

Improving Enterprise Architecture at Agile Working Financial Organizations

Document: Master Thesis

Version: 1.2 (Final)

Date: 27 February 2017

Author: Vincent Boekholtz

Master of Business Informatics

Institute of Information and Computing
Sciences, Utrecht University

Supervisors Utrecht University

Dr. S.J. Overbeek

Dr. S. España

Supervisor Rabobank WRR

Ronald Zwart



Utrecht University



Rabobank

Abstract

Organizations have to adapt to a rapid pace of technological innovation that enables new technical solutions and business opportunities. In these organizations EA should be an enabler of such adaptations by being more efficient and effective, so that in combination with working in an agile way (IT development) these organizations can evolve fast and in a structured way, to accommodate changes needed. However, EA is not always perceived as an organization wide effort, it sometimes is still seen as purely an IT effort, not always suited to work together with agile, as EA has a long-term focus, while Agile has a short-term focus.

This research aims to improve the EA discipline in organizations, so that it can become an enabler for structured organizational change, based on the vision and the mission of the business. The research showed through a Systematic Literature Review and four Case Studies that EA can be such an enabler, if it is organized and implemented the right way into the organization. It is independent of the IT development method, it doesn't even matter it is agile or plan-driven: EA can always be organized efficiently and effectively.

Eight Essential Elements (EEs) [33] were found that help make the EA discipline more effective and efficient in the organization. Three of the four case studies used most of those EEs and reported to have a streamlined EA discipline in their organization that initiated IT projects based on architecture, while one of the case studies did not address all eight EEs and reported a (very) ineffective EA discipline in their organization, which was delaying their IT development and caused their architectural landscape to be incoherent and scattered.

Keywords: Enterprise Architecture, Plan-Driven, Agile, IT-Development, Financial Domain.

Acknowledgement

This thesis is the culmination of my Master of Business Informatics at Utrecht University. For over a year this has been a big part of my life, not always as clear and present as it should have been, but still in my mind.

There are a lot of people I should thank for this opportunity, the state the thesis is in right now and for the fact that it is finished.

First of all I would like to thank Sietse Overbeek, my first supervisor from Utrecht University, without your feedback, support, the pressure of deadlines you created, but most of all your patience, I would not have made it this far. Furthermore, I am Rabobank WRR and in particular Ronald Zwart really grateful for giving me a place to conduct my research at, the guidance, insights help whenever needed. Also the colleagues of APS, the IT architects in particular were always very interested and helpful regarding the research. Furthermore, I owe Slinger Janssen a thanks a well, for his interest in my progress and friendly advices. My second supervisor Sergio España should be thanked as well, of course, for providing me with insights during the preliminary phases.

Rabobank IT Netherlands (Rob Douwes, Edwin van Gorp and Mirjam Verlinden-van den Berg), SNS Bank (John Hendriks) and ABN Amro (Jeroen Jedema) should be acknowledged and thanked for giving me some of their time, insights and information in order to make this research more valid and certainly interesting. Furthermore, without you I would not have reached the insights and conclusions I have now.

Last, but certainly not least, I want to thank my family and friends for all their help and support during this research. A special thanks for Johan, who provided me with a place where I could work after I was done at Rabobank WRR.

Table of Contents

ABSTRACT	I
ACKNOWLEDGEMENT	III
TABLE OF CONTENTS	V
LIST OF FIGURES	VII
LIST OF TABLES	VII
1 INTRODUCTION	1
1.1 BACKGROUND	1
1.2 PROBLEM STATEMENT	2
1.3 RESEARCH QUESTIONS	4
1.4 SCOPE	4
1.5 RELEVANCE	4
1.6 DEFINITIONS	5
1.7 MAIN DELIVERABLES	5
2 RESEARCH APPROACH	7
2.1 SYSTEMATIC LITERATURE REVIEW	8
2.2 CASE STUDY	13
3 THEORETICAL BACKGROUND	17
3.1 ENTERPRISE ARCHITECTURE	17
3.2 AGILE SOFTWARE DEVELOPMENT	19
3.3 ENTERPRISE ARCHITECTURE AND AGILE SOFTWARE DEVELOPMENT	20
4 CASE STUDY	25
4.1 CASE 1: RABOBANK WHOLESALE, RURAL & RETAIL	26
4.2 CASE 2: RABOBANK IT NETHERLANDS	44
4.3 CASE 3: ABN AMRO BANK	49
4.4 CASE 4: SNS BANK	52
5 ANALYSIS	55
5.1 IMPLEMENTATION OF ENTERPRISE ARCHITECTURE AND IT DEVELOPMENT	55
5.2 RELATIONSHIP BETWEEN ENTERPRISE ARCHITECTURE AND IT DEVELOPMENT	58
5.3 CASE STUDY AND SYSTEMATIC LITERATURE REVIEW	59
6 LIMITATIONS, CONCLUSIONS AND FUTURE WORK	63
6.1 LIMITATIONS	63
6.2 CONCLUSIONS	64
6.3 FUTURE WORK	66
7 REFERENCES	69
APPENDICES	73

List of Figures

Figure 1 Research Approach Overview PDD	7
Figure 2 Systematic literature review approach as described by Okoli & Schabram [41]	9
Figure 3 PDD of Systematic Literature Review	11
Figure 4 Case study approach as described by Yin [64]	13
Figure 5 Organogram Rabobank	27

List of Tables

Table 1 Key Points of Hensema [24] mapped to EEs of Lumor [33]	60
Table 2 EA of Case Study companies mapped to Essential Elements of Lumor [33]	61
Table 3 Tags Used for Interview Results Summary	77
Table 4 Interview Results Summary Example	79

1 Introduction

1.1 Background

This background describes how technological progress (increase of computing power, storage and bandwidth) enables innovative technologies (e.g. blockchain and payment solutions) to disrupt the financial domain. Established financial organizations respond to these disruptions through changed or new services and products, supported by IT, developed in an agile way. However, enterprise architecture and agile software development are in conflict with each other: traditional (blueprint) enterprise architecture is focusing on long term stability of the enterprise architecture, while agile software development focuses on fast software development and releases in order to respond to (technological) market innovations.

