too much with “urgent” investigative tasks for the IT department instead of strategic or long-term architectural tasks for the business. The enterprise architects on the other hand have the responsibility of supporting the whole company and they see these tasks as a part of that responsibility. They do not see it as a problem with doing these urgent tasks even if it means they have to put their long-term strategic plans temporary on-hold which might mean a few weeks delay, it is more important to solve these urgent matters. They should be able to work with long-term strategic goals but at the same time be able to make short-term effects and it is also a way for them to know what is going on at the business.
Motivating the models
Some of the business architects have experienced difficulties motivating their models. It is often that the business does not see the value of them or why they are needed and sometimes only want a solution without thinking about how it will fit the models or the processes. There is also similar issues motivating to the higher levels and one of the architects had particular difficulties with convincing the middle management the need for change. The top management want change based on KPIs (Key Performance Indicators) and the workers at the bottom layer wants the changes because they are experiencing problems that need to be solved. The middle management are not as committed which breaks the flow of change and nothing gets done. They have the operative responsibility and it is their resources that need to carry things out which is why the top management needs to push down on them to get it done.

Improved tools
Half of the enterprise architects and one business architect have expressed the need for improved tools. Both enterprise architects wanted a better tool for visualising connections between applications in order to reduce the amount of manual work. One of them wanted to improve the ability to identify what applications would be affected by the discontinuation of a specific supplier while the other wanted improvements for identifying which applications supports a certain capability and better visualisations of models to easier show other employees. One of them also said that they do not really have a tool for enterprise architecture strategies, they have one for modelling but not for strategies. They would also want a tool to help the developer teams know what the other teams are working on without having to go through the lead architecture forum. The other wanted a tool for producing reports from databases and repositories which are not that well put together at the moment. The business architect think their modelling tool needs to be improved when it comes to searching and finding in it.

Table 11: Less common challenges

<table>
<thead>
<tr>
<th>Challenges</th>
<th>EA (4 in total)</th>
<th>BA (9 in total)</th>
<th>SA (4 in total)</th>
<th>Total, with PMs (19 in total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less common</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
<td>● ○ ○ -</td>
</tr>
<tr>
<td>EA Ad-hoc, a problem</td>
<td>0 0 2 2</td>
<td>1 1 0 7</td>
<td>0 0 0 4</td>
<td>2 (11%) 17 (89%)</td>
</tr>
<tr>
<td>Motivating the models</td>
<td>0 1 0 3</td>
<td>3 0 0 6</td>
<td>0 0 0 4</td>
<td>4 (21%) 15 (79%)</td>
</tr>
<tr>
<td>Improved tools</td>
<td>2 0 0 2</td>
<td>1 0 0 8</td>
<td>0 0 0 4</td>
<td>3 (16%) 16 (84%)</td>
</tr>
</tbody>
</table>

5.4 Agile and architecture
This section will present the interviewees thoughts or opinions of agile, architecture and the combination of the two disciplines which have not already been brought up in the two previous sections. The interviewees will be divided by their role and labelled by their interviewee number (I1 – I19, see tables at the start of each sub-section). The general consensus among the employees seems to be that agile and architecture can be used together but some architectural tasks can not be divided into small enough pieces to fit in a single sprint or be delivered bit by bit due to the task's length or scope. They also believe that you need to have an initial plan or set of goals before you start...
working agile. The plan could change during the agile process but it will be hard to know where you are heading without having it from the start and there are risks that you will diverge too much from the intended path or the end result might look completely different.

### 5.4.1 Enterprise architects

There were in total four enterprise architects who all worked in the enterprise architecture office in the IT department.

<table>
<thead>
<tr>
<th>Interview number</th>
<th>Role</th>
<th>Method of contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interview 1 (I1)</td>
<td>Enterprise architect</td>
<td>Recommendation</td>
</tr>
<tr>
<td>Interview 2 (I2)</td>
<td>Enterprise architect</td>
<td>Recommendation</td>
</tr>
<tr>
<td>Interview 3 (I3)</td>
<td>Enterprise architect</td>
<td>Recommendation</td>
</tr>
<tr>
<td>Interview 4 (I4)</td>
<td>Enterprise architect</td>
<td>Recommendation</td>
</tr>
</tbody>
</table>

**I1**

I1 feels that a couple of years ago agile and architecture would clash more than what it does today. The challenge is that you need to have some kind of understanding of where you are heading before you start developing, you can not create everything right away. You can work agile if you have some done preparatory work with goal models, business drivers and used some architectural principles, such as the guidelines they have given the teams. It is hard to start working agile from scratch without knowing where you are heading or what you want for the different system areas.

