Enterprise Architecture
Success Factors:
Analysis from Architects’ Perspective

MERT CANAT
Enterprise Architecture Success Factors:
Analysis from Architects’ Perspective

Mert Canat
Master’s Thesis Report in Collaboration with Ferrologic and Volvo Cars
Department of Network and Service Engineering
KTH Royal Institute of Technology
Stockholm, Sweden
Abstract

Enterprise architecture aligns organizations business processes, information systems and technical layers. Its role has become more challenging than ever before with the modern day rapidly changing environment and technological advances. Considering these aspects, this thesis tries to evaluate the success factors that affect enterprise architecture management.

The thesis is done at Volvo Cars, in collaboration with Ferrologic. Twelve success factors significant for Volvo Cars’ enterprise architecture management are defined after a series of semi-structured interviews with architects working at the company. This is followed by a survey evaluating the factors sent to architects throughout Sweden. In the end, the factors are divided into four groups according to their impact level. The survey reveals that the business understanding of the technical side, requirement definitions & handling requirement changes, high-level management involvement, and cross-functionality are perceived to be most impactful success factors for the industry professionals, in no particular order.

Sammanfattning


Acknowledgements

I would first like to thank my supervisor at KTH, Robert Lagerström for his guidance and feedback during the study. He has been available whenever I needed help.

The opportunity Volvo Cars and Ferrologic provided made this thesis happen. I would like to thank my industry contacts, Magnus Jandinger from Volvo Cars, and Johan Wretö and Per Torell from Ferrologic.

This study would not have been possible without the participants. I would also like to thank everyone who participated in the study.

Finally, I would like to thank my family and everyone who has supported me all the way.
# Table of Contents

1. **Introduction** ........................................................................................................ 5  
   1.1 Enterprise Architecture .................................................................................. 5  
   1.2 The Company .................................................................................................. 5  
   1.3 The Study ........................................................................................................ 6  
   1.4 Outline ............................................................................................................ 6  

2. **Background and Theory** ................................................................................ 7  
   2.1 Enterprise Architecture .................................................................................. 7  
   2.2 Waterfall ......................................................................................................... 9  
   2.3 Agile Methodologies ...................................................................................... 9  

3. **Methodology** .................................................................................................. 10  
   3.1 Phase 1: Literature Review and Initial Evaluation ....................................... 10  
   3.2 Phase 2: Detailed Evaluation ...................................................................... 11  
   3.3 Phase 3: Investigating Success Factors ....................................................... 11  

4. **Phase 1: Literature Review and Initial Evaluation** ......................................... 12  
   4.1 Literature Review ......................................................................................... 12  
   4.2 Initial Evaluation .......................................................................................... 18  

5. **Phase 2: Detailed Evaluation** .................................................................... 24  
   5.1 Interview Questions ..................................................................................... 24  
   5.2 Demographics of Respondents ................................................................... 25  
   5.3 Common Findings ........................................................................................ 26  
   5.4 Interviews .................................................................................................... 29  
   5.5 Determining Success Factors ..................................................................... 45  

6. **Phase 3: Investigating Success Factors** ...................................................... 48  
   6.1 Survey Design ............................................................................................... 48  
   6.2 Demographics of the Respondents .............................................................. 49  
   6.3 Data Analysis ............................................................................................... 51  
   6.4 Respondent Comments .............................................................................. 56  

7. **Discussion** .................................................................................................... 57  
   7.1 Determining Success Factors ..................................................................... 57  
   7.2 Investigating Success Factors ..................................................................... 57  
   7.3 Profession and Experience .......................................................................... 59  
   7.4 Interview-Survey Comparison .................................................................. 59  
   7.5 The Company’s Future Transition .............................................................. 60  
   7.6 Validity and Reliability ............................................................................... 61  
   7.7 Limitations .................................................................................................. 61  
   7.8 Future Study ............................................................................................... 61  

8. **Conclusion** .................................................................................................... 62  

References ........................................................................................................... 63  

Appendix ............................................................................................................... 66  
   Appendix 1: Survey Questions ....................................................................... 66  
   Appendix 2: Survey Answers ......................................................................... 68
1. Introduction

This section informs the reader about the basic concepts related to the study, the company, research and research questions. An outline of the content is provided in the end.

1.1 Enterprise Architecture

Enterprise Architecture (EA) is an accepted discipline that identifies an enterprise’s hierarchy in terms of different layers [33] [48]. It provides principles to model and analyzes the organization’s current and future state. Capturing both business and IT aspects, it provides a holistic view [31]. It is often used to model organizations’ technical infrastructure, to document details, and to define connections between different layers such as business, information, application, and infrastructure. It aims to ensure alignment between different layers and increase the overall performance [36].

Nowadays, coping with the rapidly changing landscape has also become a greater challenge than before for EA [8]. To ensure the success of EA, which refers to its functionality within an organization, it is not uncommon to see pieces from the agile way of thinking being borrowed [1] [3] [24] [21] [30]. Agile principles originally evolved with software development and attempt to improve the development process by increasing flexibility and prioritizing time-to-market [18]. It supports individual freedom and boosts communication. They are not strictly enforced rules [23] [35] and consist of suggestions to be followed. While EA and organizational success are the main focus, agile principles are also considered throughout the study.

1.2 The Company

The contact person, Magnus Jandinger, works at Consumer & Enterprise Digital Department in order to make existing architecture more compatible with the new requirements. Because he is mainly working with the information layer, the study keeps that in mind and tries to be aligned with it. The information layer has a huge impact on translating business needs to the technical side. As a side note, the author is introduced to Magnus Jandinger via the contact people at Ferrologic, Johan Wretö and Per Torell. Ferrologic is a Stockholm based consulting company working with enterprise architecture and business analytics.

The existing EA tools at Volvo Cars mostly use the waterfall model, which will be revealed in Section 5.4, and they have very little or no agile component. There is a huge amount of legacy applications and the overall structure has a low level of architectural maturity. The company has overall 1300 applications with more than 40000 integrations between them. It is very common that some of these integrations have very little documentation. Some dependencies are unknown until someone tries to use a specific application. Due to the high number of applications in the company, analysis of the repository to evaluate its agility is not feasible during the span of the thesis. Consequently, this study focuses on having meetings with stakeholders to gather information regarding the company.
Consumer & Enterprise Digital Department is going through two architectural changes. The first one is that the information layer is getting overhauled to answer standardization and common platform needs in the short term. The contact person is the main responsible person for it. The second change is that the company is moving towards a cloud platform in the long term which will be able to maintain the old legacy applications and adapt to the new requirements.

1.3 The Study

Volvo Cars have been going through an ongoing change and their target plan is clear. This thesis tries to understand important factors playing a role during the transition. Focusing on the information layer issues, the aim is to and evaluate the current enterprise architecture management (EAM). The research questions are the following.

*What are the main success factors for Volvo Cars’ enterprise architecture management?*

*In general, how impactful are these factors from an EA of view?*

This thesis follows three phases to answer these research questions. First, meetings are done with the main contact people in Ferrologic and Volvo Cars, which is in parallel with a literature review. Second, detailed semi-structured interviews are conducted with eight architects in Volvo Cars. The resulting data is used to determine success factors. Third, a survey answered by fifty-one experts in the industry reveals the significance and basic characteristics of these success factors.

1.4 Outline

The report consists of 8 sections including this one. Section 2 provides a background for the relevant concepts. Section 3 dives further into the study. It explains the methodology and gives details about three main phases the study has. Sections 4, 5, 6 presents phases 1, 2 and 3, consecutively. These sections also include the author’s findings during the three phases. Section 7 discusses the findings in detail including limitations and future work. Finally, Section 8 wraps the study up and concludes it.
2. **Background and Theory**

Enterprise architecture is the most defining concept for this thesis. In addition, agile methodologies and the waterfall method have been used in software development. These concepts are explained in this section in order to have a common understanding of them. Other concepts throughout the study are explained when necessary.

2.1 **Enterprise Architecture**

There are a number of different ways of explaining EA which refer to the similar disciplines. Zachman defines it as the following [48].

> “Architecture is that set of design artifacts, or descriptive representations, that are relevant for describing an object such that it can be produced to requirements (quality) as well as maintained over the period of its useful life (change).”

In order to practice EA in an efficient manner, a couple of frameworks emerged over years. Zachman Framework [48] and The Open Group Architecture Framework (TOGAF) [49] are some well-known examples. In order to visualize the overall development method, the Open Group released the TOGAF Application Development Method (ADM) provided in Figure 1.

![Figure 1: TOGAF ADM](image-url)
As shown in Figure 2 [50], EA works on the organization as a whole. Some architects are more specialized. For example, information, application, and infrastructure architecture work on business & information, application & application infrastructure, application & core infrastructure, respectively. On the other hand, solution architecture focus on all these layers in the scope of a specific application or solution.

It is important to differentiate between enterprise architecture and IT architecture. IT Architecture covers a subset of EA in the technical layers with a distance from business processes [51].

![Figure 2: Architecture in different layers [50]](image)

In order to be able to apply certain frameworks, the organization’s current maturity level needs to be known. Capability Management Maturity Integration (CMMI) Institute [33] defines the maturity levels as shown in Figure 3. Here, the maturity refers to the level of compliance to the certain architectural criteria. Maturity levels here provides a guideline about what needs to be current focus. It mentions that the organization needs to adapt everything in the lower levels first before getting to the higher levels.

![Figure 3: CMMI Maturity Levels [33]](image)
2.2 Waterfall Model

Waterfall model is a well-established early software development method emerged in the 1970s. It includes a number of steps to be taken in the software development as it can be seen in Figure 4. In the past, the business environment was relatively more predictable compared to today. The method utilizes this and provides a plan-driven, and easily scalable methodology [22]. Waterfall model heavily influenced EA methodologies when they were developed. Becoming less and less popular, the model is still used in the industry and software development [22] [38].

![Figure 4: Waterfall model](https://airbrake.io/blog/sdlc/waterfall-model)

2.3 Agile Methodology

Agile methodology is a relatively new approach to software development formalized in 2001. It promotes flexibility and freedom with a set of principles. Agile Manifesto [18] states that

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

should be valued. Even though the agile methodology is less scalable than waterfall [22], its ability to adapt to rapid changes made it crucial for the modern software development. Frameworks such as Scrum² and Kanban³ are developed and agile methods are being heavily used in the software development.

Agile methodology impacts EA management, in the sense that the agile way of thinking is sometimes involved. Also, the agility of the architecture refers to the degree to which couplings and dependencies are managed. In this context, when agile architecture is mentioned, it refers to the application of EA where agile principles are utilized [21] [24].

---

1 Waterfall Model: [https://airbrake.io/blog/sdlc/waterfall-model](https://airbrake.io/blog/sdlc/waterfall-model)
2 Scrum: [https://www.scrum.org/resources/what-is-scrum](https://www.scrum.org/resources/what-is-scrum)
3 Kanban: [https://www.versionone.com/what-is-kanban/](https://www.versionone.com/what-is-kanban/)
3. Methodology

The study consists of a mix of qualitative and quantitative approaches. It can be summarized in three phases.

(1) Literature Review and Initial Evaluation
(2) Detailed Evaluation (with eight semi-structured detailed interviews)
(3) Investigating Success Factors (with a survey answered by fifty-one professionals)

Phase 1 involves having an initial evaluation of Volvo Cars and the existing literature. The meeting with the contact person at Volvo Cars was the starting point of the study. It was an open discussion and allowed gaining a basic understanding about the issues the company is facing. This is followed by a literature review and other open meetings with them.

Phase 2 is the qualitative part of the study. After gaining an understanding in the previous phase, a semi-structured interview has been formed. Eight detailed one-hour interviews have been conducted with architects at Volvo Cars. The findings are investigated and common points are extracted. This phase allowed having a detailed of the organization’s architectural landscape. Using the findings so far from the literature review, initial evaluation, and detailed interviews, success factors that affect the Volvo Cars’ architecture are extracted.

Phase 3 consists of conducting a quantitative survey and its analysis. The aim was now to understand which of these factors are more impactful. The factors are turned into a survey and shared with professionals in the industry. Answers of industry experts are analyzed.

The following subsections explain the underlying reasons why the study is formed with the above structure.

3.1 Phase 1: Literature Review and Initial Evaluation

This is the phase when the whole structure of the study is finalized. Meetings with the main contact person at Volvo Cars and the contact people at Ferrologic allowed an understanding about the future scope of the study.

These meetings are accompanied by a simultaneous literature review. It allowed having an understanding of EA and associated challenges. Sources from IEEE, sciencedirect.com, Google Scholar, Springer, and ACM Digital Library are used.

The end result of this phase was to determine how to proceed with the study which is a mix of qualitative and quantitative approaches. The mixture is feasible and also an effective approach [43] [44]. This methodology allows for two things. Namely, a qualitative study focused on a single company’s specific issues and a quantitative study with professionals from different organizations. The path is to understand Volvo Cars’ issues and then to investigate how the professionals in the industry prioritize these issues.
3.2 Phase 2: Detailed Evaluation

This phase builds up on the previous phase in the sense that it verifies the findings. On top of that, it gives a deeper understanding. Phase 1 defined the general EA challenges and Volvo Cars’ future path. This phase reveals which of these EA challenges are more prominent at Volvo Cars and it also reveals if there are some other challenges for them.

During this phase, detailed semi-structured interviews are conducted with the architects at Volvo Cars. The semi-structured interview format is chosen because of its flexibility. Similar to the structured interview format, the semi-structured interview format has a questionnaire to be followed in order [42]. Though, unlike structured interviews, semi-structured interviews allow the interviewer to ask other related questions to the interviewees [42]. Semi-structured interviews are accepted as a valid method and used in the academia [1] [9] [12] [17] [30].

The interview questionnaire is based on data generated during phase 1. Each interview lasted approximately one hour. In order to maximize the input from interviewees, architects who are experienced in their field and been in Volvo Cars for a significant time were contacted. In the end, a number of success factors are derived using the interviews and the findings from phase 1.

The result of this phase is an answer to the first research question mentioned in Section 1.3, which is the following:

“What are the main success factors for Volvo Cars’ enterprise architecture management?”

3.3 Phase 3: Investigating Success Factors

The main purpose in this phase is to bring the qualitative data gathered so far in a quantitative form. This phase turns the success factors into a survey which is sent out to professionals in different companies in the industry. A survey is chosen to collect quantitative data as it is a common tool used in the literature [3] [11] [13] [19]. Since the survey was distributed online, increasing the number of participants was a challenge. The survey is designed to be fast to answer and easy to understand. See Section 6.1 for details such as how the data is quantified and how the answer options are decided.

It is important to note that the results in this phase represent the view of the industry. This target group is chosen for accessibility reasons since a high number of participants is needed for accuracy in quantitative studies. How these EAM success factors are valued in general is investigated. The second research question is answered which is the following:

“In general, how impactful are these factors from an EA point of view?”
4. Phase 1: Literature Review and Initial Evaluation

Initiating the actual study process, Section 4.1 explains the existing literature and Section 4.2 presents the results of initial meetings with Volvo Cars and Ferrologic.