Because of the pace of change regarding essential technical features, which includes: 1) the power of computer chips doubling roughly every 18 months [36] 2) computer storage doubling every 12 months [62] and 3) effective bandwidth doubling every 21 months [40], there seems to be a shift from 'what is possible' to 'what can we do' with technology. Even taking into account reports of some of these developments slowing down, the increased computational power, storage and bandwidth have increased so much that the possibilities of technological innovations seem endless [6]. This leads to disruptive (technological) innovations such as the blockchain: a massive ledger of transactions made that is verified and shared by a network of nodes), which is an interesting technology for financial institutions [28].

It is assumed that these kinds of innovations may impact any sector and one of the sectors impacted by innovations is the financial sector. Organizations in the financial sector often rely heavily on IT, where the established organizations usually have legacy systems in place for their essential records and processes, which are not easily replaced or modified [9,60].

At the same time, these innovations enable new business models, with new start-ups, FinTech (Financial Technology, referring to organizations that offer financial services which are enabled through technology [3]) and cross sector moving companies, all trying to capture more and more of the banking value-chain. These kinds of companies moving into the financial sector are causing disruptions by excelling in one or more of five major areas: 1) finance and investment, 2) operations and risk management, 3) payments and infrastructure, 4) data security, and 5) customer interface [3]. Big financial organizations historically focus on all of the areas, resulting in a situation where the established organizations have to rethink their business models and roles [38].

Organizations can choose between continuity or change regarding progress in their markets [58]. Change comprises for example development, or acquisition of companies who already harnessed and offer these kinds of disruptive innovations [10]. To achieve this, (some) organizations (for example Rabobank WRR, an international focused Dutch financial institution and focus of the case study [64]) aim to change their way of working in order to be very adaptive and agile; they describe this as an "agile way of working".

Organizations in the financial domain rely heavily on IT components and solutions in order to execute their organizational processes, which will have to be acquired or developed and integrated in the organization [9,60]. This is done through IT projects, but the way these projects are carried out has changed a lot over the last two decades. While quantitative studies regarding adoption of agile software development methods are scarce, it is suspected that traditional IT project methodologies (e.g. the waterfall method) have been cast aside by many organizations and that they have adopted agile software development methods instead [63]. A 2008 survey in Norway reports an adoption rate of 58% (considering Rational Unified Process or RUP as an agile method) regarding agile methods in the Norwegian IT Industry [8]. Agile software development is a response to long, extensive projects, to enable a shorter time-to-market for (new) software products [50]. This shorter time-to-market is achieved through some of the characteristics of agile software processes described in [35]: modularity on development process level, iterative with short

cycles (ranging from one to six weeks) enabling fast verifications and corrections and incremental process approach that allows functioning application building in small steps.

As the field of IT (projects) matured in the past, organizations realized that a process should be in place that would help them with the strategic aim of their IT [31], to control and minimize risks, costs and redundancy of the enterprise IT. Lapalme et al. also state that this process has been implemented in many organizations through Enterprise Architecture (EA) [31]. Several EA frameworks have been developed (e.g. the Zachman Framework [66]; TOGAF [43]), focusing on extensive models (e.g. data model, process model, application model), depicting the (intended) current state of the EA.

Enterprise Architects are expected to have a desired future (to-be) state in mind and should steer the organization to this state, for example by reviewing, declining (giving recommendations) and approving project plans before the project starts. To reach the desired future state of the EA, policies, principles and guidelines are developed, which new IT projects have to adhere to. Every time the EA is changed (e.g. through an IT project, or through a new strategic vision), the models should be updated, making it a reactive, continuous process.

Developing software in an agile way and having an EA process in place might cause friction between these two fields within organizations. Rabobank WRR is an example of such an organization which experiences friction between traditional EA (i.e. blueprint EA as proposed by [66] and TOGAF [43]) and agile software development methods, because EA within Rabobank WRR has a long-term focus, whereas the agile software development teams focus on short-term solutions and developments. In recent literature it is argued that the field of EA needs to continue to mature and reinvent itself in order to be able to assist organizations to address the challenges of the future [31].

This background described how organizations are experiencing friction between blueprint EA and the agile way of developing software. Agile software development is needed however to be able to respond to an ever faster changing market. These market changes manifest themselves through disruptive innovations (e.g. blockchain), enabled through a long period of exponential increase of technological features such as computing power, storage and bandwidth.

1.2 Problem Statement

The situation described in the background can be mapped on the Five Forces Model developed by Porter [44]. The (disruptive) innovations enable new entrants (both new companies as well as existing organizations moving cross-sector like Apple with Apple Pay and Google with its Google Wallet), substitute products or services (such as crowdfunding and peer-to-peer lending) as well as increased rivalry among existing competitors (they innovate their products and services as well to remain competitive). This poses a problem for established organizations, because if the market changes this fast, there is a choice for companies to make. Higgins gives several options (invest, harvest, defend, divest, select and niche) as a strategy for a product life cycle [25]. This is applicable in this case too and complements the Porter Five Forces Model in giving insight in the organizations position and possible strategy. Organizations can either react inadequately or adequately, which is explained as follows:

1. The organization does not adapt (e.g. it does not have the capability to do so) to these changes and innovations; in which case competitors or newcomers might fill in the gap.
2. The organization chooses to adapt, but does so too slowly. The results may be the same as under 1) and catching up with the early adopters might be tough.
3. The organization chooses to adapt to these changes and innovations, as fast as possible and possibly not following all policies, principles and processes, in which case the EA models might become unsynchronized with reality [51]. This would make it difficult to steer the organization towards the desired future state of the EA.
4. The organization chooses to react to changes in an adequate way. If the first option is not the one the organization desires, the organization will aim to respond to the changes.