I1 said that the enterprise architecture office had to adapt to the IT department's agile way of working and have changed their methods to be more agile. I1 could not see many downsides with their new way of working even for strategic questions but there is one issue when it comes to refactoring. This occurs when they start a project with a too shallow analysis the subject because of the agile way of working and have to make a delivery based on that analysis. The delivery is occasionally not quite correct because of the shallow analysis but they are still expected to build upon that first delivery. They either have to accept that they have to do some kind of refactoring and have to redo things or have a more simpler delivery that is meant to be iterated on. Neither of which is appreciated by the receivers because they feel they were given wrong information from the beginning. Different organisation and people handles these kind of situations differently. Some people are not as willing to accept that the initial plan needs to change. This issue does not occur that much for the areas they have a lot of experience of but the problem is bigger when they work with new systems or concepts which they have little experience of or is under development and it is hard to tell in which direction things are going to take.

**I2**

They try to work agile as enterprise architects but it is a bit difficult. Partly because it is hard to divide their work into small enough pieces to be able to deliver it bit by bit. They are mostly looking at the it in such a broad sense that they need an overview of
all the parts. Some tasks can not be fitted into just one sprint and needs to be done during several sprints.

I3
Previously they worked more project-oriented and worked with many projects to make sure their solutions would fit with the rest of IT landscape. This have now changed in order to adapt more to the agile way of working and they try to use the agile manifesto for a lot of things. There is some conflict where they still want to have the entire IT landscape altogether but agile wants to have it in smaller parts and to provide both is not an easy task.

I4
Other companies have made agile architecture work by combining the two disciplines but I4 thinks that their company have not come as far. In a few years they might have matured enough to be able to make it work but the question is if the company even wants to move in that direction. They have tried SAFe and some parts of the IT department have tried harder than other. If all levels are well-developed there is a good opportunity to have control over everything, even on an enterpris architecture level. The company is not there yet and has decided to focus more on the team level but the other levels are on their way.

5.4.2 Business architects
The business architects made up almost half of the interviewees, 9 in total where one was chief architect and three of were head of a department or group.

<table>
<thead>
<tr>
<th>Interview number</th>
<th>Role</th>
<th>Method of contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interview 5 (I5)</td>
<td>Business architect</td>
<td>Initial list</td>
</tr>
<tr>
<td>Interview 6 (I6)</td>
<td>Business architect</td>
<td>On location</td>
</tr>
<tr>
<td>Interview 7 (I7)</td>
<td>Business architect</td>
<td>Initial list</td>
</tr>
<tr>
<td>Interview 8 (I8)</td>
<td>Business architect</td>
<td>Initial list</td>
</tr>
<tr>
<td>Interview 9 (I9)</td>
<td>Business architect</td>
<td>Initial list</td>
</tr>
<tr>
<td>Interview 10 (I1)</td>
<td>Business architect</td>
<td>On location</td>
</tr>
<tr>
<td>Interview 11 (I11)</td>
<td>Business architect</td>
<td>Recommendation</td>
</tr>
<tr>
<td>Interview 12 (I12)</td>
<td>Business architect</td>
<td>On location</td>
</tr>
<tr>
<td>Interview 13 (I13)</td>
<td>Business architect</td>
<td>Initial list</td>
</tr>
</tbody>
</table>

I5
I5 have worked in many projects where they have tried to work agile with mixed results. I5 likes the principle of agile but how it has been implemented has varied. Some people think the most important practice is to just have recurring meetings and nothing else while other think that it means that they do not have any documentation or architecture. I5 have tried to introduce agile architecture at previous work places but there was always someone who felt that it was not needed and wanted to keep to the old ways. I5 have been on very few businesses where they have worked with
architecture in a good way and does not think the company is there yet either. I5 thinks that some agile disciplines such as Scrum have a reversed scope-creep where they try to narrow it by making it smaller and smaller. In the end there is nothing left, instead of starting small and then expanding it. You are trying to deliver a view of the business and not just build a lot of small things. I5 is not completely convinced of the agile way of working, it depends on how you use it. It sometimes leads to insufficient requirement specifications where people only specify the amount they want even if more is needed. Waterfall on the other hand does not work either. You need to find a balance in between and need to establish a general goal architecture. Not everything can be done on the fly.

I6
According to I6 “agile” is just a buzzword and have worked that way for 25 years. It is not something new, it is how you should work. If you do not work iterative and incremental you are going to encounter some problems. They try to be agile in the early stages with architecture with a Kanban board, a prioritised backlog etcetera, but it is hard to break down architectural and analytic works into small enough pieces. The tasks will take a couple of weeks and do not fit into a single sprint, so they stay up for a bit longer on the Kanban board.

I7
I7 had nothing to add which have not already been covered by the previous sections.

I8
In I8's opinion agile at its worse is when start working agile before you have defined your direction or end goal. Which means that you can easily diverge from your intended path and end up with a very different end goal. To only manage a project based on cost and the amount of money you can develop for is not that applicable in production. The needs they have to accommodate are often well-defined and if they can not keep to those needs they will never reach their goal. Without a solid plan on how to reach those goals there is a risk that people will think that they can work agile and make the plan as they go along which mean you can end up anywhere. You can of course divide the architectural work into smaller parts as well and do it step by step. The hard part is that it is hard to know when a process will end and the next process takes over. It depends on how matured the area you are working with is, how comfortable you are in your way of working and how much experience you have, then a well-trimmed organisation can even work very well following a Waterfall model.