4.1 Literature Review

The starting point of the search was to understand EA and underlying concepts. Therefore, studies evaluating EA [10] [12] [17] [27] [52] [54] [55], software development [16] [47] and relevant concepts such as modeling [7] [14] [25] [26] are included. The relationship between EA and agile principles was also a part of the study, which is not unusual in the industry [1]. There is an interplay which happens in different ways. Agile principles getting applied in EA [3] [8] [9] [11] [19] [24], the feasibility of applying agile tools directly to the architecture [6] [20] and EA guiding agile teams [1] [30] are some examples. The final goal was to gain a broad view on EA.

Here are the findings of the literature review.

Enterprise Architecture

First, let’s take a look at enterprise architecture by itself and the challenges associated with it.

Blomqvist et al [17] strengthen the view that EA is very strong for translating business strategy into actions. EA could be very beneficial during the strategy formulation phase, but it is not EA’s main functionality. This finding is supported by the empirical findings. An interesting finding is that EA’s contribution depends on the role of the strategy. EA is effective in case the role is defining limits or capabilities, whereas it is not much effective in case the role defining mission and vision [17].

According to Lindström et al [13], the biggest concerns of the information architects are trying to improve the interplay between technical and business side, to decrease IT costs, to boost modifiability and maintainability, to improve quality of operations and maintenance, and to provide support tools to the business. Looking from the software perspective, reducing the coupling level is the number one concern of the information architects [19]. Other important information architect concerns are the lack of decision support and support for estimating costs explicitly [13].

Babar et al [24] mention that enterprise architecture has some weaknesses when it comes to software development requirements. It lacks when a requirement needs to change, inhibits flexibility due to constrained requirements and does not help to prioritize requirements. Also, estimating the amount of work and resources needed for the architectural components is not always accurate. It is easy to designate too much or too fewer sources on projects [24]
Thummadi et al [9] explain three core drawbacks of EA: roles and responsibilities might be defined too unclear, communication between modules might be too porous, and teams might get too low empowerment in their local decision making. The conclusion is that agile EA method needs to follow similar principles compared to agile teams, especially about self-organizing components.

**Architectural Models / Meta Modeling**

Modeling is a very common part of EA [40]. Being able to create the right amount of modeling and documentation is an issue in EA. Having too much of them might cause stakeholders not to value EA [1] [3] simply before it requires too much effort to understand and utilize existing models. Thus, architects need to find a middle ground. To ensure the agility of an architecture, focusing on design patterns and making these design patterns user-centered and model-driven is crucial [24] [46] [47].

Modeling is a very common principle in engineering [7]. It can be used for various purposes such as documentation, analysis, and design. To be more specific, it can help in communicating the information to different stakeholders via documentation. Similarly, a holistic model helps in analyzing how the system performs with respect to requirements and possible improvement. Meta modeling boosts the benefits of modeling even more and should be employed whenever possible [7] [14] [26] [52]. Here, meta modeling refers to the act of providing template or surrogate models for different scenarios whenever it is possible. It allows a standardization for the knowledge transfer, therefore, reducing the workload while communicating information. One of the reasons models are not employed enough is that they are sometimes perceived too complex [1]. This is often because of a need for learning how to read a model, and metamodeling decreases this need. A meta model provides a ‘pattern’ which has its own advantages and disadvantages. Patterns do not provide the total solution but they provide insights to save time during development. Developer productivity increases, proven solutions have an increased chance of being reused, consistency between different applications is established [25]. On the other hand, a large number of patterns might be needed to be useful. It can cause ‘not-invented-here’ syndrome meaning that stakeholders might be more eager to buy an external solution rather than building it from scratch in case there is no meta model. Finally, these meta models are not code and programmers might need to be educated about how to use them properly [25].

While being an essential part of EA, modeling needs a heavy workload as it requires different aspects of the organization to be considered. Data collection is an important challenge [56] and its automation for EA models increases efficiency drastically and it is applicable [15].

Modeling exists in the higher and lower levels. While tools such as ArchiMate⁴ and Architectural Description Languages (ADL)⁵ can be used to represent higher-level application, information, business layers and the links between them, lower level representations such as Unified Modeling

---

⁴ ArchiMate: [https://www.archimatetool.com](https://www.archimatetool.com)
⁵ ADL: [https://www.todaysoftmag.com/article/2241/architecture-description-languages](https://www.todaysoftmag.com/article/2241/architecture-description-languages)
Language (UML)⁶ and Business Process Model Notation (BPMN)⁷ provides a more detailed representation of narrower concepts in a layer [11].

Ambler [25] investigates UML in detail. UML provides sequence and activity diagrams to express the logic in the system. These diagrams act as a bridge between different models and explain the static structure of the system. The industry experiences a shift from traditional applications where a database is connected to applications to applications are connected to different objects as known as nodes. UML allows usage of nodes which provide extra flexibility and visibility in case one of these nodes change. While traditional structure needs a system-wide interruption in case of a change, it can now be handled within a single node using UML. UML also provides extra documentation. It should be noted that UML is most descriptive in the business layer.

Gill [11] argues if high leveling modeling standards can replace low-level ones in agile EA environments. The study compares ArchiMate, which is a high-level modeling standard with lower level standards such as UML. The end result is that both modeling languages have their own use and cannot replace one another and an architectural model which uses multiple modeling languages is proposed. One of the reasons is that higher level modeling languages have a closer connection to the business. This study implies that the standardization of tools should be considered carefully in the sense that it should be done only when it does not limit functionality.

**Business Alignment of IT**

Lindström et al [13] and Hinkelmann et al [14] links survival of modern organizations with their capability to deal with continuous rapid change. Alignment of business and IT is a major concern for success which EA can answer for [34] [36] [37] [54] [57]. Business needs drive the market which is also true in the automotive sector [22]. Flexibility and responsiveness of the software determine the success in the sense that rapid change in the software should be able to compensate relatively less frequently changing hardware. Understanding the needs in both the IT and the business side is crucial. It is often the case that architects put too much focus on IT while neglecting business side [3].

Hinkelmann et al [14] emphasize the positive effect of modeling on the alignment between business and IT. To ensure the alignment, modeling has to be done within the enterprise architecture and has to be continuously and automatically monitored with architectural tools. Thus, any information passed from business to architecture will automatically update the corresponding model. Though, giving the business side an understanding of EA functionalities and getting their involvement has been a long-term challenge.

---

⁶ UML: [http://www.uml.org/what-is-uml.htm](http://www.uml.org/what-is-uml.htm)
⁷ BPMN: [http://www.bpmn.org](http://www.bpmn.org)
Requirements

EA provides information regarding requirements and their inter-dependencies [39]. Increasing volatility of software requirements makes this attribute of architecture crucial [27]. What agile methods can bring to architecture is that it can minimize the waste so that architectural workload upon a change is also minimized. Thus, architecture can adapt to changing requirements easier. It is important to note that some requirements may need to be traded off against others to even make the overall system feasible [24].

Dependencies and Coupling

Dependency management is one of the crucial factors in the software development [53]. They should be identified as soon as possible during the development and system design phase. This would enable dependencies steering the delivery strategies to the customers [5]. Taking dependencies into consideration might seem to increase the complexity, but its long-term benefit would enable sustainability. Because of this reason, it is good to reduce dependencies whenever possible [25].

IT infrastructure covers a subset of EA. The organizations need to be able to adopt changes with minimal cost, both financially and labor-wise. The coupling level, which is degree the software components depend on each other, is a major factor that affects the IT agility of the software portfolio [2] [4] [5] [28] [29]. MacCormack et al [29] argue that applications which have higher coupling come with a higher maintenance cost and they are more stationary. It is less likely that they will be retired or commissioned. They decrease organization’s ability to modify these applications as well. While developing new systems, organizations need to make sure that dependencies are used only when its necessary and the dependency’s effect on the overall portfolio is fully understood. IT managers have a critical role to play to increase digital agility [29].

---

8 Software coupling: https://whatis.techtarget.com/definition/coupling
As shown in Figure 5, components might be connected to each other in different ways which affect agility in different levels. Another study by MacCormack et al [28] shows that IT architecture affects IT agility by a great deal through coupling levels, specifically indirect and cyclic couplings are the most effective [28]. Between them, cyclically coupled couplings which are mutually interdependent are very hard to manage [29]. If possible, having a quantitative analysis of the repository reveals the couplings between applications. Design Structure Matrices (DSMs), which evaluates the coupling levels with evaluating applications in a two-dimensional matrix, seems to be an effective way of this analysis [2] [4] [5].

Legacy Applications

Baldwin et al [4] states that in order to make the legacy applications compatible with the current systems, the legacy ends up are rarely rewritten, but instead, a platform is built on it which provides an interface to the legacy. Due to this approach, developers who work today face the consequences of decisions made a long time ago. These consequences are also called “technical debt”. Design priorities change over time and technology today needs different requirements. For example, limited hardware made performance a crucial requirement, and the limited external communication options made security less of an issue. Today, reliability is valued over performance and security is more important than ever. On top of these, the design choices are less likely to be documented compared to usage information which requires today’s developers to work an extra step to be able to modify the legacy code.

The Interplay Between EA and Agile

According to Hensema [3], EA suffers from being able to adapt to today’s rapidly changing conditions. For example, having an outdated architectural repository is a result of EA not being able to keep up with IT or business which is not desirable. This is when agile software development might play a part in. Dealing with changing requirements, reflection on the ongoing processes and focusing on the essential parts are the three most important agile practices that EA can benefit from. Agile EA focuses on individuals and tries to increase their motivation. It focuses on the essential and prioritizes time-to-market. All these show that if the philosophy of agile methodologies can be implemented into the architecture practically, this might compensate previously mentioned weaknesses of EA [3]. Babar et al [24] argue that architecture using agile principles does not require the rest of the projects to use agile methods.

Literature shows that enterprise is usually in the higher levels whereas agile methods are in the lower levels [1] [11]. Communication, reducing the gap between developers and architects, and system reusability are some important challenges [1].

Bellomo et al [19] try to gain insight about which quality attributes contribute towards an agile architecture. They conclude that interoperability (cross functionality), modifiability, performance, and availability are the biggest concerns. One thing this study does is that it specifically compares hardware involved and pure software companies. This metric might be especially valuable for
Volvo Cars as this shows modifiability and cross-functionality are even more important hardware-related involved companies. On the other hand, deployability, usability, and security are less of a concern of these companies [19]. This might be because of the fact that hardware does not need to change as often, which makes it require less frequent deployment while decreasing its significance. Similarly, having a relatively isolated hardware means that security will be less of a concern.

To have an increased agility, the architectural solutions should be scalable. The existing solution should be refactored depending on the case [24]. This is possible only when the key components are upfront, meaning that the core of the structure should be designed prior to the project and they should be adjusted during the project.

Building upon the same topic, Hauder et al [31] argues that EA management needs to be business driven, collaborative and be focused on generating valuable architects. To make EA management more agile, the involvement of architects in the project level is important. The study conducts a detailed survey on agile principles applied to EA management. The results show that cross-functionality, incremental & iterative approach, and performing tasks in a self-organized manner, incorporating feedback and leader involvement is most valued factors by the industry experts.

Hanschke et al [30] try to understand if agile methods can be directly used to create architectural deliverables and if enterprise architects can collaborate with agile software development teams. It is possible to create architectural deliverables agile tools such as Scrum. Another important finding is that the EA function can empower agile teams [30].

**Summary**

The literature review provides the challenges associated with EA, which will be later be combined with the initial evaluation to decide on the interview questions in Phase 2. Also, it is going to be used to determine the survey question in Phase 3. The following are the main findings:

- Modeling is a widely accepted principle which is one of the cores of EA. Meta-modeling allows standardization and increases reusability. At the same time, they require extra work to be understood and utilized. Architects need to find a balance and use the correct amount of modeling, which is often case-dependent.
- EA architecture is useful for translating business into strategy and to establish this, the technical side should be aware of the business needs. The same goes for the other way, meaning that the business side needs to be aware of what is technically capable.
- A good requirements management comes with a good landscape discovery, which ensures that requirements will less likely to be changed and decrease the overall cost. The discovery needs to reveal the requirements in a way that, the cost will be minimal in case of a change.
- Dependencies are highly related with the IT agility and having less of them decreases the software development cost. Dependencies should be avoided whenever possible and should be documented in order to be able to analyze them.
• Legacy applications are less likely to be documented due to relevant people not being available anymore. In the long run, this causes developers to work extra which is an undesirable situation.
• The management involvement is crucial in the organizations. Many architects suffer from decision support from the management.
• Survival of an organization is linked with its ability to perform continuous rapid changes. Cross functionality is an important step towards agile architecture, which boost an organization’s ability to perform these changes.
• An accurate communication with stakeholders increases the alignment within the architecture which can be achieved by defining roles clearly and assigning individual responsibilities.

4.2 Initial Evaluation

This section explains the findings after the author’s meetings with the main contact people at Volvo Cars and Ferrologic. An overview of the current development and architectural challenges are presented which is followed by the planned future changes in the organization. Because the current challenges are mainly involved with the information architecture, the study will focus on the information layer. The data gained here is later used to determine the nature of the survey in the following phase.

Consumer & Enterprise Digital Department, mainly divided into two parts. One part is faced towards customers and they operate work with software development including DevOps9 and Agile. The other part is faced towards architecture and it deals with traditional legacy systems. The challenge is to link these two parts and other departments in an efficient manner. Because the company aims to utilize DevOps for software development and not the architecture, it is not covered in the study. Though, they see a value in the agile way of thinking when it comes to architecture. The study investigates agile from that perspective.

Development Challenges

The software development itself is done using agile methodologies which has three issues.

• Standardization is crucial. Recently, the product teams started to work with certain standards which were not the case before. They were not even using a common modeling tool.
• The contact person works on micro services10 and intends to create centralized product clusters in the end which would help agile teams navigate in the architecture.

---

9 DevOps: https://theagileadmin.com/what-is-devops/
10 What are micro services? https://microservices.io
• Agile teams working in 2-weeks Scrum sprints\(^{11}\) is not the most effective way since this time period is not always enough to generate meaningful results. Either the sprint structure or the designated work in a sprint needs to be changed.

Despite all the issues at agile development, it is still beneficial and should not be discarded. Architectural tools need to be easily available for the agile teams so that the gap between them and the architects is minimized. The architectural structure needs to be able to utilize the teams in an efficient way. At the same time, architectural work can benefit from an agile mindset.

**Architectural Challenges**

The information layer is not united and the architectural repository is not fully updated. The department used to have a good number of information architects in the past. The number of information architects decreased over time due to managerial decisions and the department has only three people with information architecture knowledge. Managerial decisions had an especially significant impact due to the highest paid person’s opinion, also known as HIPPO, culture. The individuals had a relatively less impact on this due to the lack of accountability on them. Their responsibilities were not clearly defined and it ended up causing individual architects to not take the initiative. Right now, it is believed that increasing the number of information architects and having a common architectural repository would increase the flexibility drastically. Though, the experience in the past shows that the delivery times might not change.

Architects do not have a common enterprise tool. There is no good functionality to decide which tools should be used. They rely on their personal knowledge which also suffers from a lack of knowledge transfer between architects. For example, Sparx\(^ {12}\) is very good for single-person editing at a time, and to create tools that are going to be used by the others and to visualize. It is mainly aimed at the lower level. Though, it does not provide a tool that everyone can contribute simultaneously. Even simple tools such as PowerPoint can be more powerful in that aspect. Sparx is not that well for enterprise architecture level as well. This was one example with a very common tool, whereas there are many other tools that are now well-known. Having common enterprise tools would let architects get informed about them. Architects need to be encouraged to inform about the tools they are using to bridge this knowledge transfer gap. At the same time, only the tools that increase efficiency should be used. The ultimate goal is to increase the time spent on development and to reduce the time spent on other tasks.