While option one is not an inadequate reaction in its core (according to Higgins you can also choose to defend or divest in a product [25]), this still poses a risk for a company as competitors and/or newcomers might fill in the gap. Option 2 and 3 are a much more adequate way of responding, while option 4 is clearly the right way to respond. Options 2, 3 and 4 can be mapped to the Higgins strategies invest, harvest, select and niche. The strategy to follow is highly dependent on a specific organization: if the market attractiveness is high and the competitive position of the company is also high, an invest strategy may be appropriate [25]. If the market attractiveness is high, but the competitive position of the company is weak on the other hand, a niche strategy may be appropriate [25].

An agile way of working to respond to these changes and innovations is one way how organizations choose to adapt [37]. Agile originated in the field of software development however, introduced by the agile manifesto [19]. Organizations now attempt to extend agile to the rest of the organization [57] and thus expand it further than its original scope of software development.

The IT side of an organization is supportive to the business side of the same organization, according to Conway's law [12]. Therefore, if the organization chooses to adapt to (disruptive) innovations in the market, so does the IT side of the organization. Some of the aspects of agile software development described in the agile manifesto are: "working software over comprehensive documentation", "responding to change over following a plan" and "individuals and interactions over processes and tools" [19]. This results in short iterations, self-managing teams and refactoring. This way of agile working conflicts with traditional (blueprint) EA, as control and responsibility of documentation and adhering to the EA future state lies for a large part with the teams, as they are self-managing. However, documentation for example has been suspected to be one of the first things neglected when teams are running out of time [56]. As described before, Rabobank WRR is an example of an organization that experiences friction between its EA discipline and agile software development. It is assumed that the reason for this friction is the long-term focus of enterprise architects (to control and steer the organization EA towards its desired future state) and the short-term focus of the self-supporting teams who wish to develop products and software in an agile way:

- Agile software development revolves mainly around the short-term: working software through in every short sprint, responding to change instead of following a plan, working software is more important than comprehensive documentation. [26]
- EA revolves around the long-term: stable IT systems, a desired future state and following guidelines and principles as well as delivering artifacts by the software development teams [45].

The above friction presumably results in a conflict between agile software development on the one side and traditional EA on the other, which this research aims to resolve.

Summarizing, the aim of this study is to help organizations cope with the problem of adapting to a rapid pace of technological innovation that enables new technical solutions and business opportunities. In these organizations EA should be an enabler of such adaptations by being more efficient and fast, so that by working in an agile way these organizations will remain competitive in their market. The research will provide organizations with a set of guidelines regarding EA, with which they can review their own way of conducting EA and formulate improvements to their EA. By using this set of guidelines, it is assumed that EA will no longer be an impediment to an agile way of working, while the benefit that enterprise architects provide by keeping an eye on the big picture and desired future state of the EA will not be lost.

1.3 Research Questions

To achieve the aim stated in the problem statement the following main research question (MRQ) was formulated:

“How can the enterprise architecture discipline (including its governance, models and artifacts) be (re)designed, with the aim to improve the responsiveness of IT in agile working organizations operating in the financial domain?”

In this case, (disruptive) innovations are innovations that require a change in the EA when responded to (e.g. technical innovations such as the blockchain, or market innovations such as crowdfunding and peer-to-peer banking).

In order to be able to answer the main research question, several sub research questions (SRQ) were formulated:

SRQ1: How is the enterprise architecture discipline implemented in agile working organizations?

The first sub research question addresses the way EA is implemented in agile working organizations. In order to be able to identify where the friction occurs, it is important to know how EA is implemented: what is the EA process, what are the deliverables and how is it governed?

SRQ2: How is IT responding to changes in agile working organizations?

This second sub research question will explore what agile entails: how are organizations working in an agile way?

SRQ3: What are the differences between how IT aims to respond to (disruptive) innovations and how the enterprise architecture discipline is implemented?

This will shed light on areas of friction or areas that work well together. This combines SRQ1 and SRQ2 and identifies mismatches between (traditional) EA and an agile way of working.

1.4 Scope

The scope of this research is narrowed down to large established organizations, which have an established EA discipline and an (ambition to develop an) agile way of working, within the financial domain. The financial domain is heavily relying on IT where in such established organizations legacy systems and applications are still up and running and new innovations are implemented as well, operating next to these systems. It is assumed that FinTech companies do not have these old legacy systems and applications and are therefore usually better able to adapt to technical innovations. Large established organizations are affected by a phenomenon called ‘handicap of a head start’ [52], which does not affect the (startup) FinTech companies. Successful recommendations made for this domain may be potentially generalizable to other domains as well.

1.5 Relevance

1.5.1 Societal Relevance

The societal relevance of this research is mainly the fact that it aims to bring scientifically grounded ideas to the field of Enterprise Architecture and Agile IT Development. Up to now most of the work is done by practitioners, mostly from the agile field. Also providing guidelines into the EA discipline in order to improve IT responsiveness could help organizations structure their EA in order to get the most out of their IT development.

1.5.2 Scientific Relevance

The scientific relevance lies in the fact that little scientific research has been done on the intersection of EA and agile IT development. This research will perform case studies in organizations in order to compare them to scientific literature. The literature found so far is pure

scientific, with little real world validation. Conducting case studies at organization and validating the scientific research done so far would improve the field of Agile EA research. The Systematic Literature Review will chart the research done in the field of Agile EA research so far and provide an overview of the work done. This could be used as a baseline report on the status of scientific research done regarding agile EA so far.

1.6 Definitions

Enterprise Architecture

A coherent set of descriptions, covering a regulations-oriented, design-oriented and patterns-oriented perspective on an enterprise, which provides indicators and controls that enable the informed governance of the enterprise's evolution and success. [42]

1.7 Main Deliverables

Systematic Literature Review

A systematic literature review [41] will be performed on the combination of Enterprise Architecture and Agile Software Development. Since previous research on the combination of these is scarce [7,22], this will be a scientific addition to the body of knowledge and help other researchers identify gaps in the body of knowledge. It will be presented as part of the thesis.