I9
Previously the projects themselves had to do their own process modelling because there were no previous models, no one had knowledge of them or they were not used. Every project needed to spend time on their own pre-study instead of having access to up-to-date models that shows how to do things. With the process models that I9's team have created it is easier for new projects to know how to work with certain things and where they are heading before the project starts. This, in I9's opinion, makes the projects more agile when they already have a plan or foundation to start from but it relies on the models being up-to-date so they can be trusted. I9 also thinks that you can work agile when developing the models. They have often failed when they tried to design everything at once, it takes too much time and by the time you are finished
things have changed. It is also important to be flexible when making architectural decisions by keeping your options open and not painting yourself into a corner.

I10
I10 thinks agile and architecture work together where you have solid frames. At the moment they are too agile where they start a project one moment and starts another the next. They need to have a semi structured environment and where you take something from point A to point B. You can only proceed with the next task if you get stuck on the current task, when the new task is completed you go back to the previous one. This is hard to achieve but it requires you to work on the tasks you have started and not be afraid to drop assignments if you notice that you have taken too many projects. They can get mad but it does not matter because the worst thing you can do is to take someone's resources and not gain anything from it, it is better to say no and free up that resource. Otherwise the agreements, budget or deadline will lose their meaning if you can just delay them. Focus, prioritise and determination, if you have a project that has a six months deadline it should be done in 6 months and you should not plan additional projects that might jeopardise that deadline. People do not care if you work with three project and get no progress, they only care about progress even if you only work with one project. It is better to get progress on one project than zero progress on three.

I11
I11 works in an initiative where they use agile and architecture together. It was decided from the beginning that there should not be any conflict between agile and architecture which have been the case at company before. The two disciplines should go hand in hand and people were not allowed to push the architectural or agile angle in any direction. People did not have any difficulties thinking the two could be combined since they made it so clear from the start. During the development of the first prototype they realised they needed a way to keep track of their documentation. They all agreed on using a repository for their documentation and they are now heading towards a model based architecture. Compared to other repository initiatives I11 has been a part of which usually have been driven by a single person while the rest of the team did not quite understand why it was needed and it all falls flat after awhile. This time it is needs-driven and the team has realised by themselves that they need to get their architecture sorted out otherwise they can not be agile. Now models are suddenly being created and people are prioritising architectural meetings which do not usually happen when they work with Scrum, which one of the weakness of Scrum when people tend to become too idealistic. People assume that everything can be simplified too a high degree and that they already have a well-modelled architecture where all the dependencies have been removed. The reality is usually not that simple and if you want a solution to work you need to work with other organisations which might not have an agile way of working or you are dependant on a limited resource. Not everything is solved by being agile.

Their next step for combining agile and architecture is to combine it with project management. Even though they have Scrum masters and self-organised teams they need some traditional project management in order to get everything working together. The challenge is to find the right level of project management where you can be sufficiently organised while not being so strict that you lose the agile part. It is important that the project management has an open mindset towards agile and does
not just think it is an excuse to not document. Before they started this initiative I11 was convinced that SAFe was the way to go but after talking to people in the IT department I11 is not as convinced. If the IT department recommends them to use SAFe they will start using SAFe and do not want the business to decide it for them.

I12
I12 thinks the company works in an agile way but some of their products have very well-defined specifications over what the products should be able to do at a certain date and release directives that can not be delayed. The actual development of the product is very agile and iterative but they are still working towards a fixed goal. With the agile manifesto you can not forget that it is still extremely important with documentation. It says it is better to develop than to document, it does not say that you should not document at all, which people sometimes forget. I12 has asked many times what they will get if they invest a certain amount money into a project and the response is that they will receive twice as much than if they invested half the money. This is not really the answer I12 is looking for, I12 wants to know the specification and the goal they will work towards. They can not just rely on agile and think they will figure it out when they get there. Of course you can not know all the details but you still need to have some kind of direction and some check-ups to make sure they are on the right track. The goal can of course change if they realise it was not really what they wanted. For the cases where they have those clear specifications and goal they need to be more strict when it comes to the agile mindset. I12 sees no problem with even combining Agile with Waterfall. For software development I12 is convinced that they need to work agile in order to know what works and what does not. However, since their owners are less agile than them whenever they need to ask for money or permission for projects they need to have a description beforehand that is more towards Waterfall and includes some sort of goal. That they work agile to achieve those goals do not concern the owners but they can not come up with an “agile” description when asking for funding. I12 believes that they can work very agile but not only agile, some surrounding architecture is needed.

The company have had an agile way of working for a very long time but have not called it by the modern terms. When people starts pushing “agile” it sometimes scare people even though it is already well-established at the company. Previously they used prognosis instead of a budget for estimating how much money would be needed and after a certain amount of time they reviewed if they needed more or less money. Now they have a different method with a set budget which can not be exceeded. This creates a less agile way of working because you will always set a high budget as possible to reduce the risk of exceeding it or set a low budget in order to get it approved easier and it will also look more profitable with lower expectations. What they have observed is that their production is suffering at the moment, partly because they are unable to give their subcontractors a sufficient enough heads up about volumes. The subcontractors can not help them when the volume increases if they have put themselves on the lower side of the budget in order to present good numbers. This would not be as big of an issue with prognosis and they could update the subcontractors if the volume would increase.