Meta-modeling is another concern. They are frequently used in the information and business layers in order to provide a template. They provide a level of standardization in terms of approach to a given problem. It is a discussion among the architects in the department that they should not try to standardize tools and the repository in application and infrastructure layers simply because it will be too costly, and it will be hard to get a full grip of the situation for the people working on these layers. They believe that the focus should be information and business layer.

---

\(^{11}\) Agile sprint: [https://searchsoftwarequality.techtarget.com/definition/Scrum-sprint](https://searchsoftwarequality.techtarget.com/definition/Scrum-sprint)

\(^{12}\) Sparx: [http://www.sparxsystems.com/about.html](http://www.sparxsystems.com/about.html)
In order to boost usage of meta-models, understanding why meta-models are used are important. Different meta-models are connected with each other, but not the actual structure. This means that meta-models create an easy way of understanding the overall processes without actually creating an extra workload.

Everything is going towards a micro service-oriented approach where scaling changes on a minute basis. Ensuring alignment between different layers becomes a harder issue in this case. In order to achieve this alignment, application and infrastructure layers should be included to meta-models in addition to information and business layers. Though, it is true that the architectural work on application and infrastructure layers generates relatively less benefit. The work on these layers is moving towards DevOps and also agile in terms of software development. These two are growing disciplines that give different answers in different parts of the organization. There is no generalizability.

Architects need to have a broad view and be aware of the full spectrum of enterprise architecture. Focusing on one point and ignoring the rest is not a good approach.

**Volvo Cars’ New Cloud Platform**

Not everything is about development and architecture. Business needs change and it impacts the technological side. The world is in the big data era which requires Volvo Cars to adjust. This is why the company is creating a new cloud platform to be deployed in their cars.

This translates to a new layer in the current architecture. In order to avoid directly dealing with legacy applications, they intend to create a mid-layer that links legacy applications with the business layer. This new system is scheduled to be ready in 5 years which is a very long time. To be specific, this new layer is an improvement over currently deployed Scalable Product Architecture (SPA) and it is called Digital SPA (dSPA). SPA provides a core platform which connects all of the electronics in a car. dSPA aims to connect different ecosystems Volvo Cars has and to monetize the data providing additional services.

The ambition for the dSPA is to create a new digital platform for cars. To be able to leverage on this goal, they need to organize the company’s data in a centralized global way and they need to respond to a service-oriented architecture in a modern way with micro services. dSPA will require a massive amount of architectural work to transition.

**Ongoing Changes on the Information Layer**

dSPA will not be released for a while. In the meantime, the organization has to address existing architectural issues. The contact person is the main responsible for a new information layer application which is based on the graph theory and developed with Sparx. The ultimate goal is to create meta-models that are modifiable in every possible way. It will be able to adapt itself according to different business needs.
The main architectural blocks are put together in the form of core nodes according to the graph theory [45]. The nodes are later extended into smaller ones. The new application is a step toward creating a centralized product cluster. It aims to connect different blocks at the higher level. The organization has major practical differences in different processes, even if these processes are quite similar. The new application enables architects to be able to reach information from the same source, thus increasing the standardization.

Nodes contain information about the person, but they are not tied to it. They are in the architecture level. Instead of having a consumer connected to a specific car/vehicle, there is simply a customer has a certain data, ID, and status. The division of nodes is inspired by relevant components of TOGAF. TOGAF is very big and not flexible. It is not practical to fill all of its components, so only the needed ones are used. It is important to note that ADM is not used as it is not flexible.

The contact person is trying to have a party data service while assigning people with IDs. The current system holds duplicate information about people about their roles. For example, if someone is an employee and customer at the same time, that person has two records. The duplicate is mainly caused by the conflict of CRM and internal information system. The nodes have three main concepts: the party (person), the role (employee, customer etc), and the standing (salary position, consumer role, type of the car etc). This eliminates the need for duplicates, makes the overall management simple, and also GDPR-friendly. One can uniquely identify people and simplify things with it.

This new application seems to be successful in increasing standardization in the technical side. Though, it is not working well for business and enterprise architecture which should be prioritized as an urgent issue. A new architecture needs to be developed preferably without disrupting the current one.

Going back to the architectural consequences, the new system allows HR and CRM to use the same infrastructure, even though they do not agree on every detail. They only need to agree on what a person is and how it is specified. Still, it can be problematic to present this in the business layer which might require another level to communicate with them. Right now, they have a conceptual view of the asset (simple with no extensions, it has enterprise and functional objects) and it is cross-functional.

The new Sparx application contains mostly the tools, annotations, and the structure. There is no new content. It simply is a new way of communicating in a holistic view. The application is not official yet, and support from other stakeholders is needed to make it more widespread. It provides extra knowledge transfer between architects, decreases the gap between agile teams & architecture thanks to a better communication medium and allows the business side to understand technical side better. If this value can be advertised within the company, the tool will be used more and will have more content over time, increasing the benefits.

13 GDPR: [https://www.eugdpr.org](https://www.eugdpr.org)
Grassroots movement is the motivator here. Letting people try the application by themselves and understand its value is more lasting as opposed to openly making them use it. The long-term goal is to have a platform where people can contribute with no prior education with a small effort. The beginning would be slow and take a long time since reaching the critical mess is reached. Though, a feedback loop would be created after a while which makes it continuous. This is a good way of being sustainable and avoiding the HIPPO culture.

**Capability Management Maturity Integration**

Different from other subsections in this section, the discussion about Capability Management Maturity Integration (CMMI) came up during a meeting with Ferrologic. CMMI is included in the study because it is considered a good starting point to give a direction to an organization.

In order to do that, CMMI defines the current architectural maturity level of an organization and tells what is missing. It is organized with five levels. Each maturity level includes a number of conditions to be fulfilled. An organization can move to a higher level if only they satisfy all the conditions in the lower levels. These conditions are architectural practices such as having a common repository, standardizing tools, ensuring business understanding of the technical side, clearly defining processes, certain architectural governance criteria etc.

Satisfying higher level requirements while lacking some lower level requirements does not increase the maturity level. This restriction makes CMMI a good candidate as a guide, because what needs to be done at any given time is clear.

**Summary**

The initial evaluation provided information about regarding the changes the company is going through. In the short term, they are working on a new tool which will unify the current information layer with a holistic view without creating new content. In the long term, Volvo Cars’ platform, SPA will be replaced with dSPA which is going to focus on the cloud. Here are the current challenges they are experiencing:

- High level management involvement has a huge impact on the EAM. For example, the number of information architects decreased over time due to managerial decisions which led to the present day issues.
- In order to avoid negative impact of the management, individual architects need to be empowered and gain increased accountability. Making sure that individuals have responsibility is crucial. This would also enable individual standing up change and improvement.
- Dependencies have very little documentation and person-dependent. Sometimes, dependencies are not discovered until that specific application is used.
- The company has a high amount of legacy apps. Managing them and their dependencies is a challenge.
• Architects do not have a common toolset. Also, the lack of knowledge transfer between them sometimes causes them to have different approaches to similar problems and to miss better practices. Developers who work in smaller agile teams can also benefit from standardization of the toolset.

• The current information layer is divided and the architectural repository is not regularly updated. Bringing the common repository to life would greatly increase architectural agility, meaning that the current architecture would be more robust to changes.

• There is discussion among some architects that standardization of tools and centralization of repository is not feasible, because it is too expensive and a high number of people needs to be educated about changes.

• CMMI determines the architectural maturity and it is useful for knowing where does the organization stand and providing guidance on what to do next.
5. Phase 2: Detailed Evaluation

Phase 1 has provided fundamentals about Volvo Cars issues and corresponding work in the literature. They are trying to rebuild the current information layer to provide unity among architects and align their business and technical sides better. Phase 2 aims to investigate characteristics that affect the current architecture better. In order to achieve that, detailed interviews with eight architects in Volvo Cars have been conducted. This section explains the details of these interviews.

5.1 Interview Questions

The questions in this phase intend to give interviewees freedom so that the results can be used to extract a number of EAM success factors. Because of this, the study follows the semi-structured interview format and interviewees were flexible in their answers.

The questions are chosen in a way that they will result in discussions related with the company’s current standing mentioned in Section 4.2 (Initial Evaluation) and connect it to existing architectural concepts mentioned in Section 4.1 (Literature Review). The findings in the literature can be either confirmed or conflicted depending on the answer. The interviewee’s experience on the enterprise architecture is also included to ensure the reliability of the results.

<table>
<thead>
<tr>
<th>#</th>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Please mention your personal background.</td>
</tr>
<tr>
<td>2</td>
<td>What is your experience with architecture modeling? Which specific tools did you use?</td>
</tr>
<tr>
<td>3</td>
<td>How would you differentiate different levels of architecture such as information, business and enterprise?</td>
</tr>
<tr>
<td>4</td>
<td>What comes into your mind when agility of the architecture is mentioned? Do you have a clear way of identifying it?</td>
</tr>
<tr>
<td>5</td>
<td>What are the common practices in go-to-market in terms of development, operations and testing?</td>
</tr>
<tr>
<td>6</td>
<td>How does your working methods change depending on if the customer is internal or external?</td>
</tr>
<tr>
<td>7</td>
<td>What do you usually do when a new requirement occurs?</td>
</tr>
<tr>
<td>8</td>
<td>How do you handle dependencies within the code?</td>
</tr>
<tr>
<td>9</td>
<td>How do you decide on different layers or blocks while coding?</td>
</tr>
<tr>
<td>10</td>
<td>How would you describe unsuccessful projects and their underlying reasons?</td>
</tr>
<tr>
<td>11</td>
<td>Do you know how other departments in your organization or others organizations operate?</td>
</tr>
</tbody>
</table>

Figure 6: Interview Questions
Please see Figure 6 for the list of questions. In general, the purpose is to be able to get a comparison between the previous findings and the interview results. Questions 3, 5 and 9 aim to understand participants’ standing on where does the architecture stand and how does it affect the end-product. Questions 2, 7, and 8 aim to understand the participants’ views on the topics revealed in the literature review, which are modeling, requirements and dependencies. Questions 6 and 11 highlights the communication. Agility of the architecture is found to be a rising issue in the literature. Questions 4 and 8 investigate on it. Answers to Question 3 and 4 reveals the interviewee’s views on the role of enterprise architecture. Similarly, answers to Question 5 and 10 can lead to discussions related to the ongoing changes in the company which is first discovered during the initial evaluation.

5.2 Demographics of Respondents

Interviewees have been contacted by the main contact person at Volvo Cars. Specifically, experienced architects who are familiar with the current circumstances are contacted for a 1-hour interview. The interviews might contain sensitive information regarding interviewees’ jobs, therefore their personal details will not be revealed. They will be referred to as I1, I2, I3, etc. for the corresponding Interview 1, 2, 3, etc. Please see Figure 7 for the list of the interviewees.

<table>
<thead>
<tr>
<th>Interviewee</th>
<th>Role</th>
<th>Interview Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>Program lead</td>
<td>In person</td>
</tr>
<tr>
<td>I2</td>
<td>IT/Business Architect</td>
<td>In person</td>
</tr>
<tr>
<td>I3</td>
<td>IT/Business Architect (consultant)</td>
<td>In person</td>
</tr>
<tr>
<td>I4</td>
<td>Solution Architect (consultant)</td>
<td>Video Call</td>
</tr>
<tr>
<td>I5</td>
<td>System Architect (consultant)</td>
<td>In person</td>
</tr>
<tr>
<td>I6</td>
<td>Information Architect</td>
<td>In person</td>
</tr>
<tr>
<td>I7</td>
<td>Head of Architecture Services</td>
<td>In person</td>
</tr>
<tr>
<td>I8</td>
<td>IT Director, Strategy Architecture Lead</td>
<td>In person</td>
</tr>
</tbody>
</table>

*Figure 7: Overview of the Interviews*
5.3 Common Findings

It is observed that many interviewees mentioned similar points. These points are presented in this section as a list to ease the reader’s job for Section 5.4 where interview highlights are presented. Figure 8 provides a list of common statements made by the interviewees with respect to how many interviewees mentioned that it. These statements are not necessarily the architectural success factors for Volvo Cars’. Some of them simply explain their way of working. The statements are included here as they help understanding why the company is experiencing their current issues. In Section 5.5, these statements are going to be beneficial for determining the success factors.

<table>
<thead>
<tr>
<th>#</th>
<th>Statement</th>
<th># of times mentioned</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>There is no common architectural repository.</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Tools used differ from project to project even on similar projects, and the organization does not have a standard set of tools.</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>Architects are often too technical and they do not understand business needs.</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>The current documentation on dependencies is project-dependent.</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>The organization is trying to have a new data integration hub where the higher level connection is available as a holistic picture.</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>The company is successful on the business side but on edge of the technical side. Business should realize this and give the technical side necessary support.</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>Having an increased accountability and trust in individuals would increase success.</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>There is a huge amount of legacy apps in the organization which will need to be subject to change with the technological developments.</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>The effects of separation from Volvo Group in the 90s is still visible in a negative way.</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>The amount of documentation is not adequate.</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>There is a lack of knowledge transfer within architects.</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>Architects’ role used to be governance in the past. Right now, there is a shift towards guidance.</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>‘Highest paid person’s opinion’ makes avoiding wrong decisions harder.</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>Outsourced personnel sometimes have a too narrow point of view.</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>Good exploration of the landscape is a very common success factor.</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>Key for agility is to understand innovation and how it happens.</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>Architects should understand how other companies operate.</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td>The new requirements demand a global network with more end users with more options for them.</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>The organization is working towards a new open source cloud platform integrated with the legacy apps which will help to establish agility.</td>
<td>2</td>
</tr>
<tr>
<td>20</td>
<td>Architecture agility can be established if projects start with minimal detail, and get more detailed gradually in a changeable way.</td>
<td>2</td>
</tr>
</tbody>
</table>
Scalability is a common concern. Products sometimes perform well in the beginning when the number of users is small. Though, problems occur as product become more widespread.

The business model is changing. There is a shift from product-centered to customer-centered (data-driven) approach.

The number of business and information architects in the organization is not adequate.

There is a lack of business drive to perform technical changes which often give a long-term profit with a cost of short-term loss.

Not knowing the landscape or the information structure makes the projects unsuccessful.

Increased collaboration and having more cross functional teams is needed.

Developers should automate every possible task and try to focus more on translating business needs into code.

Active involvement of decision-makers is the main success factor.

Underestimating the required amount of work as a common failure reason.

Making everything less person-dependent is crucial.

Architects on mid-level high-level solutions, developers on low-level solutions

Figure 8: Common findings of the semi-structured interviews

Some of the above statements are not self-explanatory and might mean different depending on the interpretation. To avoid confusion, they are explained below.