Case Study

A Case Study based on Yin [64] will be performed, mainly within Rabobank WRR, but validated outside of Rabobank WRR. This Case Study will be well documented, to ensure it enriches the body of knowledge towards Enterprise Architecture and Agile Software Development. It will be presented as part of the thesis.

Guidelines to make Enterprise Architecture more agile

Based on the Systematic Literature Study and the Case Study, a general set of guidelines will be composed to make Enterprise Architecture more agile and presented at the end of the thesis.

2 Research Approach

The research method will consist of two methods, a systematic literature study as proposed by Okoli & Schabram [41] and a case study as described by Yin [64]. First, a systematic literature review will be conducted, followed by a case study, where the systematic literature review will be seen as a source of evidence [64]. The entire research is depicted in a process-deliverable diagram (PDD) as proposed by [59]. A PDD depicts activities to be performed on the left side and deliverables (connected by dotted lines) of those activities on the right side of the diagram. All deliverables depicted in the diagram are assumed to be generally available for all activities, hence there are no lines depicting usage of deliverables by activities. The PDD can be found in Figure 1 Research Approach Overview PDD.

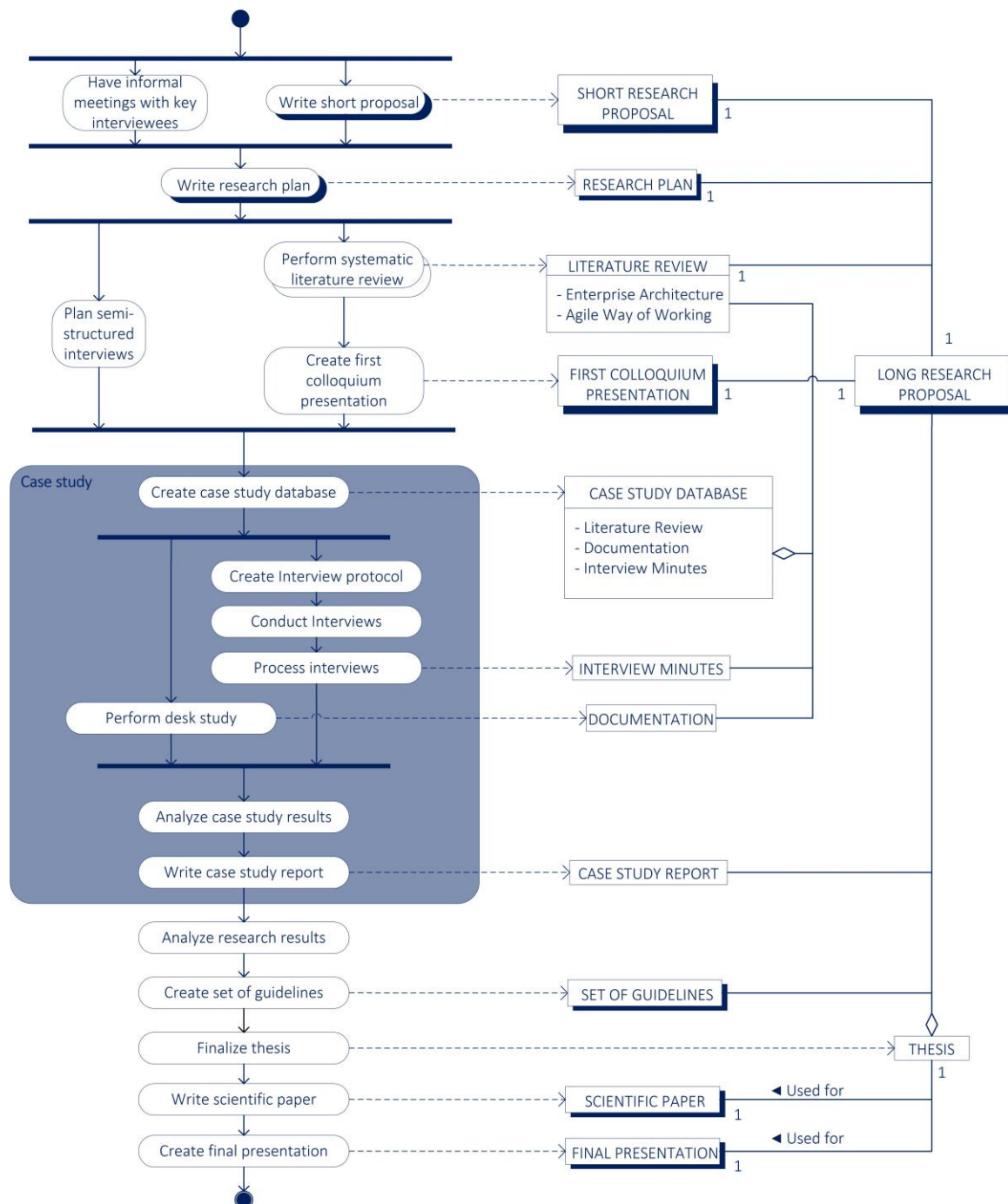


Figure 1 Research Approach Overview PDD

The choice for a case study method instead of a design science research method has to be explained, because when the research questions in chapter 1.3 are interpreted, the expectation might arise that a new method or method chunks might be developed. However, this is not the case, as the goal of this research is to develop guidelines with which organizations improve their EA and not to propose a new method. Therefore the decision was made to perform a case study, as the purpose of a case study may be to “understand the complex relationship between factors as they operate within a particular social setting” [13]. Understanding these complex relationships between factors will provide the possibility to propose improvements to the EA of organizations (if their situation is similar to that of the case study unit of analysis).