I13
I13 thinks that agile and architecture can work relatively well together but not everything can be done with agile. Some tasks take such a long time before any
progress is done because they are dependant on information from subcontractors which can take time. Agile does not have to mean the same thing for everyone. One of the organisations they work with have a much longer pre-study before the actual development starts while they usually draws a sketch of how they want it, iterate it a bit before they start the development and take it from there.

5.4.3 Solution architects

Four solution architects were interviewed, the first two are lead solution architects and the last two are regular solution architects.

<table>
<thead>
<tr>
<th>Interview number</th>
<th>Role</th>
<th>Method of contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interview 14 (I14)</td>
<td>Solution architect</td>
<td>Initial list</td>
</tr>
<tr>
<td>Interview 15 (I15)</td>
<td>Solution architect</td>
<td>Initial list</td>
</tr>
<tr>
<td>Interview 16 (I16)</td>
<td>Solution architect</td>
<td>Recommendation</td>
</tr>
<tr>
<td>Interview 17 (I17)</td>
<td>Solution architect</td>
<td>Recommendation</td>
</tr>
</tbody>
</table>

I14
According to I14 they try to adapt to the agile way of working and it works fine when it comes to pre-studies and discovery sprints. They are able to focus on one area and gather different architects with different perspectives then they iterates until they end up with a solution. However, it relies on external conditions and there are very few projects where they are able work this way. At the moment they are not that agile because of the high pace at the company and with many project at the same time. There is a lot of focus on documentation and handing the work over to another person.

I15
The IT department is undergoing a large process of change where many people are working for the future while still getting the present to work which forces them to make solutions that “just works” and how well it fits in the architecture is not the most important aspect. They will get a new architecture in the future anyway so the solutions developed today might also be temporary solutions for the old architecture. I15 believes that many people do not think that verifying the solutions towards the current architecture is that important. The current environment at the company is very stable so many of the solutions will land within its borders anyway.

I16
I16 thinks that an agile architecture is a challenge because they have a reference architecture that is not really anchored yet and they are in the transition phase which requires a lot of planning. There is a risk that they will lose their long term plan when they have to focus that much on the current problems. They try to have contact with the business to understand what the their goals are and not just focus on their current problems. They have a portfolio and have a pretty good picture of what the business thinks about. They have started on intentional architecture from the SAFe framework but it depends on the experience of the respective solution architects and if they know the holistic view or not. An important factor is cost, the business wants to know beforehand how much it will cost in order to allocate a budget for it. This is another
challenge for agile architecture, to provide a reasonable budget estimation at an early stage for the solutions even if they do not have a clear picture. They try to convince the business to not follow the budget estimation and accept that they can deliver on a feature level or in modules. The problem is that the business have their own budget which gets allocated higher up in the organisation.

I16 thinks that you can make agile and architecture work together but you have to be careful and not be too quick to make decisions that will have large consequences later on. However, it is a good to start to think about agile architecture and how to work more agile. In the end they want to connect the lowest level, such as physical servers and virtual machines, to the architecture in order to see what the impact of shutting down a server have on a capability level and to see how sensitive things are. The issue is that on those low levels it is hard to keep everything updated but I16 thinks that the developers themselves need to update the information and models. For example if a developer switch one component to another they need to change the model accordingly and it will update the other models that is connected to the first one. Today they only have a baseline that is published but the published version does not match reality since it have to go through several steps to get approved. They want to know the current state and not just the state that got approved one year ago which is not very agile, it is more towards Waterfall. Even after each project is not enough, the best way to do it is directly otherwise it is very probable that you will forget and because of the high pace you have to jump to the next project immediately when you are done with the first one.

I17

Until 6 months ago they worked according to the Waterfall model with IT within production by having a big pre-studies, deciding everything and then start the development because of the tight deadlines. In the end they would deliver something which was not what the customer wanted because they were not involved in the project, they had changed their mind or the conditions changed. Now they are working much more agile and closer with the customers which is much better but they still want to take it to a higher level. They are at the moment only working agile on a project or development level with only one team but they want to take it to a program level where they can work with the architecture and with the development in the same backlog. Some groups have implemented this but not I17’s group. There are people who do not believe in the agile way of working but I17 wants to work more according to SAFe with a joint backlog for development and architectural tasks.