**Statement 1:** Historically, there used to be a common repository which still exists. Though, it is outdated and no longer used by many people. One of many reasons why it got outdated was because the cost to maintain it and is dependent on people. The maintenance cost is high due to the fact that there are different layers to deal with which need a lot of attention. Most departments use different systems for documenting applications and processes which becomes a problem when a holistic view is needed. Currently, there is a work towards a common information layer where everything is connected in the higher-level information layer. It is a meta model based which comes with many benefits as explained before.

**Statement 2:** Not having standardized tools come from a lack of architectural governance. Standardizing tools might seem too limiting, therefore an opposition to the agility. This is not the case though. In this case, removing individual flexibility of projects makes the overall structure more flexible, because it is more adaptable to unexpected change. Too much individual freedom decreases the knowledge accumulation, therefore, causing unsuccessful practices.

**Statement 3:** The core purpose of the IT should be to add business value. In order to establish this, there should be a level of business understanding and IT tools should be developed for the business’s benefits. Most architects have an engineering background which makes it difficult for them to understand business needs. The solution can be to give architects a couple of business-focused projects so that they can leverage themselves through these projects to the business level. This way they would see business layer needs better and also show architecture’s value to the business side.
Statement 6: The organization has a huge amount of legacy applications which still work and generate income. Though, it has a limited lifecycle remaining and needs to be replaced/updated. Business side should be aware of this and put necessary attention before it causes a loss to the company.

Statement 7: Individuals often want responsibility for tasks. Though, this is not followed by accountability. In case of an unexpected result, individuals are reluctant to be accountable for it. The feeling of ownership should be increased to overcome this.

Statement 12: A significant amount of architects still uses old EA frameworks what explains hierarchies and tries to govern them. With an upcoming fast-paced agile world, architecture should be flexible. Architects should design high-level concepts with big boxes. They should guide the teams do their job in these boxes without directly interfering. Architects should only interfere when different teams in different boxes overlap and help them negotiate.

Statement 13: To achieve success, the decisions should be based on consensus. This allows implementing agile methodology easier because consensus gives flexibility to everyone within their zone. This is what the HIPPO culture lacks the most. When decisions are based on one person’s opinion, it limits others’ actions. One way of avoiding this to move managers between different management roles occasionally, so that they experience different perspectives.

Statement 19: Volvo Cars’ old legacy system is designed for a traditional industry company. They need to transition to a new format to adjust themselves to modern requirements. Volvo Cars have planned to get partially open source in the future.
5.4 Interviews

Interview 1

Il is the program lead at dSPA. Il used to be a consultant with a developer origin and has been in Volvo Cars for 11 years. Il has different experiences in manufacturing, sales, and R&D.

His definition of an enterprise architect in an agile culture is a person who needs to be at the edge, sees what is coming next and make the company decide upon them. The old way of architecture which only says the right way to do things is gone. The new architect is a speaker and a motivator with a vision. Agile teams should have the freedom to choose what they want to choose. They only need to be steered.

Looking at how developers work with the product, they should focus on the things that are good at. Automating repetitive and not productive tasks is important. A typical example is testing. Developers should be focused on translating the business problems into code.

Communicating about dependencies has always been and will always be an issue. The hardest part is communicating the decisions and what has been done before. Developing centralized catalogs and making people publish their work is a good way of documenting. The level of detail does not need to be high, but the documents should contain the necessary information. Even though this will increase the architectural maturity over time, it is never going to be perfect. One reason is the size of the company which has multiple leadership dynamics.

Today, the way of solving dependencies is directly communicating with involved developers or people who have knowledge about it. The company tries to get away from this structure. Developers should register for using services so that the organization knows which applications are being used. This would ensure that the response team can estimate the capacity. Also, if team members change, they would know whom they should communicate with. Any extra work is a loss of manpower. Architects should increase the developing time, not the managing time. In the end, what makes value for the company is developing.

They handle requirements with agile tools, to be specific with SAFe\textsuperscript{14}. It ensures communication with the business owner in the sense that the developers are aware of the needs. The old way was the waterfall method\textsuperscript{15}.

The agile approach has benefits for the architects as well when we focus on its philosophy. To make architects embrace agile, they should be taught about it. It is hard because the extra business value of agile architecture is not directly visible, mainly because the business value of architecture is not directly visible either. Architects ensure that the room is clean so that there is a good foundation to start working in the first place. It is important for architects to have a couple of business-focused projects so that they can leverage themselves through that projects to the

\textsuperscript{14} Scaled Agile Framework (SAFe): https://wwwscaledagileframework.com

\textsuperscript{15} Waterfall Method: https://airbrake.io/blog/sdlc/waterfall-model
business level. This is how they can get support on a larger scale. They need to be the ones who make everyone understand why architectural tools are necessary.

Fixing faulty components after deployment has very situational solutions. Automatic software updates or manual fixes often solve the issue even before it becomes a problem. In the worst case scenario, this is not discovered until a system crash. Ideally, these faults can be fixed very easily if every team registers what they are developing or using. The second important thing is proactively monitoring and defining team roles clearly. The third point is about automating rollback operations.

Successful projects have four main attributes.

(1) Active involvement: Active involvement with the decision makers makes projects successful. Getting answers fast is crucial for the success of IT projects. This is exactly why the agile movement is so big. Feature driven tools are very important. Even while doing the waterfall, trying to borrow principles from agile is useful.

(2) Continuous improvement: Continuous improvement is also important thing even more important than having an active development team. Without continuous improvement, the application will turn into legacy very fast.

(3) Increased product owner responsibilities: Giving product owners both responsibility and accountability is another key factor. It is often the case that, they are responsible but not accountable. They should also be empowered their jobs.

(4) Consensus seeking: The final key factor is about trying to have a consensus. This makes implementing agile principles much simpler. This is why hierarchical cultures have a harder time adapting agile. HIPPO culture is another threat to success which is avoided with consensus seeking.

Different types of architects deal with different levels. Business architects have an understanding of the business models, processes and their connection with the people. They do not need to have an understanding of the actual IT landscape. Information architects should ensure that information is not diversified and the same thing should be observed even while looking from different aspects. Solution architects understand the IT landscape of the certain part of the company. Enterprise architects should have a broad understanding of the whole landscape. It does not have to go through the details but should be able to advocate management people level about how the full stack is operating. In some cases, enterprise architects become the HIPPO. This should not be the case and is not sustainable in the long term.

The power should be distributed more amongst architects in a way that no one should have a final say without reasoning. The frozen middle is also another problem. It is often the case that the people at the bottom who actually does the work and the top management who want to make things more efficient. Both of them wants to transform and change. Though, these changes tend to remove power from the middle management. Since the middle level has the most to lose, they become the biggest obstacle.
There is no single common active architecture repository in Volvo Cars. The current TOGAF-based repository mainly has legacy applications which is no longer updated. The reason why people stopped using the repository was lack of focus and lack of empowerment of the responsible architects.

The company has two different types of outsourcing: with consultants and with developers in a cheaper location. They have outsourced many architects, even the solution architects. This means that they have to create education packets to teach them about the internal work that has been going on. Because the education part was lacking, these architects could not find the needed parts in the repository until too late. Talking specifically about development, outsourcing slow-paced applications that require only maintenance is a good idea. The high-paced applications which require a level of development should not be outsourced. Outsourcing a whole project and cutting the internal connection is not a good idea either. Ideally, applications should stay internal as well. Only the teams can be outsourced in a way that they can be helpful to a product owner who works within the company.

As a concluding remark, I1 mentions that EA is a field where it is hard to find a common answer. Most of the EA frameworks are stuck in an old-fashioned way with hierarchies and they need a change. Agile is full of fast-paced development. Enterprise architects should be designing high-level boxes in the architecture but should allow free movement within the box to the teams. If two teams overlap, the enterprise architect should negotiate with the work division. We can define agile as doing things in small increments, understanding what you have done, restarting and starting over. Designing the architecture in 6 months only to realize your mistakes in 3 years after deployment is no longer an option. The perfect architecture will never be deployed. One needs to constantly analyze, re-plan and update the current structure.

**Interview 2**

I2 works as an IT/Business Architect for the HR and Financial domains in which they currently are transitioning into agile product teams. I2 has an EA background focusing on process and information aspects, specifically on the operating models, application portfolio, concept and information modeling. I2 also worked as a solution architect on BMS tools.

Deciding on which dedicated tool to use for the EA repository is a challenge which should be deciding on that if the end users are able to use it without a problem. If there is no a specific repository, then it is a problem. When I2 was working in a software company, I2 had the opportunity to work on the repository and he combined existing architecture into a single repository. It is rarely the case that companies have a single repository because they deal with different layers. I2’s department is an example of it, they have different systems for documenting applications and processes. Getting a holistic view is not possible, and it might be too expensive or complex to implement, but it is beneficial regardless.

---

The tools should be specialized on the tasks. Otherwise, they will not ease the tasks, thus making fewer people using them. Architects need to be aware of this. Architects need to sell their work and show people better ways of working. Specifically, for I2’s area (process information), the results need to be exported or reported somehow to the end users on the business side. This lets them gain some attention, show the value and get more funding. Architects need to be able to inform developers as well with expressing what kind of information different roles need and connect this to a specific system and process. Simplicity is a crucial factor in all of these. The blueprints should not be more complex than they supposed to be. Currently, they usually send these blueprints/documents directly instead of using a repository. They had issues with it since not every developer had knowledge about accessing the repository, and they were not able to get the blueprints. They are considering agile teams and trying to figure out how they can work better in small teams. Ideally, they should use the repository instead of a document. Practically, it is hard to teach everyone, so having one or two people in a team who can convey the message in the repository to the others might be the best option.

Even though I2 does not work with requirements or dependencies directly, there some known practices in the company about them. The requirements change continuously. The common way of managing them is trying to minimize the required changes. Regarding dependencies, the information is conveyed through direct communication. If the teams depend on documentation, they cannot be certain that the application works exactly the way it is described. This is because teams usually keep their own version of the documentation depending on their purpose.

I2 describes a good pre-study and unity within the project as a common success reason. Success is highly correlated with a good pre-study that explores requirements, business needs, and the current landscape. Project leader and business analyst have a crucial role in that. When it comes to applications and integrations, unity within the project is also important. The net success is determined by how much money you have saved. How success is defined depends on the perspective. A failure for the end user might be a success for the implementation team, considering the challenges they overcame. Success and failure are not black and white.

Underestimating the required amount of work is a major failure reason. I2 experienced an example working on a solution that would be effective globally in 30-40 different applications. They realized that some legacy systems would need to be decommissioned because of the new solution. This is not a failure but they certainly need to scope down the project depending on how much resources they have.

I2 sees agile and enterprise architecture as a good combination (smaller agile teams working with help of an architecture) but does not consider agility of the architecture. Architecture is solid and often not changed unless it is needed. If that is the case, the change is often tailored for manually for that case. Architects are usually involved with agile teams during initial design, and teams consult them about the way they are working. He thinks that it would be beneficial if the architects are more involved in the implementation phase. Solution and domain architects can be helpful with that. Currently, architects are more involved with the initial design and documentation.

Evaluating EA alone, the company face two main challenges, a lack of common methodology and having a huge amount of legacy. Both architects and some business analysts are involved
with architectural work in large companies. They should identify the processes and the requirements. An adaptation of agile principles within the architects might be a good practice. This can be used while defining different layers and their interaction. The second issue is about legacy applications. This creates a difficulty getting the bigger picture because it is hard to get information about legacy applications. They often need to start over and try to identify the key points. Having an increased communication among different architects and flexibility during decision making are needed.

Looking from agile software methodology in the company, agile teams should have access to different types of architects (integration, solution, process architects etc) to succeed. It is often the case that they have access to mainly technical-focused architects, this does not give flexibility in case of a change. They need to have a feedback culture as well.

**Interview 3**

I3 works on the same project with I2 as a consultant from an external company. He is mainly responsible for Java-based architecture, customer-premises equipment, and huge scale integration projects. I3 has more than 20 years of experience and started his career as a software architect. I3 was heavily involved in IASA\(^\text{17}\) and took place defining software architect certificate requirements in Sweden.

The way technical architectural layers are categorized has changed recently with the emergence of full stack developers. The separation between frontend and backend become less visible. Application layer became larger. What they develop communicates with information and business layers. Developers do not label these layers to avoid confusion. During the design phase, they only need to consider where the application is running. The interface should be decided depending on accessibility and the end user.

Agility is about flexibility to reach out different functionalities. Creating models from different points require a good interface which is one of the most important factors for agility. This is also beneficial while presenting to the stakeholders. Agile architecture requires distributed agile teams working with a one common logic base which provides the people around a developer to use that developer’s services without a hassle. I3 mentions that DevOps is in an emerging state in his department. It provides frequent builds and their results while boosting software agility. Increased agility in the software level allows the architectural level to be more flexible in their decisions as well. The bulk of the projects they have is traditional where there is an agile team. Some welcome the changes with DevOps, some resist. The attitude towards change is one of the main challenges.

Having a standardization of computer languages/tools is another requirement for agility. For example, they let developers know which languages would provide the best practice. As soon as they start working, they also conform to enterprise architecture principles. How each team chooses to do their respective applications is up to them as long as they stick to the core framework. Standardization cuts off an extra unnecessary search phase.

\(^{17}\)International Association of Software Architects (IASA): [https://iasaglobal.org](https://iasaglobal.org)
Dependencies also an important issue for Volvo Cars mainly because they are working on a physical product. It is common that different vendors produce different parts of the car and they need to ensure that everything is compatible with each other.

How these dependencies handled varies a lot depending on the case. They are using two different tools for the application directory (Alfabet\textsuperscript{18}) and integration directory. There is a connection between these two directories and it becomes problematic if one of the tools are located on separate directories. They also realized some of the information in these directories are not actually true and outdated. At the time being, there is no clear separation about which one they should rely on. The application manager needs to be contacted directly in case an information is missing. A project might have up to 50 integrations and if they need to contact the manager for most of them, it becomes very time consuming to handle dependencies. Decommissioning old applications are also necessary when new application override or make obsolete the old ones.

Some applications have been used for a very long period of time. The oldest one has been running since 1978. In the beginning, legacy applications were not an issue. As the number of applications grew, different directories were deployed because of different technological needs.

Half documentation is another issue. It is a case that teams document the beginning of the project, but they stop after some point which leaves the documentation incomplete. There has been no governance regarding centralizing the information description. Documenting the old applications is very expensive. The most feasible way is to start documenting them as soon as possible in an agile way. In this context, agile refers to being fast and documenting only the relevant parts initially. Not used applications which make up 30% of the repository do not need to be documented. To sustain the system, developers should be accountable of documenting their own applications.

Most of their projects use old DITS (developing IT systems), model. It is an internal Volvo Cars development methodology. It has agile components, but it is not fully agile in terms of the way of building. It is not possible to fully implement agile software development principles in the automotive industry where one deals with production. A physical object is being produced and everything around the components should fit each other. It is not possible to be flexible after some level. DITS includes a number of testing processes which is used for a period after production. Issues are taken care of by the development team during this phase. Operations teams are responsible after this.

There are ways to mitigate software errors. Most errors occur after an update, which is when the system is downgraded immediately. The first thing to try is changing data models required for the project. If this does not solve the issue, the code has to be modified or rewritten in the worst case. This heavily depends on the case. Usually, the fixes end up with a trilateral conversation between the business analyst, the architect and the team. It should be noted that technological limitation is never an issue.