2.1 Systematic Literature Review

The steps of the systematic literature review (SLR) as proposed by Okoli & Schabram [41] are shown in Figure 2 Systematic literature review approach as described by Okoli & Schabram [41]. Because this is a method developed to write a standalone SLR [41], the steps are elaborate and extensive. Okoli & Schabram argue that for a Master’s thesis the student is usually held to lesser high standards compared to a standalone SLR or a doctoral dissertation, because there is a more limited timeframe to complete the study [41]. For a Master’s thesis it is usually a background or theoretical foundation and the data collection and analysis are part of the primary study [41]. Therefore, the step “writing the review” as proposed by Okoli & Schabram [41] will not be carried out as though the SLR is a standalone research, but as a part of a Master’s thesis. This results in a time-manageable SLR (no rigorous recording and discussing of all literature needs to happen), which fits in a Master’s thesis timeframe, while the SLR will still be thorough (it can be assumed that most, if not all relevant literature will be found).

It has to be noted that while little scientific research has been devoted to EA in combination with agile software development, it has been getting more attention from practitioners and thought leaders [7,22]. Reports, blogs and publications of these practitioners and thought leaders are generally published through blogs, whitepapers and other sources. Because these kinds of publications are not peer reviewed and generally are based on gut feelings and experiences with little scientific references, these publications will not be discussed in the SLR

2.1.1 Planning

Purpose of the literature review

There are several purposes for this literature review. First, it is important to analyze the progress of a stream of research already done in a field [41]. Such an overview will expose gaps (if there are any) in a body of knowledge. Furthermore, this literature review will provide an introductory section for the thesis [41], on which the thesis will be built and enables the research to be placed in a context of related work exposed by the literature review. While a less thorough literature review might have been sufficient if the purpose was limited to the first three arguments, this literature review aims to be a source of evidence [64] for the case study in the subsequent phase of this research. Because it is aimed to have this literature review as a source of evidence, it is important to have a rigorous method for the literature review in order to establish and maintain a chain of evidence [64].

The goals of the literature review are therefore fourfold:

1. Identify gaps in the existing body of knowledge
2. Provide an introductory section for the thesis
3. Provide a context in which the research can be placed
4. Provide a source of evidence for the case study

Protocol and training

The protocol describes the locations to be searched, the search criteria to be used, inclusion and exclusion criteria, how the data will be extracted and aggregated (synthesized) and how it will be

reported [41]. This protocol will improve the rigor of the literature review, reproducibility of the literature review and traceability of the chain of evidence to be used in the case study. The protocol of the SLR is described in the remainder of this paragraph.

A training of the reviewers is not applicable. Training in this context means informing the researchers on the literature review protocol to make sure every researcher adheres to it. As there is only one researcher carrying out the literature review, this is not applicable.

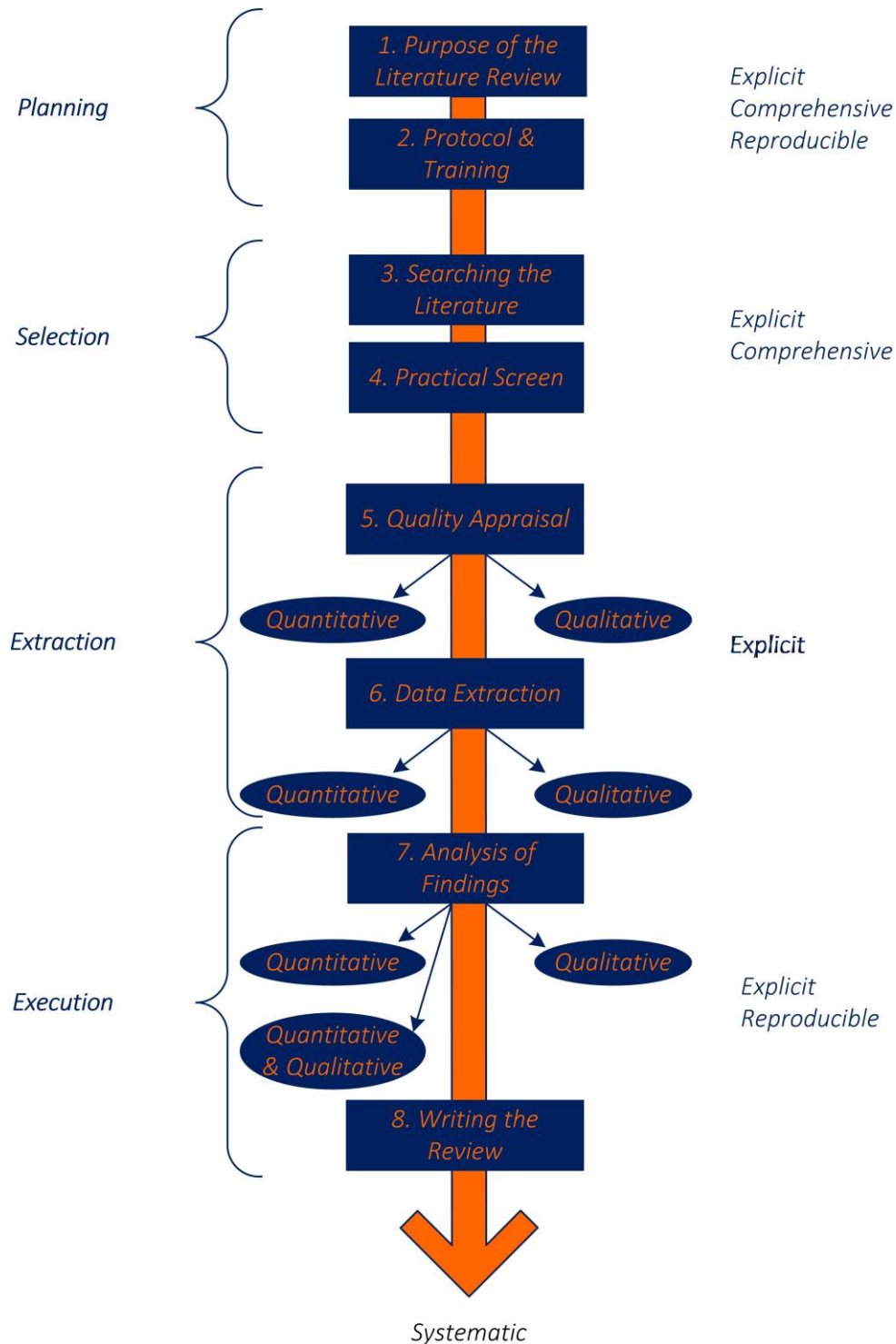


Figure 2 Systematic literature review approach as described by Okoli & Schabram [41]