I17 thinks that the architecture have to enable the agile way of working, it needs to be flexible and you have to give feedback on it. Otherwise it can not adapt to the new requirements which is why they are working with the business, to find out what adaptations they need to implement. It is possible to do agile without the architecture but you will lose control over the system landscape and risk getting multiple systems with the same functionalities. You need a holistic perspective in order to keep an eye on the system landscape as well as a goal model for everyone to know where they are heading and how to reach it. Otherwise you will lose control and you can not maintain or support the systems. With a common structure it is easier to maintain the landscape.
5.4.4 Portfolio managers
There were two interviewees who did not work as architects and did not have the same way of working as the rest. They still experienced both agile as well as architecture in their work and have formed their own opinions about them or how they can work together.

<table>
<thead>
<tr>
<th>Interview number</th>
<th>Role</th>
<th>Method of contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interview 18 (I18)</td>
<td>Portfolio manager</td>
<td>On location</td>
</tr>
<tr>
<td>Interview 19 (I19)</td>
<td>Portfolio manager</td>
<td>Initial list</td>
</tr>
</tbody>
</table>

I18
For I18, agile is a way to create something that the users want and are happy with. Agile from an IT perspective is to build something, if the users are not happy with it then they will not use it and the work has been wasted. The architects know how to create conditions and gather information from different sources on how to build things. The business needs to be involved when the work is carried out in order for them to get what they want and make sure it fits the rest of business.

I19
Part of I19 work is to make sure that a project's goals are met and that the project leaders give them continuous reports about the progress which I19 can give feedback on. It usually starts to go wrong when a project leaves the pre-study phase and they start to realise what should be developed or when they should deliver something for the business to interpret. There are many reasons why the projects go wrong, a common one is the agile approach to project management. I19 think the projects lack a holistic planning and holistic solution, they do not really have any clear goal model to work towards. It is good to work agile by splitting up the solutions with fast deliveries and using demonstrations to get the business involved with “look and feel” in order to get faster feedback. However, what usually happens is that they plan the sprint and they divide the activities but if they do not complete a task in time everything just gets pushed into the next sprint. As a result they lose track of what was already planned for that next sprint, what the consequences are of pushing it to the next sprint and how everything is connected. There is a lack of an overall schedule over what activities should be performed in a project, who should perform them and when. They can still divide them into sprints but they need to keep track of the bigger picture and thus be able to see the effects of delaying certain sprints or tasks on the project as a whole. Another problem is when they split a task into different sprints they lose control over the overall flow. They only test the sprints isolated from each other and not how they work together or in real life scenarios.
6 Discussion

The literature review detailed in Chapter 4 will be used for discussing the results of the research together with suggestions brought up in the interviews. Two additional interviews have also been done with people not working at the company to get different perspectives on the issues in order to analyse the results. They will be labelled as Interviewee A (IA) and Interviewee B (IB). IA worked as a lead architect for a telecommunications company and IB worked as an enterprise architect consultant whose most recent assignment was at a large bank.

6.1 Discussion of the results

The methods of the architects use are somewhat expected where they are working agile and not really following any standardised framework. IB states that it makes sense to work in an agile way with the enterprise architecture considering how businesses function nowadays where almost all aspect of it should progress quickly. Requirements change faster, you are expected to respond quicker, multiple projects run at the same time and no one really have a complete understanding of how they are connected or affects each other. There are different challenges now than previously and you can not work in the same old way but these new circumstances fits the agile way of working very well. Not following any framework also makes sense for IB, it is rare to use any specific framework without modifying it to fit the company but you can take inspiration from them and use as a foundation. The challenges the interviewees face can be put into three categories: challenges caused by agile, challenges that could be solved by agile and non-agile related challenges.

6.1.1 Caused by agile

There were two types of challenges that could have been caused by agile, the removal of the meetings that the developers needed to have with the EA office before going to production and the lack of coordination between different business organisations.

EA meetings

The EA meetings where made non-mandatory as the company transitioned to agile and they removed functions that were not considered “agile”. These meetings were a good way for the enterprise architects to have a dialogue with the teams and give the teams architectural guidance. These meetings are still available to the teams but are optional and some of the architects believe they are not utilised enough. This have meant more responsibility and pressure on the lead solution architects who now have to gather the information from both the enterprise architects and from lead architects representing other areas. The enterprise architects have lost a communication channel to the teams as well some control over what the teams are doing and how their assignments are carried out. However, others think the meetings are not needed as much because of the new guidelines given to the teams by the enterprise architects. If they keep within these guidelines there should not be a need for the EA meetings, especially when working in an area which are already well-known to them. The meetings are still needed for newer areas or if they need to go outside of the guidelines for some reason.
Coordination
Some challenges are not directly connected to how the architects are working but to the business being allowed too much choice. Having more choice is not necessarily bad but it can have negative consequences, such as reducing the reusability of the company's solutions and having multiple different methods for the same task. This is consistent with the literature review and the study [27] which found that the lack of knowledge of already developed solutions can be a reason to the lack of reusability. An example of this was also brought up by IB where the company bought a new system for over €100 million. After implementing this system it was discovered that another department had already bought the same product which could have been used instead. According to IB, to prevent such issues the top management needs to decide on an architectural tool that everyone should use to store their products. In future situations when a new system is to be bought they can use this tool to see if there already exist a system that could serve their purpose. For companies of that size these kinds of decisions need to be made on multiple different levels and by multiple individuals so there are a lot of people that need to be convinced that this is the right thing to do. However, one reason for the lack of reusability that was not found in the literature review was that the business sometimes believe they are too unique and their requirements or problems are too unique and therefore need to have a unique solution.