\textsuperscript{18} Alfabet: https://www.softwareaggov.com/products/business-it-transformation/alfabet-enterprise-architecture-management/
The biggest EA challenge they are facing is architectural governance. There are 50-60 solution architects who are working in an individual level. Their solutions are personal and there is not much communication going on between architects. Because of this, architects come up with the same solution over and over again without even realizing. Sometimes, they have opposing solutions which inhibit the progress. Having common tools and frameworks would be helpful for maintenance and scalability. Dialogues and open meetings are the best way to solve this.

**Interview 4**

I4 is an external consultant working with HR systems with more than 25 years of experience. He is experienced with SAP, SAP Success factors and SAP integration solutions. I4’s task is to replace Volvo’s HR and management systems with SAS solutions. Since I4 is located outside Sweden, the interview is conducted via video call.

To differentiate between different levels, they determine the building blocks and the information they contain (employee data, financial data, organizational data etc.) and focus on data models. Communicating this information with the others is also an issue. It is important to consider how this information will be used by the target and understand both IT and business aspects.

Standardization is crucial for agility. Especially with the case with Volvo, one needs to consider the global aspect and standardize the models across different locations to increase their control. Removing individual flexibility might seem like the opposite of agile, but it actually makes the overall structure more flexible due to increased knowledge. The only difference they have legal and statutory differences between different countries.

Requirements might change during development. If the change is minor, it is fixed during the regular procedure. If the change is something that requires a chain of changes, this might be time-consuming and costly. The worst case is when the requirement changes need the whole project to be resigned, which almost never happens. Generally speaking, they have templates for different cases such as integration, migration, development and they are able to follow it most of the time.

Architects and developers operate on a different level. The architects are more around the overall medium to high-level solutions. They do not need to have a detailed understanding of programming, the limitations of the solution, what is possible or what is not. Developers need to have low-level programming skills. They do not need to know business processes.

Architects help to transform business requirements into technical ones. They have solution architects who are responsible for the overall solution. They are involved in integration and design. They have technical architects who work on technology around the application. Data architects work on the overall management. These people try to translate the business requirements into developers.

Cloud solutions and SAS solutions are handy to make dependencies less person-dependent. Everything is documented. Teams have access to different APIs. If they need a specific functionality developed in an API, this has to go through a development request process. The team cannot modify the API on-premise and use it for their specific needs. If something is not
working as documented as it is supposed to be, the changes have to go through a release process as well.

SAS enables testing and detecting errors. SAS documentation allows teams to know about what changes are coming in the next couple of weeks. I4 gives an example of SAP’s SAS factor inspector. Each quarter, a new release is published. If there is a change in any data structure, API or technologies used, they have a review about it and test it. In case the problems are not detected during this cycle, they do not have a real control over the issue. If it comes to this, they would need some custom approach.

Reusability of the solutions highly increased especially after APIs and SAS solutions are standardized. They often had SAP and ERP presentations where solutions were heavily customized with very small reusability. It is very good that they no longer have to keep reinventing. This also made documentation more efficient and let it be used by more people. Now that software a solution is available, projects are more flexible and less person-dependent.

Before concluding, I4 mentions the importance of having accurate expectations on projects. Sometimes, the solutions are oversold in terms of what they provide. The solutions might be very good from the architectural point of view, but it is not really successful if people are not using it. Adopting to the stakeholders’ needs and being able to inform them is crucial. To be able to do that, one should have contact with other people and learn what they are doing.

**Interview 5**

I5 is working as a consultant in Volvo Cars, mainly involved with designing the requirements in the high-level architecture regardless of their technical backgrounds. I5 has been working in this field for over 25 years with a background from software development.

Agile architecture means two things for I5. First, there is no need to have a detailed end solution in the beginning. What is important is to have a fair idea about the direction. Second, the architecture needs to be set up in a way that it is changeable. One should be able to modify the architecture depending on the current needs.

There are three historical events which are impactful on the current situation of the company. The first one is Volvo Cars getting separated from Volvo Group during the late 90s when Ford purchased them. There were a competency and information loss due to the people not appreciating the change and not staying at Volvo Cars. The second one is 11 years later when Ford sold Volvo Cars to Geely after realizing the cooperation is not working efficiently. The company experienced a second wave of knowledge loss this time. Data needs to be transformed accordingly depending on the application area. The company is experiencing hardship mainly related to transforming data because the people who worked with it were lost during these acquisitions. The final one was the big data challenge. The company had to had to deal with it less than 10 years after its acquisition by Geely and the shaky situation of data management did not help. A rigid legacy from a technical perspective and misalignment across the data entities are the two main challenges.
Data is something which is often times overlooked and designed in the final phases. Information layer proceeds KPIs, business process layer, and application services. This should be the opposite. Conceptual data models are crucial for a company of this size to be aligned within itself.

Volvo Cars is successful on the customer-side even with all these challenges. The technical situation is challenging and not perfect, but it is working. Overcoming these challenges and having a healthy architectural structure would let the company operate even more efficient. They need to find a way to use the legacy in a good way before IT becomes a bigger bottleneck for the company.

The future of the legacy apps is not so clear. They are important for the company as they still generate an important amount of income. There are chains of integrations. Multiple applications are dependent on each other. Not having a clear documentation and knowledge about these dependencies make the whole development process harder. It is sometimes the case that the dependencies are not fully known even if the application runs with no problem. Data is nowhere, everywhere and somewhere across the landscape.

The dependencies somehow need to get documented. One thing that adds up to this complexity is an important amount of applications run on premise. When they were designed, Volvo Cars was not a global company. Now, the company is operating in many different locations and the applications need to be modified accordingly. They should be easily accessible to everyone working at Volvo Cars, regardless of location. Scalability is also of an issue here. The system needs to be changed in a way that the future changes would be less cumbersome. The third factor is time-to-market. These changes should be performed rapidly and be ready to release as soon as possible. This should be in an agile fashion.

Currently, one needs to have a good understanding of what kind of data they need in order to be able to access it. They need to learn this from other stakeholders which sometimes makes the process problematic. The type of access needed is also a factor here. Changing a piece of data is much harder than only reading it. To whom the data should be provided is another issue as well. There are different access levels and not everyone is able to access everything depending on ERP, CRM, product lifecycle management applications etc. Finally, the issue is about the age of the application. Getting info about legacy apps is very hard, simply because the people who wrote the code is retired. The system should not be person-dependent.

Changing business model has also an impact. It used to be the case that Volvo Cars, did not use to get involved in retail and aftersales at this level. Now, they are more involved and the technical infrastructure needs to be modified accordingly. Volvo Cars used to be product-centered (product-driven) and still continue to operate in a similar manner. The applications should be centralized on the customer instead and they should be data-driven.

They need to have a globally distributed application platform which includes repositories, runtime environment, micro services and all the relevant components. It should directly interface with customers through ERP, CRM etc. Volvo Cars have already been working on this and want to bring it to life in the following years. Enterprise view should form a bridge between the legacy
applications and the global application platform. This enterprise architecture platform should have data management, data integration hub, and data distribution capabilities.

Architectural work in the company faces many challenges. Most architects in the company mainly work with application and technology layer, and their point of view is much closer to engineering. They do lack business and information architects who can act as a translator between them and the upper-level management. Another architectural challenge is the driver. Architects work with “stow pipes”, meaning that they operate individually and they have own solutions to the problems. There is a lack of knowledge transfer and this makes architects not see the bigger picture. Looking at the business stakeholders, they look at the short-term benefits. Business stakeholders in the middle level and project managers have been solving urgent business problems one at a time. Though, they are not interested in paying for the greater good of the company. Having these architectural changes are expensive. Their immediate payoff is rather low, but the eventual payoff is massive. Not having them will harm the company. Explaining the consequences of having a poor data quality to the business community is very difficult. The high-level management needs to be convinced that making the current system will global will only scale up the complexity which will make it very costly to handle. Having an architectural bridge between the legacy apps and the global application would let to decouple the legacy complexity.

The automotive industry is going through a massive big data change and companies will need an architecture to let them recover fast in order to survive. It is common to work on projects with no clear outcome, and the architecture should be able to see if it is failing and compensate for it. Architecture should also increase accessibility to data and decrease integration issues, which are two major failure reasons.

Accessibility problem will be greatly reduced by the data integration hub which gives an enterprise view on not only the master data but all the data. Master data management is not enough simply because it only includes the data which are classified as “master”. Traditionally, other kinds of data (for example sales data) would have been stored in different places (for example CRM) which is not the most efficient case.

Architecture should have an agile mindset with a structured view on the topic. Creating some meta models is a good approach to being structured while not being too constrictive. Providing different interfaces depending on the stakeholders is important. Business side becomes easily scared by models (ArchiMate, boxes etc.). This is where a lot of IT persons historically failed with presenting the business architecture in an IT-like presentation. The higher up you coming into the organization the more you need to simplify.
Interview 6

Working as an information architect, I6 focuses on what the data means for the business and provide value for the business. I6’s task is to create a common view of information which includes analytics and BI tools. The implementation is done with Erwin19, Teradata20, and Sparx.

I6 defines architecture as a framework which connects everything in the high level. It is developed continuously but should not move too fast. Agility is implemented in architecture on the practical level. It impacts how decisions are made. The higher level is more related with the architecture whereas the lower level with all the details is more affected by agile methodologies.

There is a couple of architectural issues in the comp. I6 mentions architects working in “stow pipes” an agrees on I5’s points on this. The fact that dSPA will use micro services to help with this issue. Each project will be able to create their own micro services for their own needs and reuse the existing ones. This is a step to avoid standalone applications that only support one business function. The new approach will decrease the distance between applications created by different departments and increase reusability.

I6 have used DITS similar to I3. Though, I6 is more experienced with the waterfall side of DITS, which makes I6 think negatively about it. Developers work with agile but DITS is quite opposing to this. DITS is also not suitable for getting business support. Every subject matter expert fit in the key positions within the business and they are not happy with the tools. They usually end up creating a documentation for their own understanding and this is not fed back to some sort of structure capital where this documentation can be used by everyone. I6 believes that this issue will be solved thanks to the information layer tool the contact person is working on.

Three major failure reasons are lack of landscape discovery, scalability and lack of ownership feeling. The surrounding conditions should be discovered very carefully even before the project starts. There are lots of projects which are initially successful but experience problems scaling afterward. I6 gives an example of the accessory webshop. When they opened the shop, everything was working smoothly, though, someone had to manually enter the numbers in the background. This caused them to spend too many resources in the long term, limiting the scalability. This was not maintainable. Apart from that, it becomes problematic to fix issues when the team does not feel ownership for the product. This is exactly why teams need to work with a business architect who is aligned with an information architect.

19 Erwin: [http://erwin.com/bookshelf/9.7.00/Bookshelf Files/PDF/erwin_Overview.pdf](http://erwin.com/bookshelf/9.7.00/Bookshelf Files/PDF/erwin_Overview.pdf)
20 Teradata: [https://www.techopedia.com/definition/25987/teradata](https://www.techopedia.com/definition/25987/teradata)
Interview 7

I7 has been working at Volvo Cars for a year as the head of architecture services. In the past, I7 is experienced with integration development, middleware integrations, software as a service solution and SAP. Before working as an employee at Volvo Cars, I7 worked as an external consultant with program architect role on their first China plant.

In the past, companies used their own EA grouping where they had frameworks, principles, guidelines, and responsibilities. Back then, an IT-driven architecture works which is not the case anymore. Their business has become more mature. They want to get more involved in what to use and how to use things. This is one of the main reasons he was brought in to modernize their architectural approaches. Instead of bottom-up, they try to have a top-down approach and boost the importance of information, business and process architecture. Business side does not need to decide on applications and systems, instead, they should be dealing with the consequences of them and communicating technical level about their needs. When business side gets involved with applications and systems, it also creates another issue in the sense that their modifications end up creating architectural problems later on. Being able to transfer IT knowledge to the business side in a non-technical manner and let them understand how their decisions affect IT is very important. One problem in the past was that some project managers who used architects only for documentation which did not add a value. Now, the organization is trying to use the benefits of working with architects both from the business and IT perspective.

Agile architecture is an open-minded flexible architecture with a good governance perspective to let stakeholders follow the same path. The struggle is about how to establish this good governance perspective. I7 believes that the way is business driven and it should include all the stakeholders.

An issue is that the company do not have one common architectural repository. This is because of the rapid growth of the company. No one took the time to set up common guidelines in the process. I7 is trying to fix this issue by lifting everyone up from the IT and infrastructure layer to information and process level. Together with the business, they need to set the information layer so that stakeholders have access to necessary data easier. It is very important to include the dependencies between layers once a common repository is established. They need to be able to see how does everything fit together from a specific domain perspective and higher level Volvo Cars perspective. It should be detailed enough to be able to be used as a template while setting up a new factory.

Communication and including everyone is very important to achieve these changes and I7 is working towards it. For example, their domain architects, who are responsible for different domains such as manufacturing, finance, HR etc, were not involved with the governance review model before. They are now involved from the early stages. They have been trying to let the business side know how beneficial it is to use architects in the early stages. Architects should have more collaboration. No one is in a single domain, simply because of the dependencies between them. The collaboration within the team should be very high so that everyone gets informed.
They should be more flexible about when to bring new ideas and when they need to terminate an existing solution. Today, it is the case that they are not terminating legacy solutions even though they have a better alternative. In the end, their legacy ends up growing much bigger than it supposed to be and becomes unmanageable.

The toolset should be more standardized. From a technological point of view, they should utilize cloud more often. Cloud solutions have been planned for a quite long time in Volvo Cars. The challenge is that they need to gradually decommission their old legacy applications as they move to the cloud. The transition process creates the main challenge here. The old legacy applications work really well but they are close to their end of lives. It will not possible to use the same application in five years from now. The shift to the cloud will cause changes in both the business and technical sides. They need to be flexible in their mindsets which puts agility in a very important position. One needs to challenge existing the current architecture to move forward. At the same time, this challenging process should be in a certain amount of governance to make it productive.

From a financial point of view, Volvo Cars is doing very good and growing fast. Their business model is changing. Instead of having one large company, Volvo Cars now have a number of sister companies and joint ventures who are owned by them. This increases the speed of business projects and programs. They are the ones driving the architecture. Developing and being able to reuse them from a business perspective is important.

Volvo Cars are using a lot of consultants who focus on projects and programs. These consultants don’t have a broad perspective on what is good for Volvo Cars. I7 wants to change this way of working and give consultants give a wider perspective. The only troubling thing is that guidelines and principles are not set by Volvo Cars clear enough. Setting these guidelines should employee-driven. The higher level needs to receive the information from everyone to get a clear picture. They are doing that by giving their domain architects a focus on both the business side and consultants.

Buying an IT solution from outside the company is common. The reason is mainly because of the belief that IT development is not fast enough and time-to-market is valued more. This is not a desirable solution as the maintenance becomes problematic. The people in the company has little knowledge of this solution which often times do not follow the principles and the architecture. This is why they started a new governance forum that aims to get in touch with relevant stakeholders before buying an IT solution to ensure that it fits the organization.