2.1.2 Selection

Searching for the literature

Searching for literature will be done in five primary search engines for information and computing sciences. The results from these primary search engines will be selected based on their titles: if the title is clearly not applicable, the result will be excluded. The list of results will be complemented with results from two secondary search engines, of which the title will determine whether or not the result will be included. This last step is carried out to see if there are important results that were not found using one of the primary search engines. An overview of the SLR process for this research can be found in Figure 3 PDD of Systematic Literature Review which depicts the search process and how the following steps contribute to the literature review.

The primary search engines selected for this research are selected:

1. ACM Digital Library
2. IEEE Computer Society Digital Library
3. Science direct
4. Web of Science
5. SpringerLink

The secondary search engines are:

1. Google Scholar
2. JSTOR

A predetermined set of keywords will be used in each of the search engines (both the primary and secondary ones), in order to ensure a comprehensive results list, and make the search repeatable. There are two main sets of keywords to be distinguished in order to be able to answer the research questions: a set revolving around EA and a set revolving around agile.

For the EA set, a broad set of keywords will be used, so that every aspect of EA will be included. While the set of keywords exceeds the field of EA, they may yield interesting results nonetheless as they overlap the field of EA:

- Enterprise Architecture
- Enterprise Modeling
- Enterprise Transformation

The second set of keywords regards the agile aspect of the research questions. This set of keywords will encompass a broad range of keywords because the research questions revolve around agile in the broadest sense: not only agile software development, but also an agile way of working within organizations. An agile way of working means that organizations aim to extend the agile principles to the rest of the organization. This results in the following set of keywords:

- Agile software development
- Agile SDM
- Agile way of working

The next step is a backward search and assesses the titles of the references used in the studies listed in the results list, which are used to further complete the results list. For the added studies to the results list, this step will be carried out once more.

The backward search will be followed by a forward search: if the search engine supports a "cited by" option, the titles of articles that cite the study, will be assessed and, if applicable, be added to the results list. This step will be carried out until the "cited by" count is zero. A backward search as described in the previous step will be carried out on the newly added results once. If this yields

new results, a forward search on the new results will be carried out on these new results to identify new studies, which will be iterated until no new results are found.

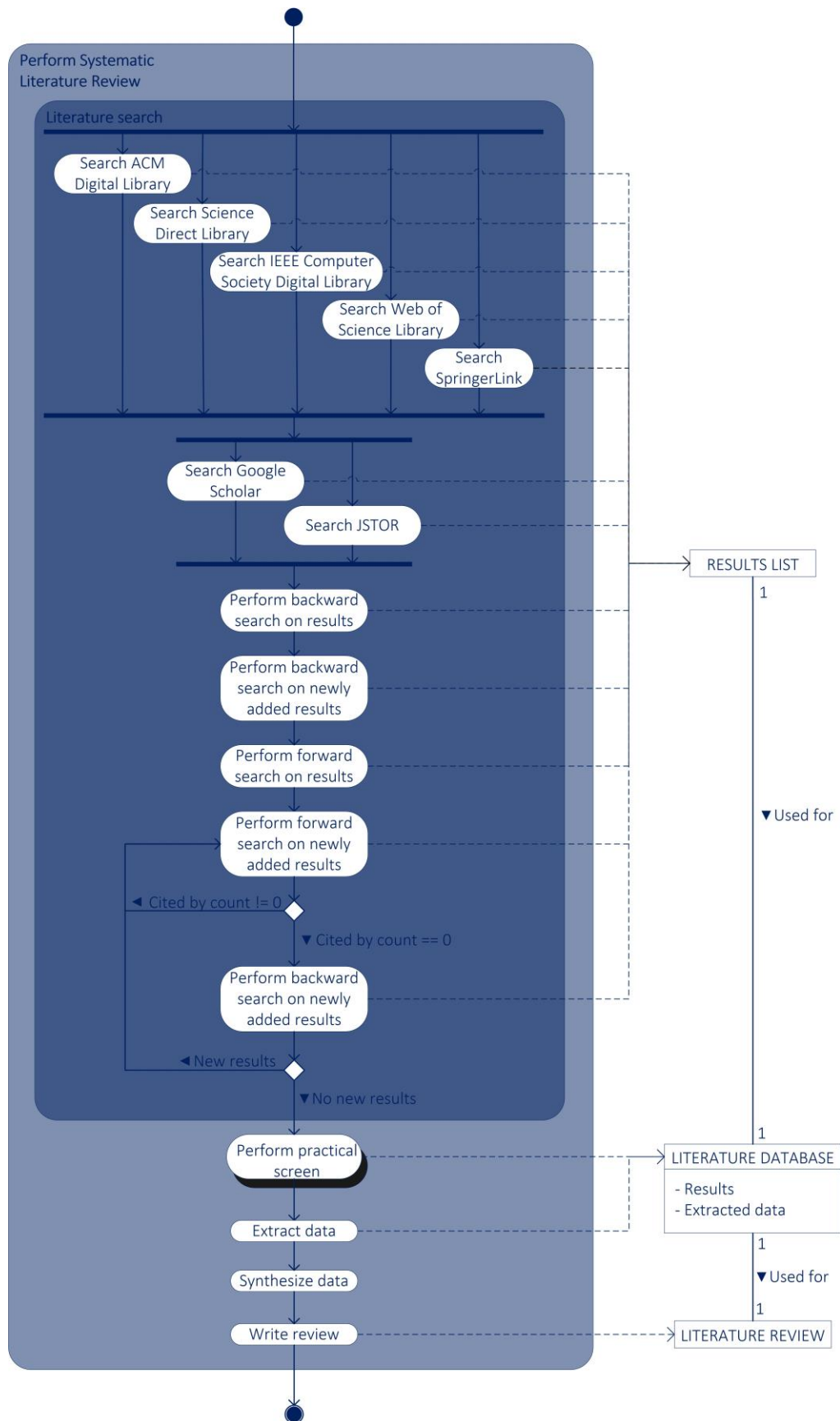


Figure 3 PDD of Systematic Literature Review

Practical screen

The practical screen will determine which studies should even be considered for the review: the studies content has to be applicable to the research and further explicitly defined criteria restrict the total number of articles considered [41]. The first step in the practical screen is therefore selection on inclusion criteria:

- A date range of studies published up to and including 2015. No start date will be used to screen for inclusion. The end date of 2015 is used in order to make this literature research repeatable.
- Only journal & conference papers will be used in order to ensure high quality results.
- The publication language has to be English as this is a language understood by most scholars.
- The availability of the study has to be free of charge or accessible via the Utrecht University proxy, because there is no funding for this research.

To determine if the study is applicable to this research, the abstract will be reviewed. If the abstract proves that the study is applicable, it will be included in the results. If the applicability of the study is doubtful, the result will be flagged for further examination (i.e. the background and conclusion will be assessed on applicability).

2.1.3 Extraction

Quality appraisal

The quality appraisal is also a screen for exclusion: studies are examined more closely to assess their quality and insufficient quality studies are excluded for the review synthesis, not all literature reviews will eliminate studies based on their quality though [41]. This literature review will not screen studies on their quality, if the studies are included in the results list following the practical screen, any useful information will be extracted from the studies in the following step.

Data extraction

After the list of results has been narrowed down to applicable studies, the data will be extracted from them. Each article will be read and the useful raw material for the synthesis stage will be extracted and documented [41]. Determining whether or not data is useful will be assessed based on the research question while reading the article.

2.1.4 Execution

Analysis of data

The data and facts extracted from the studies will be combined and organized into categories so that a coherent overview of information will emerge. This will enable discussion, aggregation and comparison of the information. The information can then be used to compose a structured and coherent literature review, but also serve as a source of evidence during the case study.

Writing the review

This final step comprises the documentation of the review. The documentation consists of the process of the literature review, documented in sufficient detail so that that entire procedure is reproducible [41]. Also an overview of the results is part of the review. This overview will satisfy the goals for the literature review, as described earlier:

1. Identify gaps in the existing body of knowledge
2. Provide an introductory section for the thesis
3. Provide a context in which the research can be placed
4. Provide a source of evidence for the case study

Because the literature review in itself is not to be published, the literature review will be limited to a discussion of the studies, not of the process of the literature review, as that is discussed in this chapter.

2.2 Case Study

To answer the research questions in chapter 1.3, a case study as proposed by Yin [64], consisting of six phases will be carried out. The phases of the case study are depicted in Figure 4 Case study approach as described by Yin [64]Figure 4. Each of the phases will be elaborated in the following subsections.

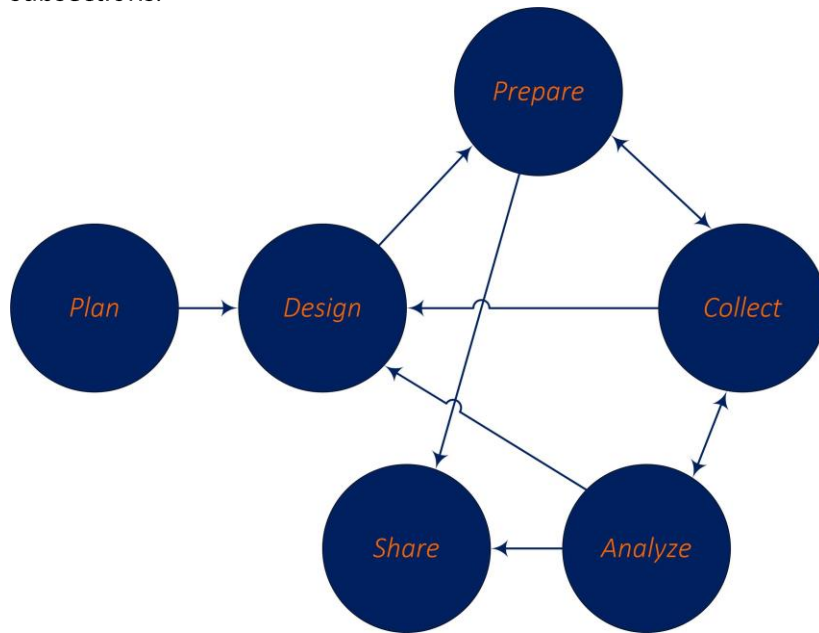


Figure 4 Case study approach as described by Yin [64]

2.2.1 Plan

A case study is the preferred method in several situations: if 1) “how” or “why” questions are posed, 2) the investigator has little control over events and/or 3) the focus is on a contemporary phenomenon within a real-life context [64]. This research fits these conditions because:

1. The aim of this research is to answer a “how” question (see Section 1.3).
2. The investigator has little control over events as he is not an employee of any financial organization, but a Master student performing an internship at a financial organization. The events will not change during this research because of the investigator because first the organization will be observed, before recommendations regarding improvements will be done.
3. It concerns a contemporary phenomenon within a real-life context as EA is performed in the organization as a part of their business.

2.2.2 Design

Study questions

The questions used in this study have already been defined (see Section 1.3).

Propositions

There is a friction between traditional EA and agile ways of working (see Section 1.2). This is presumed to be the result of the long-term focus of traditional EA and the short-term focus of agile ways of working. The SLR will attempt to verify this presumption. The purpose of this case study is

therefore to develop a theory by researching both sides and determine what the main causes for the friction are.