Another possible cause is the structure of the company with independent organisations which could have contributed to some of the stovepipes that some of architects have experienced. It might also be the reason why the organisations are sometimes unwilling to use existing solutions or come to an agreement of a common way of working. The communication suffers between the different business organisations when they operate independently and some think the company have too strong informal networks which makes it more difficult to communicate to other departments without already knowing someone there. However, these structures have also contributed to the company's success by creating a smooth way of working and the organisations can improve themselves without having to include any other parties.

6.1.2 Solved by agile
The company already work considerably agile which have solved some challenges that are common with traditional EAM but there were a few challenges which can possibly be solved by using additional agile methods. These challenges fall under two categories, coordination and resources. Coordination can be improved through different kind of forums and a better prioritisation can help the lack of resources.

<table>
<thead>
<tr>
<th>Possible solution method</th>
<th>Challenges</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectural forum between EA and BA</td>
<td>Lack of EA contact, ivory tower, coordination</td>
<td>EAM, SAFe ART, Scrum and IA's company</td>
</tr>
<tr>
<td>Forum between the business organisations</td>
<td>Coordinating businesses, similar solutions, different methods, stovepipes</td>
<td>EAM, SAFe ART and IA's company</td>
</tr>
<tr>
<td>Better prioritisation</td>
<td>Lack of resources, many projects simultaneously</td>
<td>Kanban and IA's company</td>
</tr>
</tbody>
</table>
Solved by current methods
By comparing the architects' current way of working with the literature review you can see that they already work fairly agile. They incorporates methods which are associated with agile such as working iterative and incremental with smaller or early deliveries in order to receive faster feedback. This makes the architecture adapt to change better as well as ensuring early and periodically EA deliveries which is brought up in [15]. It also helps with aligning the architectural efforts with the interests of the stakeholders also found in [15] and avoiding having the models being out-of-date when they are delivered which is brought up in [6] even if only a few of them said that they prioritises time over quality or completion. Many have said that this way of working is an improve compared to how they previously worked, they have also received positive feedback from both stakeholders and customers. Most of the business and solution architects worked closely with the business and even designed the models together but almost none of the enterprise architects worked that way. Including the business or customers in the design of a model reduces the risk of it being unused or not utilised enough, something which is brought up in [6]. The company with its current methods do not seem to have any issues with ad-hoc EA demands which were the number one challenge found in the similar study [12]. Half of the enterprise architects at the company explicitly said that they do not see these unexpected assignments as an issue even if a few of the business architects see them as a problem. The enterprise architects are able to put their more strategic tasks on-hold while they work with these short-term issues and it is also a way for them to know more about what is happening at the business.

Figure 7: Agile development cycle with continuous feedback[49].
Coordination

There are two different types of forum that could be helpful for the company to implement, one is a forum with enterprise architects and business architects from a certain department. The other is a forum between business architects or other representatives from multiple different business organisations. The first forum is similar to the forum the enterprise architects share with the solution architects that is used for discussing any questions regarding architecture. There used existed a similar forum before which was discontinued but it can quite easily be resumed again according to one enterprise architect. It was easier for some department, depending on how clear their business plans and problems are as well as how transparent the department is. Many of the interviewed business architects work in one of those departments and those architects also think they have too little contact with the EA office at the moment. Some of them were positive towards a forum and the same goes for the majority of the enterprise architects. This type of forum would increase the contact between the two roles and help spread awareness of the EA office to the business considering there were instances where the colleagues of the business architects did not even know an EA office existed at the company. This could also be used as a way for the enterprise architects to get better insight into how business operates in order to reduce the risk of ivory tower[17] and reduce the gap towards the business. The second type is a forum with representatives from different business organisations where they can discuss common problems or processes and decide on common solutions or methods to use which will weaken the stovepipes and improve the reusability as well the coordination between the organisations. An example of this kind forum could be where business architects from the different organisations act as the representatives since they have knowledge of both the problems the business face and the processes they use. It could also be needed considering only one business architect that were interviewed participated in any regular forum with other business architects.