It is a common misconception that when someone mentions the architecture, usually system and application landscape is mentioned, and the business and information layers are missing. Another important function of architecture is to plan the transition from as-is to to-be, which is often neglected.
Interview 8

I8 has been working with Volvo brand for around 35 years and worked with more than 30 programming languages. He started his career as a mainframe coder, moved towards product responsible and shifted to only enterprise architecture for the past 15 years. The architectural group in Volvo Cars did not exist before he started working. I8 is one of the people who established the department. The department now consists of architects as well as external consultants. The department is rapidly expanding and I8 has been the main responsible for it.

If architecture is not agile, one is never able to handle disruptive ideas. The question is about how to become more agile at an enterprise level. One needs to get involved in tasks that require a change to understand the mechanics. For example, I8 got involved with the innovation team and trying to make sure that people who are working at the architecture level are involved or at least informed about the innovation team’s process. People working in a different way should not be changed but instead evaluated. New ways of working should be discovered. It is very common for big industrial companies in Sweden to have same architects working in the same department for a long time, thus making them working in old-fashioned ways. Architectural positions should give guidance instead of governance. The old-fashioned way was the exact opposite.

Most architects have an engineering background. They are very talented about understanding technical details, but not so much about business processes and needs. Architects also need to know about how external companies operate. If the other companies are performing some changes, it is only a matter of time that you will be doing the same.

The needs come from the business. Information architecture makes it visible to the more technical layer through process descriptions and access to data. Using this information, the required applications are determined. The next step is to design the infrastructure that supports these applications. Enterprise architecture deals with all these layers as a whole. Solution architecture is similar but it evaluates the landscape only in terms of a specific solution.

I8 stresses the importance of the split of Volvo Cars from Volvo Group and agrees I5’s points on the history of Volvo Cars. People are still able to see the impacts of Ford and Geely acquisitions. I8 explains what happened in detail. The old Volvo Group had separate sub companies such as Volvo Cars, Volvo Bus, and Volvo Data. It had separate business units in each different branch. For example, Volvo Cars and Volvo Bus had their own separate IT department. The interviewee belonged to Volvo Data which was the principle of centralization. They dealt with the information within Volvo Group. Due to the increasing importance of IT, it was decided to create a new Volvo IT that combines all the IT departments. The employees were insourced overnight and Volvo IT with around 2000 employees created in the 90s. Shortly after that, Volvo Cars was sold to Ford. Now, they almost had no people with IT skills. The infrastructure of Ford was not the most compatible with Volvo Cars. Volvo Cars ended up getting help from Volvo Group with hiring them on hourly-basis. The interviewee was one of the people who was working at Volvo Group and sent to Volvo Cars for a limited time. The company suffered a lot at the time being.

Getting help from Volvo Group for some applications is not something negative. They still have some applications running which were developed before the separation which Volvo Group helps
on occasion. This is just like buying an external solution. There is no point in reinventing the wheel.

An important part of the current application portfolio is old. There are even some applications from the late 70s while there are more from 90s. Most of them are built in the notion that the company is a traditional industry company with a data center and users in the same location. The important part was to secure its own perimeter in closed walls. Most of the legacy apps have limited authentication and authorization system. They depend on the core network. Now, the core network has a number of holes to communicate with the external system. They also have an undefined extended network which consists of different locations (data centers) of Volvo Cars. On top of this, the overall system is communicating with places other than Volvo Cars. The company needs to ensure that the legacy applications are safe and functional. They also need to plan how to replace them once the lifetime is over.

The old way of working which is mainly about connecting a central database to provide a service has rapidly changed in the recent years and Volvo Cars have a huge legacy infrastructure to take care of. The new requirements demand more end users being able to connect to the system by giving them the opportunity to modify their own information such as address and other personal details. This service should be available 24 hours and should have ideally no downtime.

I8 explains the new cloud dSPA in detail which was previously mentioned by I1, I6, and I7. dSPA makes Volvo Cars able to meet the new requirements. It will be cloud and not dependent on their data center. Though, not every detail is finalized. Some believe cloud-only is the best option whereas the others support cloud-first. The interviewee is the latter. I8 believes this would make integration of the legacy apps better while keeping sensitive data local. The cloud part is aimed to be open-source so that the community can contribute. Letting this happen would make the architecture more agile because it lets other people contribute meaning an increased tolerance to changes. When someone designs the architecture by himself/herself, this creates a lock-in effect. The designer would know how it works and the knowledge transfer to new people becomes harder. This comes with a plus of having a very secure system. In case an open source community is established, the system should be based on open standards to minimize the security issues. The second argument against the open source is the decreased governance. The interviewee believes that the governance does not decrease but changes its format. It becomes more important to describe the guidance promptly and making it mandatory in some cases. For example, there should be a type of handling identities. Otherwise, it will become expensive.

The cloud platform is a matter of survival. The current systems have around 3 years of lifecycle left and something has to be done before too late. The company is in a good financial shape but the IT is on the edge from an architectural point of view. In the mind of the business, it is not desired to fix something which is not broken. This is why they are reluctant about the changes. From the IT side, some people are afraid to make changes because they think it will affect the business negatively.

They have two major IT workloads: developing and maintaining. The changes will increase the development cost in the short term while decreasing the maintenance cost in the long term. The net gain is positive, creating room for resources that can be spent on different things. This means that the overall agility increases because they can spend more time and money on building new
things, which equates to more flexibility. Agility is a whole. You cannot be agile on the product level if you cannot be agile in the enterprise level.

Instead of trying to get the final product in the beginning, trying to get the minimum viable product and then developing is gradually till all the requirements match usually leads to success. This approach allows one to deliver a product even when not all requirements are feasible to match. Starting small and extending the functionality over time while not pressing all the functionalities in a single release is a good application of agility.

When it comes to architecture, it is very easy to focus on details too soon, mainly because it is easier than the alternatives. Instead, one needs to have a holistic approach and make sure that the people around have a complete picture of the situation. Coming up with simple solutions to complex problems is what architectures need to do.
5.5 Determining Success Factors

The common findings in Section 5.3 give a good starting point to determine what are the main success factors specific to Volvo Cars. In order to answer the first research question, these are combined with the findings in Phase 1 and twelve success factors are derived. The resulting list is presented in Figure 9. Please note that the order below does not indicate the factors’ significance.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Success Factor</th>
<th>Initial Evaluation</th>
<th>Literature Review</th>
<th>Interview Statements (from Fig. 8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>High-level management involvement</td>
<td>x</td>
<td>x</td>
<td>5, 13, 22, 23, 28</td>
</tr>
<tr>
<td>F2</td>
<td>Mid-level management involvement</td>
<td></td>
<td>x</td>
<td>5, 22, 28</td>
</tr>
<tr>
<td>F3</td>
<td>Software dependency management maturity</td>
<td>x</td>
<td>x</td>
<td>4, 8, 19</td>
</tr>
<tr>
<td>F4</td>
<td>Having a defined architectural maturity (CMMI)</td>
<td>x</td>
<td></td>
<td>1, 2, 4, 5</td>
</tr>
<tr>
<td>F5</td>
<td>Clearly defining individual responsibilities</td>
<td>x</td>
<td>x</td>
<td>7, 20, 26</td>
</tr>
<tr>
<td>F6</td>
<td>Using standardized tools within the company</td>
<td>x</td>
<td>x</td>
<td>2, 5, 10, 11, 30</td>
</tr>
<tr>
<td>F7</td>
<td>Having a common architectural repository</td>
<td>x</td>
<td>x</td>
<td>1, 5, 11, 30</td>
</tr>
<tr>
<td>F8</td>
<td>Outsourcing level</td>
<td></td>
<td></td>
<td>7, 14, 19, 30</td>
</tr>
<tr>
<td>F9</td>
<td>Business alignment of the technical side</td>
<td>x</td>
<td>x</td>
<td>3, 5, 19</td>
</tr>
<tr>
<td>F10</td>
<td>Technical understanding of the business side</td>
<td></td>
<td>x</td>
<td>3, 6, 24</td>
</tr>
<tr>
<td>F11</td>
<td>Requirements definition and handling its changes</td>
<td></td>
<td>x</td>
<td>15, 18, 19, 21, 22, 25, 29</td>
</tr>
<tr>
<td>F12</td>
<td>Cross functionality</td>
<td></td>
<td>x</td>
<td>17, 18, 19, 22, 25, 26</td>
</tr>
</tbody>
</table>

Figure 9: The list of the success factors

Most of the factors in the list are selected to be there if they are mentioned in the initial evaluation, the literature review, and the interviews. Though, there are some exceptions. For example, F2, F10, and F12 were not brought up during the initial evaluation. It is added after the literature review and the interviews. On the opposing side, F4 is on the list mainly because it was mentioned during the meeting with Ferrologic. F8 was not seen as a success factor until it was mentioned multiple times during the interviews. Since the purpose is to extract the factors
significant for Volvo Cars, it is included. Note that this does not imply that F8 does not exist in the literature. It was simply not chosen as a major success factor initially.

Here is the detailed information regarding the EAM success factors.

- **High-Level Management Involvement**

  Opinions vary about if the mid or high-level management inhibits the change. The HIPPO (highest paid person’s opinion) culture becomes problematic if it disregards the opinions of the others and limits their flexibility.

- **Mid-Level Management Involvement**

  It is sometimes the case that high-level management speeds up the positive change and this is supported by the developers or the people who do the actual tasks. Though, if these changes shift the power of middle-level management, it makes them reluctant.

- **Software Dependency Management Maturity**

  As explained before, agility is highly related to how dependencies are managed. It affects the new applications that are being developed, the maintenance of the existing applications, and retirement of legacy applications. It is very common to have a bottleneck if the teams spend too many sources on handling dependencies

- **Having a Defined Architectural Maturity (CMMI)**

  Capability Maturity Model Integration (CMMI) gives a good indication of where the organization is standing. It is very easy to apply. Though, the stakeholders sometimes might not be willing to use it, because it also questions their own decisions.

  Please note that EA maturity is a widespread topic and most of the factors are related to it. For the scope of the survey, the aim is to investigate if measuring the maturity level is impactful on the architecture agility.

- **Clearly Defining Individual Responsibilities**

  Responsibility and power not always accompanied by accountability. Increasing responsibility of individuals and making them accountable for certain tasks might make these tasks run smoother, as well as boosting agility of the architecture. This factor is related to EA governance.

- **Toolset Standardization Level (using standardized tools within the company)**
If the toolset is standardized within the company, it would create knowledge accumulation on some tools and allows greater flexibility in operating with them. At the same time, it cuts off the unnecessary step of searching for appropriate tools and learning them in each different project.

- **EA Repository Centralization Level (having a common architectural repository)**

  Being one of the core steps of CMMI, some organization do not have a centralized common architectural repository. The underlying reasons should be investigated. Once the repository is centralized, it should avoid duplicates, classify legacy applications and show the dependencies.

- **Outsourcing Level**

  Having some tasks done by external people have become common. The outsourcing can be in the form of consultancy/service of another company or simply outsourcing employees to a different physical location. It’s impact and how to handle it should be investigated. While it might be good to outsource noncritical tasks, the others might need to stay internal.

- **Business Alignment of the Technical Side**

  If the technical side is aware of what is needed from a business perspective, they would be able to balance out metrics such as project cost, time-to-market and requirement compliance.

- **Technical Understanding of the Business Side**

  Business side knowing about the technical aspect allows them to have realistic expectations from technical teams and thus to be able to have a more feasible business plan. At the same time, business side interfering IT might cause unexpected changes in the architecture and be harmful.

- **Requirements Definition and Handling Its Changes**

  Understanding the needs before the projects begin is another key success factors. The requirements should be discovered, the workload should be estimated properly and the business needs to be known so that changes in requirements are predictable.

- **Cross Functionality**

  There are other disciplines on top of agile and EA. For example, some project still uses Waterfall while not utilizing agile. DevOps is a growing discipline which Volvo Cars wants to deploy. Having different functionalities within a team also highly impact agility.
6. Phase 3: Investigating Success Factors

In the previous sections, data gained through three different ways have been explained. Namely, the author gathered data from the literature review, had meetings with the main contact people and conducted semi-structured interviews with architects. Afterward, which architectural success factors these data point to is evaluated. The next step is to investigate which of these factors are more impactful on the overall success of the EAM. The success in this context refers to the compliance with the management’s wishes. Section 6.1 designs a survey which is going to be used to rank these factors. This survey is sent out to professionals in the industry and Section 6.2 explains their demographics. In the end, the results of the survey are presented in Section 6.3 and 6.4.

6.1 Survey Design

Survey participants are asked to choose the importance level of the twelve factors in Section 5.5. The aim is to measure these factors in a quantifiable manner. To achieve that, the survey has five options which denote the importance level of each factor. The options are ‘not important’, ‘slightly important’, ‘moderately important’, ‘important’ and ‘very important’ which are later converted into a number between 0 and 1 according to Figure 10. These numbers are used for analysis purposes in the following steps. The participants are asked to answer all the questions.

It is considered that the ordering of the factors might create a bias effect and might change the participant’s approach to the survey depending on the first shown factors. To avoid this, the order of the questions is randomized for each.

<table>
<thead>
<tr>
<th>Survey Option</th>
<th>Corresponding Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Important</td>
<td>0</td>
</tr>
<tr>
<td>Slightly Important</td>
<td>0.25</td>
</tr>
<tr>
<td>Moderately Important</td>
<td>0.50</td>
</tr>
<tr>
<td>Important</td>
<td>0.75</td>
</tr>
<tr>
<td>Very Important</td>
<td>1.00</td>
</tr>
</tbody>
</table>

*Figure 10: Survey options and their corresponding values*

The choice of survey options was not random. An initial survey with more detailed feedback option was sent to 14 participants, 3 of whom responded. They were asked to rank factors in a way that they choose 1st, 2nd, 3rd etc. most impactful factors. This meant that they had 12 options for each factor and they were forced to decide which factors are more important. However, the feedback was that some factors equally important to them. Also, another feedback was that asking ranking for 12 factors is too complicated. Another possible drawback which was discovered later on was that having factors ranked does not necessarily mean that the participant
values that specific factors on the same linear scale. The participants might almost identically value 1st and 2nd factors whereas the 3rd ranked factor might be much less valuable. Because of these reasons, a simple 5-option survey is chosen instead of a 12-option one.

The survey questions are available at Appendix 1.

6.2 Demographics of the Respondents

The survey has been sent to 325 industry experts through online channels. 51 of them responded positively and participated in the study. Please see Figure 11 which summarizes participants’ roles. The study intends to capture an architectural point of view which is established by having a high concentration of architects. Though, there might be some benefits to having participants from different backgrounds which is discussed in Section 7.

Figure 12 lists the industries participants are coming from. The largest groups are consultancy, insurance, finance, software, automotive, and retail sectors. The fact that there is a high number of consultants enables the opportunity of comparing them with the other architects who have a relatively narrower point of view.

Another important thing to point out is that participants have greatly varying years of experience. Figures 13 and 14 shows their details. Diversified years of experience distribution gives the opportunity to compare perceived success factors depending on the experience.
### Role # of participants

<table>
<thead>
<tr>
<th>Role</th>
<th># of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Architect</td>
<td>12</td>
</tr>
<tr>
<td>Solution Architect</td>
<td>11</td>
</tr>
<tr>
<td>Business Architect</td>
<td>5</td>
</tr>
<tr>
<td>Information Architect</td>
<td>4</td>
</tr>
<tr>
<td>IT Architect</td>
<td>3</td>
</tr>
<tr>
<td>System Architect</td>
<td>2</td>
</tr>
<tr>
<td>Application Architect</td>
<td>2</td>
</tr>
<tr>
<td>Integration Architect</td>
<td>2</td>
</tr>
<tr>
<td>Program Manager</td>
<td>2</td>
</tr>
<tr>
<td>IT Manager</td>
<td>2</td>
</tr>
<tr>
<td>Scrum Master</td>
<td>1</td>
</tr>
<tr>
<td>Business Area Director</td>
<td>1</td>
</tr>
<tr>
<td>Portfolio Manager</td>
<td>1</td>
</tr>
<tr>
<td>Domain Architect</td>
<td>1</td>
</tr>
<tr>
<td>Business Analyst</td>
<td>1</td>
</tr>
<tr>
<td>Managing Architect</td>
<td>1</td>
</tr>
</tbody>
</table>

*Figure 11: The roles of the participants*

### Industry # of participants

<table>
<thead>
<tr>
<th>Industry</th>
<th># of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consultancy</td>
<td>14</td>
</tr>
<tr>
<td>Insurance</td>
<td>5</td>
</tr>
<tr>
<td>Finance</td>
<td>5</td>
</tr>
<tr>
<td>Software</td>
<td>4</td>
</tr>
<tr>
<td>Automotive</td>
<td>4</td>
</tr>
<tr>
<td>Retail</td>
<td>4</td>
</tr>
<tr>
<td>Telecommunication</td>
<td>3</td>
</tr>
<tr>
<td>Healthcare</td>
<td>2</td>
</tr>
<tr>
<td>Government</td>
<td>2</td>
</tr>
<tr>
<td>Public Sector</td>
<td>2</td>
</tr>
<tr>
<td>Fashion</td>
<td>2</td>
</tr>
<tr>
<td>Travel</td>
<td>1</td>
</tr>
<tr>
<td>E-commerce</td>
<td>1</td>
</tr>
<tr>
<td>Home appliances</td>
<td>1</td>
</tr>
</tbody>
</table>

*Figure 12: The industries survey participants work at*
### 6.3 Data Analysis

The data analysis reveals the validity of the EAM success factors, how impactful they are, the differences between consultants and architects, and the differences between varying experience groups. Individual responses to the survey are available in Appendix 2.

The answers to the survey are converted into a number between 0 and 1 as explained in Section 6.1. The next step was to calculate p-values, which indicate the probability of the success factor is valid. Chi-squared test is used during the calculation. p<0.05 is a widely accepted limit for accepting hypotheses and this limit is used for the study. p<0.05 limit indicates that there is less than 5% probability for the null hypothesis to be true, meaning that the tested hypothesis is highly likely to be true. It should be noted that p-values do not provide a logical connection and it simply states the probability. The logical reasons require further investigation.

---

21 Hypothesis testing with p-value: https://onlinecourses.science.psu.edu/statprogram/reviews/statistical-concepts/hypothesis-testing/p-value-approach

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Success Factor</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>High-level management involvement</td>
<td>0.000001</td>
</tr>
<tr>
<td>F2</td>
<td>Mid-level management involvement</td>
<td>0.000009</td>
</tr>
<tr>
<td>F3</td>
<td>Software dependency management maturity</td>
<td>0.022315</td>
</tr>
<tr>
<td>F4</td>
<td>Having a defined architectural maturity (CMMI)</td>
<td>0.040646</td>
</tr>
<tr>
<td>F5</td>
<td>Clearly defining individual responsibilities</td>
<td>0.000000</td>
</tr>
<tr>
<td>F6</td>
<td>Using standardized tools within the company</td>
<td>0.040644</td>
</tr>
<tr>
<td>F7</td>
<td>Having a common architectural repository</td>
<td>0.015982</td>
</tr>
<tr>
<td>F8</td>
<td>Outsourcing level</td>
<td>0.086077</td>
</tr>
<tr>
<td>F9</td>
<td>Business alignment of the technical side</td>
<td>0.000000</td>
</tr>
<tr>
<td>F10</td>
<td>Technical understanding of the business side</td>
<td>0.000047</td>
</tr>
<tr>
<td>F11</td>
<td>Requirements definition and handling its changes</td>
<td>0.000000</td>
</tr>
<tr>
<td>F12</td>
<td>Cross functionality</td>
<td>0.000001</td>
</tr>
</tbody>
</table>

Figure 15: The list of the success factors and corresponding p-values

The p-values are shown in Figure 15. The expected value = 1.00 refers to the case that a given factor is chosen to be “very important” by the participants. F1, F9, F11, F12 have the lowest p-values, therefore they have a higher significance. The expected value = 0.75 refers to the case that a given factor is chosen to be “important” or “very important” by the participants. At this point, most factors obey to p<0.05 which validates them. Though, F8 appears to be not significant.

Figure 16 and 17 ranks the factors according to the average answer given to them. The answers’ averages are taken after being converted into numbers. These figures are consistent with the calculated p-values. Though, the order could have been different due to difference variance between individual answers for factors.

<table>
<thead>
<tr>
<th>Factor</th>
<th>F9</th>
<th>F11</th>
<th>F1</th>
<th>F12</th>
<th>F2</th>
<th>F5</th>
<th>F10</th>
<th>F3</th>
<th>F7</th>
<th>F6</th>
<th>F4</th>
<th>F8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>0.804</td>
<td>0.784</td>
<td>0.740</td>
<td>0.730</td>
<td>0.721</td>
<td>0.676</td>
<td>0.672</td>
<td>0.525</td>
<td>0.510</td>
<td>0.505</td>
<td>0.461</td>
<td>0.407</td>
</tr>
</tbody>
</table>

Figure 16: Perceived success factors
In this step, consultants’ and other architects’ answers are separated from each other and evaluated to see their differences. Figure 19 shows the average of the answers. The comparison between two sides is hard only with averages since the absolute values do not reflect the ranking. Therefore, Figure 19 is scaled to Figure 20 in a way that the factor with the highest value is 1.00 and the factor with the lowest value is 0.00. It is important to note that 0.00 does not mean “not important” and this scaled view does not represent the transformation in Figure 10, Section 6.1. The scaling process is done for comparison purposes according to the formula in Figure 18. The results indicate that consultants value F3, F6, and F7 more than architects. Consultants tend to choose lower scores and they have the same maximum (F9) and minimum (F8) as the architects. The fact that consultants have a lower value on other factors is because of that.
In this step, participants are divided into three groups according to years of experience they have to see their differences. Figure 22 shows the average of the answers. It is scaled according to Figure 18 and its results can be seen in Figure 23 and 24. The differences are significant. Professionals with 0-10-year experience value F1, F2, F5 and F12 much less than more experienced professionals. The other factors do not show a clear pattern.
Years of experience | Data set size | Factors
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0-10 years</td>
<td>19</td>
<td>0.66</td>
<td>0.59</td>
<td>0.50</td>
<td>0.46</td>
<td>0.64</td>
<td>0.51</td>
<td>0.47</td>
<td>0.38</td>
<td>0.86</td>
<td>0.68</td>
</tr>
<tr>
<td>11-25 years</td>
<td>19</td>
<td>0.78</td>
<td>0.80</td>
<td>0.53</td>
<td>0.47</td>
<td>0.71</td>
<td>0.49</td>
<td>0.53</td>
<td>0.47</td>
<td>0.70</td>
<td>0.72</td>
</tr>
<tr>
<td>26+ years</td>
<td>13</td>
<td>0.81</td>
<td>0.79</td>
<td>0.56</td>
<td>0.44</td>
<td>0.67</td>
<td>0.52</td>
<td>0.54</td>
<td>0.35</td>
<td>0.75</td>
<td>0.58</td>
</tr>
</tbody>
</table>

Figure 22: Success factors with respect to years of experience

Figure 23: Scaled success factors with respect to years of experience

Perceived Success Factors Depending on Years of Experience

Figure 24: Scaled success factors with respect to years of experience
6.4 Respondent Comments

Survey participants were allowed to leave comments at the end of the survey. Here are the most frequent comments.

- The answers depend on which the stage the project is in.
- The answers will be different for different types of projects and for different parts of an organization.
- The respondent has seen a number of attempts to create an EA repository which always failed. The repository ended up not being updated and architects turned away the focus to other important things.
- The respondent mentions product-driven development as a success factor.
- The best way to succeed in EA is to understand that specific enterprise. The rest comes later on. In the end, the understanding of the business leads to right decisions. It is never about the technology.
- The survey should have a ‘no answer’ option. This would avoid not intended answers. (The consequences of the survey design will be discussed in Section 7.5.)
7. Discussion

Section 7.1 argues about the interviews and the beginning of the study. Section 7.2 and 7.3 discusses the results of the survey and tries to come up with impact levels of the factors. Section 7.4 goes back to the semi-structured interviews and analyzes the changes companies going through. Section 7.5 elaborates on future transition of the company. Section 7.6 evaluates the validity of the study. Finally, Section 7.7 mentions limitations and future study.

7.1 Determining Success Factors

The architectural success factors are determined with a combination of the literature review, initial evaluation and the semi-structured interviews. As it can be seen in Figure 9, twelve factors have been extracted. In order to ensure that no success factor is missing, these three sources are cross-referenced. Reoccurring statements were added to the list. It is important to highlight details of data sources to ensure their reliability.

The interview participants were working in the same department and the answers were consistent. Therefore, there was no conflict while listing the statements in Figure 8. Having participants working in the same department also provided the study to have a deep involvement. Even though the interview questions were the same, the given answers varied depending on the participant’s position.

Because the initial evaluation was unstructured, it was relatively harder to be objective on the subject. This step allowed gaining directly practical information which is going to be involved in the study. Results of the literature review were rather supportive. Topics mentioned during the initial evaluation was searched through the literature.

7.2 Investigating Success Factors

In order to determine which EAM success factors are more impactful, let’s investigate Figure 15 and 16. Figure 15 indicates two different probabilities, the probability that a given factor is considered to be ‘important’ and ‘very important’. Figure 16 takes a simple average of the given answers. It informs about which factors are considered to be more important, but it does not provide an indication if the answers vary from each other.

The data do not allow a clear separation of the factors. This is in fact expected because both interview and survey participants mentioned how situational success factors can be. Even if a clear ranking was possible, it might not have been accurate. On the other hand, it is possible to have a simplistic approach and to group the factors according to their impact levels.

In Figure 17, the score differences are so low that the ranking could have changed slightly if a couple of architects answered differently. Though, this graph gives an important piece of information. F9 and F11 are clearly the highest scoring factors while F8 is the clear lowest. There is a big score difference between F10 and F3. Coincidentally, the average score is 0.628 and this is the point that separates above and below average by a great margin. F9, F11, F1, F12, F2, F5
and F10 stand above average while F3, F7, F6, F4 and F8 stand below average. This point will be used to divide factors into two impact levels.

The next step is to investigate Figure 15 to see how the two impact levels can be further elaborated. F8’s p-value at expected value = 0.75 makes it the least impactful factor. F1, F9, F11, and F12’s p-values at expected value = 1.00 makes them the most impactful factors. As expected, the separation here complies with the separation in Figure 17. The factors mentioned in this paragraph are above average for the most impactful factors and below average for the least impactful factor.

The logical steps taken here allows creating four impact levels that the factors can be separated into, which answers the second research question. The success factors for EAM are impactful on different levels which are shown in Figure 25. Impact Level 1 consists of the factors most essential for success which are high-level management involvement, business alignment of the technical side, requirements definition and cross functionality. On the other hand, Impact Level 4 is the least essential factor which is outsourcing level. Outsourcing has been rejected as a general success factor in Section 6.3 due to its high p-value, which shows that industry experts do not consider it to be important for the organizational success. Though, it is still considered an architectural success factor for Volvo Cars as it has been mentioned several times during the semi-structured interviews. The purpose of this study is to evaluate factors specifically affecting them.

<table>
<thead>
<tr>
<th>Impact Level</th>
<th>Abbreviation</th>
<th>Success Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F1</td>
<td>High-level management involvement</td>
</tr>
<tr>
<td></td>
<td>F9</td>
<td>Business alignment of the technical side</td>
</tr>
<tr>
<td></td>
<td>F11</td>
<td>Requirements definition and handling its changes</td>
</tr>
<tr>
<td></td>
<td>F12</td>
<td>Cross functionality</td>
</tr>
<tr>
<td>2</td>
<td>F2</td>
<td>Mid-level management involvement</td>
</tr>
<tr>
<td></td>
<td>F5</td>
<td>Clearly defining individual responsibilities</td>
</tr>
<tr>
<td></td>
<td>F10</td>
<td>Technical understanding of the business side</td>
</tr>
<tr>
<td>3</td>
<td>F3</td>
<td>Software dependency management maturity</td>
</tr>
<tr>
<td></td>
<td>F4</td>
<td>Having a defined architectural maturity (CMMI)</td>
</tr>
<tr>
<td></td>
<td>F5</td>
<td>Using standardized tools within the company</td>
</tr>
<tr>
<td></td>
<td>F7</td>
<td>Having a common architectural repository</td>
</tr>
<tr>
<td>4</td>
<td>F8</td>
<td>Outsourcing level</td>
</tr>
</tbody>
</table>

Figure 25: Success factors and their impact levels (Level 1 is the most impactful. The order within the same level does not indicate ranking.)

The study has an unexpected outcome. The success factors are dependent on each other. Supported by the literature review, having standardized tools (F5) have a huge impact on business alignment. Similarly, a common repository (F7), and a good level of dependency management (F3) is often followed by a good requirement management. Defining CMMI (F4) allows a holistic view, therefore, it impacts almost all of the other factors. The common point of
these factors is that they have a much lower score than the others. The respondents had the option to assign the same importance to them, which did not happen.

This indicates that some commonly accepted practices in the literature are not actually valued as much by the architects. This might be because the importance is case-specific as one of the survey respondents suggested. Another possible reason is that a bad previous experience might deter from practice. A respondent pointed out that they were unsuccessful in trying to implement a common repository. This point needs further study to fully understand it.

7.3 Profession and Experience

When consultants’ answers are compared with the others using Figures 19, 20, and 21, there is an interesting pattern. Consultants value dependency management maturity, using standardized tools and having a common architectural repository more than the architects. These are Level 3 factors which are more aligned with the architectural frameworks. All of the Level 1 and 2 factors apart from requirement definition and handling its changes are more related to the way of thinking rather than architectural tools. Consultants focus on architectural frameworks more than the architects.

The survey respondents have different years of experiences. Their data can be utilized to see how architects’ view differ depending on how long they have been working. Figures 22, 23 and 24 reveal that more experienced architects tend to value clearly identifying individual responsibilities, cross functionality, high-level and mid-level management involvement than the less experienced architects.

7.4 Interview-Survey Comparison

The survey reveals that Impact Level 1 and 2 factors are mainly about managerial topics whereas Impact Level 3 factors about architectural tools. In practice, the results are not surprising since the architectural methods are applied in order to support the organization. Software dependency management maturity, standardizing the toolset and having a common repository are steps toward boosting the management involvement, business understanding of the IT and cross functionality. They are not the actual goal, simply the way to go. This makes it plausible that they are not the priority.