IA's company have something similar to the second type of forum. The IT department IA works in is divided into three areas and each area have a number of lead architects who are responsible for a smaller part of the area, called domain, with one or two developer teams. The lead architects represents their domain and teams in an architectural forum where they discuss architecture that affects all of them. If a new product is to be developed, one representative from each affected domain will meet together with a representative from the business in workshops where they discuss how the product should be implemented and if their systems need to change in order to support the new product. There is also an architectural forum on a higher level where they discuss the architecture more thoroughly and collaborate in order to get the bigger picture. If a team not belonging to the IT department wants to implement a new product that affects the IT, a member of this forum will also take part in the workshops. To reduce stovepipes and lack of reusability the teams are responsible over their own areas and owns the data in them. If another team wants to access the data or use their services, that other team have to go through the REST APIs.
These types of collaborations between different business organisations and business layers can be found in both SAFe's ART[50] and in conventional enterprise architecture management[4] where they promote coordination and discussion. ART also can be useful for coordinating different developer teams, which have been pointed out as a problem by all of the enterprise architects. Having regular discussions or meetings could also be found in Scrum where they have a review after each sprint as well as daily meetings to discuss the progress and any problems the teams might have[43]. There are however issues with establishing brand new forums, even for the first type which have existed before. It requires an initial amount of coordination and planning to be able to set it up which can be time-consuming as well as complicated to make everything fit. These kind of forums could be useful for other companies that have isolated business organisations or have issues with coordinating different departments. They may not need them as much since not every company have the same distance between the business organisations or between the business and the enterprise architecture office as the company in this study.

**Resources**

The most common challenge the interviewees faced was the lack of resources. It was not only the lack of personnel, they often had too many projects at the same time. They are unable to focus on what they are working on and switching between them waste unnecessary time and effort. IB described a similar situation where they had too many projects at the same time. One of the reasons was that there were many people in high positions who had strong opinions about what needed to be done. Not everyone shared the same opinions which lead to multiple people initiating their own projects that needed the same resources. When the resources ran out, they had to hire more consultants to handle some of the projects. IA's company on the other hand did not have the same problem because they have put a lot of effort into having good prioritisation. For them, prioritisation were much more important than trying to scale up projects which some of other interviewees agrees with. If a competence or resource has enough projects and was unable to receive more, any future projects were put on hold until that resource was free again which reduced unnecessary context switching.
They also tried to stress the importance of technical debt to whomever was ordering the product. Technical debt occurs when a solution is developed too quickly without being properly designed and with non-optimal design decisions, either intentional or unintentional. If the consequences of those decisions are not fixed in time, interest will build up, similar to financial debt, and the technical debt will increase even further. Eventually if it is not paid off, the debt will lead to increased difficulties for implementing change.[29] They tried to have a dialogue with their product owners and provide options, they could get a solution quickly but it would create debt or they could wait for the proper solution without creating any debt. For example, if they did it the correct way it would take a month and if they did it incorrectly they could deliver it in two weeks but it would take two months to pay off the debt created.

A possible way to mitigate the resource problem is to have a more agile and flexible prioritisation of the projects similar to IA's company or the limitations used in Kanban[44]. No additional projects should be assigned to an employee who already have too much to do until that person have time for it. This would not immediately solve the issue if they already have too many projects but it would eventually, when enough projects have been completed and the employees have enough time to work efficiently. In order to speed the process up, low priority assignments could be assigned to someone else who has the time for them, or they could discuss them with the product owners to see if they could be temporarily dropped or delayed. Incorrect resources management and priorities are known issues[4][52] and not unique to this company. Having a better prioritisation such as the one suggested could be used to address these issues. The problem is to know when the limit has been reached since it is based on the individual employee. The company also have to consider the consequences of putting new projects on-hold and decide which should be prioritised.

6.1.3 Non-agile related

The majority of the interviewees thought that the enterprise architecture office should not be a part of the IT department and that it should be closer to the business. The main issue the interviewees had with the current location was that they believe enterprise architecture is not an IT discipline, which is the same opinion as many of the sources in the literature review[4][5][6]. It is primarily a business function even though IT is an important part of it and should support the whole company, not just focus on the IT department as some feel it does today. They need to have contact with both the business and with IT but moving the enterprise architecture office closer to the business in order to have a better contact with them and to better understand what they do can have a negative impact on their relationship with the IT department. The question is where in the organisation, according to the literature review a company with a centralised IT should have a centralised EA office and a company with decentralised business organisations should have a decentralised EA office. The problem is that this company fulfils both those requirements. However, since half of the interviewees have explicitly said they think the EA office should report to a CIO, who is close to the top management, it would be more suitable to have it centralised as seen in the image below. The current EA office is also centralised even though it is at the IT department so it makes sense to keep it that way. If they were more centralised they could have better contact channels and make it easier reach out to the different departments easier which would help prevent ivory tower. They have tried a centralised EA effort before but it did not have the desired effect, partly because they
did not really work on what the business wanted them to do. Relocating the entire EA office is a large organisational change that can be a complicated process which will cost both time and money. The company does not have a CIO at the moment who the EA office can report to, the closest they have is the head of IT. An alternative to moving the entire EA office to a new location is to move it higher in the IT department's organisation. If they were closer to the top they would still get more influence and say in thing compared to their current situation where they are a couple of level below the head of IT. It is normal for a company to have a CIO that the enterprise architecture office reports to, the issues will therefore not be completely the same for those companies but their EA office can still lack influence or contact with the business. The solution of moving it closer to the business or higher up in the organisation could therefore still be applicable.

![Figure 9: A centralised enterprise architecture model][4].