On the contrary, most frequently mentioned statements during the interviews at Figure 8 were the lack of common repository, the lack standardization, the lack of business understanding of the architects, and the lack of dependency documentations, which are mostly related with architectural concepts. This shows that architects in Volvo Cars’ value them to generate results in the end.
7.5 The Company’s Future Transition

Volvo Cars have been through two acquisitions for the past 20 years which affected the company significantly. Apart from these two challenges, there is a third challenge coming up with the era of big data. They need to modernize their infrastructure while dealing with legacy applications and reworking the current architecture. Consumer & Enterprise Digital Department has been experiencing two ongoing changes at the moment, one in the short term and one in the long term. The planned changes intend to change the current EA structure.

The short-term challenge is to work on the current information architecture and to create an application that connects different architectural blocks. It aims to overcome the lack of common repository and knowledge transfer tool. Having a common architectural repository in the future would increase architecture maturity. The second step is to gain active involvement and make the repository more sustainable. In addition, documenting dependencies at least among the frequently used applications would be beneficial. A software portfolio without dependencies documented makes it hard to perform rapid changes. Documenting the dependencies would also allow quantitative analysis on the repository [2] [4] [5]. Thus, evaluating architecture agility would be possible, which would allow decreasing dependencies and the software development cost.

Historically, the number of information architects decreased over time and the existing repository stopped getting updated due to managerial reasons at Volvo Cars. In order to reboot it, getting involvement from both high-level management and mid-level management is crucial. The survey revealed that they are both very important, but the priority seems to be at the high-level management according to the industry professionals. The new application plans to create meta models to be used which is an effective approach [24] [25]. It aims to increase overall agility with giving everyone access and promoting them openly to contribute to their individual work. This way, knowledge transfer and standardization over the existing solution will be established. The final goal is to increase the business alignment which is again supported by the literature [3] [27].

The long-term challenge is to transition to the new dSPA platform. This platform is rather driven by the industry needs and business alignment of the technical side is a major success factor for it. The technical side needs to understand the needs and it has to be customer-centric as opposed to the old product-centered approach.
7.6 Validity and Reliability

The study investigated success factors specific to Volvo Cars. On this part, the external validity of the results is limited to the organizations who have a similar architectural maturity and who are in the same industry. On the other hand, the survey has been sent out to the architects in Sweden. The survey has a larger external validity limited to Sweden.

One thing noted by the survey respondents was that the survey did not have ‘no answer’ option. This might have decreased the reliability of the answers and it is better to conduct the survey with an extra option. Though the number of participants was high, and p-values in Figure 15 indicate that the results are still reliable.

7.7 Limitations

Interview participants are assumed to be experts in their fields and the answers are trusted blindly. Also, the interviews had a limited time up to an hour sometimes it needed to be rushed in order to get all the necessary info.

Survey questions had 5-options and not more. In the end, not every factor was clearly separated. A different type of survey can be used in order to make ranking easier. At the same, the 5-option survey was on purpose. All of the participants voluntarily attended the study. The survey is designed in a way that it would be fast to answer.

Hauder’s study [31] indicates a lack of a common definition of how to operationalize EA. This means that the interview and survey participants own definitions might have been different and affected their answers. The study tried to minimize this effect by increasing the number of participants.

7.8 Future Study

During Phase 3, the survey was sent to industry professionals. If the same survey can be sent to both Volvo Cars internally and the industry professionals, this would provide a solid comparison point about how Volvo Cars’ values differ from the others. At the same time, this would provide a more solid result regarding how does Volvo Cars prioritize the success factors.

Since the target group was the architects, the interview and the survey participants were also architects. A supplementary study can be done on the developers and the business side to evaluate how their opinions matter.

Finally, Section 7.1 makes a statement about commonly accepted practices not being accepted by architects due to previous negative experiences. The evidence in this study is not strong enough and an alternative study can be focused on this statement.
8. Conclusion

Architectural issues that are impactful on the success of Volvo Cars have been revealed with semi-structured interviews in the company. These interviews are later on wrapped up with a literature review and twelve EAM success factors were derived. The list of success factors and their sources are available in Figure 9. This figure answers the first research question and lists the main success factors in Volvo Cars’ EA management. After that, the factors are evaluated through a survey answered by professionals all over Sweden.

The result of the survey groups these success factors into four levels depending on how impactful they are, which is the answer of the second research question. The full list can be found in Figure 25. The most significant success factors are the business alignment of the technical side, requirements definition and handling its changes, high-level management involvement, and cross functionality. There are some side results as well. Consultants prioritize architectural frameworks more than architects. Also, more experienced architects tend to value cross functionality, management involvement and identifying individual responsibilities more than the ones with less experience.
References


Mert Canat  

Enterprise Architecture Success Factors: Analysis from Architects’ Perspective


Appendix

Appendix 1: Survey Questions

Enterprise Architecture Success Factors

This is a short survey for a research on Enterprise Architecture at KTH. It is conducted as a part of a master thesis and aimed for anyone working in a field related to enterprise architecture.

It is fully anonymous and you will not be contacted again unless you want to get informed about the results.

*Required

Which industry do you work in? *
For example automotive, telecommunications, consultancy, etc. Mentioning the exact company name is optional.

Your answer

What is your current role? *
For example enterprise architect, solution architect, developer, manager, project leader, product owner, enterprise architect (consultant), solution architect (consultant), etc.

Your answer

What are your previous role(s)?
For example enterprise architect, solution architect, developer, manager, project leader, product owner, enterprise architect (consultant), solution architect (consultant), etc.

Your answer

How many years experience do you have with your current role? *

Your answer
Please select the following factors impact on the overall success of an organization/a project. *

<table>
<thead>
<tr>
<th></th>
<th>Not Important</th>
<th>Slightly Important</th>
<th>Moderately Important</th>
<th>Important</th>
<th>Very Important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outsourcing level</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Using standardized tools within the company</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technical understanding of the business side</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clearly defining individual responsibilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Having a defined architectural maturity (CMMI)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High-level management involvement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross functionality</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software dependency management maturity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements definition and handling its changes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Having a common architectural repository</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business alignment of the technical side</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MId-Lever management involvement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you would like to get informed about the results, please enter your email.

Your answer

Do you have any feedback or comments?

Your answer

Thank you for participating!

For any question or comment, please contact canat@kth.se

The results of the study will be sent out late September.
## Appendix 2: Survey Answers

<table>
<thead>
<tr>
<th>#</th>
<th>Industry</th>
<th>Position</th>
<th>Years</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
<th>F10</th>
<th>F11</th>
<th>F12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Automotive</td>
<td>Portfolio Manager</td>
<td>42</td>
<td>Very</td>
<td>Important</td>
<td>Very</td>
<td>Important</td>
<td>Very</td>
<td>Important</td>
<td>Very</td>
<td>Important</td>
<td>Very</td>
<td>Important</td>
<td>Very</td>
<td>Important</td>
</tr>
<tr>
<td>2</td>
<td>Consultancy</td>
<td>Business Architect</td>
<td>26</td>
<td>Very</td>
<td>Important</td>
<td>Very</td>
<td>Important</td>
<td>Not</td>
<td>Important</td>
<td>Moderately</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
</tr>
<tr>
<td>3</td>
<td>Consultancy</td>
<td>Business Area Director</td>
<td>25</td>
<td>Very</td>
<td>Important</td>
<td>Very</td>
<td>Important</td>
<td>Moderately</td>
<td>Important</td>
<td>Important</td>
<td>Very</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>4</td>
<td>Finance</td>
<td>Business Architect</td>
<td>15</td>
<td>Moderately</td>
<td>Important</td>
<td>Very</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>5</td>
<td>Telecom</td>
<td>Information Architect</td>
<td>5</td>
<td>Moderately</td>
<td>Important</td>
<td>Not</td>
<td>Important</td>
<td>Not</td>
<td>Important</td>
<td>Moderately</td>
<td>Important</td>
<td>Not</td>
<td>Important</td>
<td>Very</td>
<td>Important</td>
</tr>
<tr>
<td>6</td>
<td>Consultancy</td>
<td>Solution Architect</td>
<td>27</td>
<td>Slightly</td>
<td>Important</td>
<td>Moderately</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Not</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>7</td>
<td>Healthcare</td>
<td>Information Architect</td>
<td>7</td>
<td>Very</td>
<td>Important</td>
<td>Moderately</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Very</td>
</tr>
<tr>
<td>8</td>
<td>Consultancy</td>
<td>Enterprise Architect</td>
<td>5</td>
<td>Moderately</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>9</td>
<td>Automotive</td>
<td>Enterprise Architect</td>
<td>16</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Very</td>
<td>Important</td>
</tr>
<tr>
<td>10</td>
<td>Government</td>
<td>Business Architect</td>
<td>28</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Very</td>
<td>Important</td>
</tr>
<tr>
<td>11</td>
<td>Consultancy</td>
<td>Business Architect</td>
<td>18</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
</tr>
<tr>
<td>12</td>
<td>Insurance</td>
<td>IT Architect</td>
<td>10</td>
<td>Moderately</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Important</td>
<td>Not</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
</tr>
<tr>
<td>13</td>
<td>Software</td>
<td>Enterprise Architect</td>
<td>21</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
</tr>
<tr>
<td>14</td>
<td>Retail</td>
<td>Solution Architect</td>
<td>10</td>
<td>Slightly</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
</tr>
<tr>
<td>15</td>
<td>Retail</td>
<td>Enterprise Architect</td>
<td>4</td>
<td>Very</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
</tr>
<tr>
<td>16</td>
<td>Software</td>
<td>Solution Architect</td>
<td>25</td>
<td>Moderately</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
</tr>
<tr>
<td>17</td>
<td>Insurance</td>
<td>Business Architect</td>
<td>18</td>
<td>Very</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
</tr>
<tr>
<td>18</td>
<td>Public Sector</td>
<td>Information Architect</td>
<td>30</td>
<td>Very</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
</tr>
<tr>
<td>19</td>
<td>Consultancy</td>
<td>Solution Architect</td>
<td>30</td>
<td>Very</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>20</td>
<td>Government</td>
<td>Managing Architect</td>
<td>6</td>
<td>Very</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>21</td>
<td>Retail</td>
<td>System Architect</td>
<td>20</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>22</td>
<td>Automotive</td>
<td>IT Manager</td>
<td>37</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>23</td>
<td>Finance</td>
<td>Domain Architect</td>
<td>28</td>
<td>Very</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>24</td>
<td>Insurance</td>
<td>Enterprise Architect</td>
<td>35</td>
<td>Very</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>25</td>
<td>Healthcare</td>
<td>Solution Architect</td>
<td>8</td>
<td>Moderately</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>#</td>
<td>Industry</td>
<td>Position</td>
<td>Years</td>
<td>F1</td>
<td>F2</td>
<td>F3</td>
<td>F4</td>
<td>F5</td>
<td>F6</td>
<td>F7</td>
<td>F8</td>
<td>F9</td>
<td>F10</td>
<td>F11</td>
<td>F12</td>
</tr>
<tr>
<td>----</td>
<td>--------------</td>
<td>----------------</td>
<td>-------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>26</td>
<td>Travel</td>
<td>Application Architect</td>
<td>1</td>
<td>Slightly Important</td>
<td>Moderately Important</td>
<td>Moderately Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Very Important</td>
<td>Very Important</td>
<td>Very Important</td>
</tr>
<tr>
<td>27</td>
<td>Public Sector</td>
<td>Architect</td>
<td>21</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>28</td>
<td>Software</td>
<td>Solution Architect</td>
<td>20</td>
<td>Very Important</td>
<td>Important</td>
<td>Not Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Not Important</td>
</tr>
<tr>
<td>29</td>
<td>Telecomm.</td>
<td>Enterprise Architect</td>
<td>7</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>30</td>
<td>Consultancy</td>
<td>System Architect</td>
<td>8</td>
<td>Slightly Important</td>
<td>Moderately Important</td>
<td>Important</td>
<td>Very Important</td>
<td>Important</td>
<td>Important</td>
<td>Very Important</td>
<td>Important</td>
<td>Important</td>
<td>Very Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>31</td>
<td>Consultancy</td>
<td>Senior Master</td>
<td>10</td>
<td>Important</td>
<td>Not Important</td>
<td>Moderately Important</td>
<td>Not Important</td>
<td>Not Important</td>
<td>Not Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Not Important</td>
</tr>
<tr>
<td>32</td>
<td>Finance</td>
<td>Enterprise Architect</td>
<td>27</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>33</td>
<td>Retail</td>
<td>Enterprise Architect</td>
<td>35</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>34</td>
<td>Consultancy</td>
<td>Enterprise Architect</td>
<td>21</td>
<td>Very Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Not Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>35</td>
<td>Finance</td>
<td>Solution Architect</td>
<td>20</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>36</td>
<td>E-commerce</td>
<td>Solution Architect</td>
<td>2</td>
<td>Slightly Important</td>
<td>Moderately Important</td>
<td>Important</td>
<td>Very Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>37</td>
<td>Fashion</td>
<td>Enterprise Architect</td>
<td>5</td>
<td>Very Important</td>
<td>Important</td>
<td>Not Important</td>
<td>Moderately Important</td>
<td>Important</td>
<td>Important</td>
<td>Not Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>38</td>
<td>Consultancy</td>
<td>Integration Architect</td>
<td>27</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>39</td>
<td>Consultancy</td>
<td>Integration Architect</td>
<td>20</td>
<td>Moderately Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>40</td>
<td>Appliance</td>
<td>Application Architect</td>
<td>8</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>41</td>
<td>Insurance</td>
<td>Enterprise Architect</td>
<td>28</td>
<td>Very Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>42</td>
<td>Consultancy</td>
<td>Information Architect</td>
<td>18</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>43</td>
<td>Consultancy</td>
<td>Solution Architect</td>
<td>20</td>
<td>Very Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>44</td>
<td>Automotive</td>
<td>Program manager</td>
<td>20</td>
<td>Very Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>45</td>
<td>Telecomm.</td>
<td>IT Architect</td>
<td>10</td>
<td>Moderately Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Very Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>46</td>
<td>Consultancy</td>
<td>Solution Architect</td>
<td>7</td>
<td>Very Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Not Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>47</td>
<td>Finance</td>
<td>Solution Architect</td>
<td>18</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>48</td>
<td>Fashion</td>
<td>Program manager</td>
<td>5</td>
<td>Very Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>49</td>
<td>Insurance</td>
<td>Business Analyst</td>
<td>9</td>
<td>Very Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>50</td>
<td>Software</td>
<td>Enterprise Architect</td>
<td>12</td>
<td>Very Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td>51</td>
<td>Software</td>
<td>IT manager</td>
<td>21</td>
<td>Very Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Slightly Important</td>
<td>Important</td>
<td>Important</td>
<td>Important</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1</td>
<td>High-level management involvement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2</td>
<td>Mid-level management involvement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F3</td>
<td>Software dependency management maturity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F4</td>
<td>Having a defined architectural maturity (CMMI)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F5</td>
<td>Clearly defining individual responsibilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F6</td>
<td>Using standardized tools within the company</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F7</td>
<td>Having a common architectural repository</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F8</td>
<td>Outsourcing level</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F9</td>
<td>Business alignment of the technical side</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F10</td>
<td>Technical understanding of the business side</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F11</td>
<td>Requirements definition and handling its changes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F12</td>
<td>Cross functionality</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>