### 6.2 Limitation

The interviewees were limited to the employees of the company who were willing to participate and those who have the time. As a consequence the sample size is rather small, especially for quantitative data, and it limits the duration of the interviews. The interviews themselves were also a limitation since they were done in Swedish before being translated into English and some parts could not be translated perfectly. The result is based on the experiences and opinions of the interviewees, as a result the information may be subjective or bias. The information is also limited to what they are willing to tell or remember and is reliant on them being honest which is why the participants are anonymous so they are more inclined to tell the truth. It is assumed that they can be trusted and are telling the truth. The interviews were also semi-
structured in order to promote discussion and letting the interviewee speak more freely but in combination with time restrictions meant that some of the methods and challenges were not mentioned (-) during every interview. As a consequence, some of the challenges or methods could be more common than what the result shows. When ignoring the interviews where they were not mentioned (-), most methods and challenges only had responses that confirmed that they were being used or experienced. Most of these were either only relevant for a small group of people and were only mentioned by them, or they were popular and only a few interviewees did not talk about them, possibly because they did not think of them or forgot to mention them. There were also a few instances where the interviewees did not completely confirm a challenge or method and only partially agreed (●), but those answers were a minority. These circumstances in addition to the low number of participants made the quantitative part of the result less substantial.

6.3 Validity and reliability
The research focuses on a single company and the experiences of its employees which makes the external validity limited to companies and architects in similar situations. The roles of the interviewees were, however, distributed across four different roles which increases the external validity even if the business architects were over-represented. The company and its employees have been anonymous for the study to make sure the interviewees could answer truthfully and thus ensure the reliability of the data. Every interviewee participated voluntarily and were asked for permission to record the interview which everyone agreed to. The research can be considered as reproducible and repeatable since it followed the methodology described in Chapter 3 which could be used to repeat the study and come up with the same results. However, since the results were dependant on the employees' answers, the same interviewees are needed and the study could therefore not be replicable.

6.4 Future research
The research can be used by the company or other companies in similar situations to implement changes based on the discussion in Chapter 6.1 or use the data from the interviews to gain, possibly new, knowledge of what challenges the architects face. The research could also be expanded on to further investigate the use of agile methods in architecture. A complementary study could be done with developers or people from the business to get their perspectives on issues that concerns them, such as their contact with the enterprise architecture office or with other business originations. The research could also benefit from a larger quantitative study which could be achieved by increasing the number of participants, either within the company or by widening the scope to include other companies and countries. Having a broader scope would also allow researchers to examine how the results varies depending on the company or country. To accomplish this, a survey could be sent to the participants, which would avoid some of the uncertainties since everyone gets the same questions. It would also make it easier to reach out to other companies and especially those abroad. This kind of survey could have also been sent out to the interviewees in this study to clear up some of the inconsistencies in the result.
7 Conclusion

The research investigates what agile methods architects can use to improve their way of working. The approach of how this was achieved is detailed in the thesis and the result of the research is presented. However, the possible improvement from agile is fairly modest since they already have implemented many agile methods. They work iterative, incremental, have early deliveries and perform changes based on feedback. Some of them follows a standardised framework with Astrakan and SAFe being the most popular. The only regular well-established forum the interviewees have was a weekly architectural forum between the enterprise architects and the lead solution architects.

The interviewees experience a number of challenges, the most frequent was the lack of resources with one of the biggest issues was that they had too many project simultaneously which takes unnecessary time and effort. A possible solution for that could be to adopt a more agile prioritisation and not assign more project to the employees when their limit have been reached or cancelling excess projects if they are over the limit. There were also challenges with coordination, between the different business organisations as well as between the enterprise architects and the business architects. For the first one, it was mainly that the different business organisations did not discuss enough with each other and they were developing or using different systems which do pretty much the same thing as well as not reusing existing solutions. A way solve those issues could be to have a forum where representatives from the different business organisations can discuss common issues or solutions and come to an agreement of what to use. For the second challenge the issue for the business architects were the lack of contact with the enterprise architects and many believe the EA office suffers from some form of ivory tower. Another forum could be a potential solution for this but between the enterprise architects and the business architects, similar to the one they have with the solution architects. This would improve the contact between the two and the enterprise architects would get more knowledge about the business. On the subject of the EA office, most of the interviewees thought it should not be at their current location which is at the IT department. The main concern was that they think enterprise architecture is a business function, not an IT function and believe it should be closer to the business. Some of the employees also worry that the EA office does not have enough say in things because they are too far away from the top management. Centralising the EA office at the business under a CIO could potential solve it but moving the office closer to the top in the IT department could also be an alternative to still get more influence without relocating the entire office.
References


[23] Priyanto, A. A.*Bringing Enterprise Architecture to the Boardroom*. Master of Science in Business Information Technology, School of Management and Governance University of Twente, 2013


Software Quality Journal, 18(4), 437-468, 2010


[48] Hensema, M. Applying Agile in Enterprise Architecture. Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS), University of Twente, 2015


[53] Tsigkas, A. *The lean enterprise from the mass economy to the economy of one*. Springer, 2012