Unit of analysis

A case study can comprise a single case or multiple cases [64]. Yin [64] states that a single case study can be performed if the case study is a representative or typical case. Yin [64] furthermore argues that often multiple case studies offer more compelling evidence, but that they also “require extensive resources and time beyond the means of a single student”. Because of the time and resource constraints regarding this research, a hybrid form will be executed: one extensive case study will be performed, in which an embedded design (multiple units of analysis within each case study) is used [64]. To be able to provide more compelling evidence, three smaller case studies will be performed, which will focus on the same subjects as the extensive case study, but not as elaborate.

Rabobank Wholesale Rural and Retail (Rabobank WRR) has been selected as the case unit of analysis for the extensive case study. Three other smaller case studies will be performed in order to verify that Rabobank WRR is not the only organization struggling with these problems (the friction between EA and an agile way of working) and questions (how to relieve this tension). These semi-structured interviews will use an interview protocol similar to the interviews conducted within Rabobank WRR. These semi-structured interviews [32] will be conducted at organizations in the financial domain: Rabobank IT Netherlands, SNS Bank and ABN Amro.

The units of analysis within Rabobank WRR are the different domains (business units). These domains are structured in the same way, but employees from these separate domains (architects and developers) mention different levels of friction between EA and their agile way of working. The embedded design is chosen in order to be able to distinguish reasons for differences in friction between traditional EA and agile ways of working. As there are different levels of friction experienced in each of the domains (by architects and developers) between EA and their agile way of working, researching each business unit individually should yield differences between the business units, which could be the reasons for less (or more) friction between EA and their agile way of working.

A detailed description of Rabobank WRR and each domain can be found in Chapter 4.

Data to be collected

There will be two sources of evidence [64] in the case study: semi-structured interviews and a desk research.

Semi-structured interviews [32] will be performed to gain insight in how units of analysis are actually working and provide the rationale behind it. There is a risk for biased and/or distorted information. Biased information may be provided to make a point and/or negate or exaggerate importance of what the interviewee is talking about. Distorted information may be provided when there is a difference in the actual way of working and the intended (official) way of working. This may be done by the interviewee to protect him/herself and/or the team, or because they perceive this to be the correct answer.

Desk research will in turn provide information regarding the intended or official way of working and can be used to corroborate and understand the information gathered from the interviews. Usually there is no rationale provided in documentation, but it ensures an unbiased source of information.

Data interpretation

The data collected from the sources of evidence will be documented per unit of analysis, in a structured unified way in order to be able to analyze the results. This will be done through pattern matching and explanation building [64].

Pattern matching will be applied between the units of analysis: if two units of analysis report less friction because of a certain reason and a third one reports increased friction because it is handling it different, that may be an important aspect to take into account to reduce friction between EA

and the agile way of working. Also the results from the SLR may expose differences in patterns, which may be interesting to reveal.

Explanation building is the second technique used to interpret the data. An explanation for friction between EA and the agile way of working will be the basis on which the tool for improvement of EA will be developed. This too, will also use the SLR to help explain the observations made in the case study.

2.2.3 Prepare

Part of the preparation is the development of a case study protocol [64]. For this research, the protocol will encompass a description of how access will be gained to key interviewees and documentation as well as how the human subjects of the case study will be protected.

Gaining access to key interviewees will be done by introducing the researcher in an early stage of the research and by giving the interviewee information regarding the research. This will be done through informal meetings of roughly 30 minutes, even before the SLR is performed. This will create trust, an increased sense of involvement and reduce the introductory time during the semi-structured interviews where as much data needs to be gathered as possible. During the SLR, follow-up interviews will be planned with the same interviewees and new interviewees recommended by them. By planning the interviews early on, the risk of the interviewee not being present or not having time will be minimized.

Protection of the human subjects will happen, if they so desire it, through anonymization of the data gathered. Even though interviewees will be asked if there are any objections to the interviews being recorded, the eventual summaries and reports would not contain any name of interviewees. In that case, general descriptions of the functions and/or roles of interviewees will be used to put the data into context. This is not desirable, however as it would impact the traceability of information used as evidence in the case study description in a negative way.

Another aspect of the preparation phase in the case study is a pilot case study. While this could in theory be performed on a unit of analysis, due to time constraints and the availability of interviewees, a pilot case study will not be performed. When scheduling the interviews, it will be attempted to have a diverse type of interviewees early on, to test the interview protocol and make changes based on these, if needed.

2.2.4 Collect

Sources of evidence

Multiple sources of evidence will be used to counter problems regarding construct validity [64]. The sources of evidence will be documentation and interviews. For the three smaller case studies, only the interviews will be used as evidence, only if interviewees have documentation to share, this will be incorporated as well.

Documentation will be gathered through desk research and will consist of examining documents from Rabobank WRR, which will be collected and researched. These documents will be requested in the semi-structured interviews and collected via the internal document systems of Rabobank WRR. This documentation will be used to corroborate information gathered from the interviews [64] and later on in the analysis to find mismatches between reality (how is Rabobank WRR actually working) and theory (how work is intended to be done).

Interviews will take mainly place as semi-structured interviews [32] and will be performed across various disciplines within Rabobank WRR, in order to ensure a balanced view. An interview protocol (Appendix A Interview Protocol) for the semi-structured interviews will be developed to be able to explore matters as they come up in the interviews, while still asking all the questions needed for the case study. If agreed upon by the interviewee, these interviews will be recorded, so that it can be summarized later on. These summaries will then be presented to the interviewee so he/she can provide final feedback, elaborate further on, or clarify data in the summary in order to prevent misunderstandings, which might occur during the interview or summarizing.

Case study database

A case study database increases the reliability of the case study because other researchers can then use the same evidence (if they have access to it) to derive the conclusions drawn [64]. Therefore all collected data will be stored in a case study database. All relevant documents, interview recordings and interview summaries will be stored here.

Chain of evidence

The chain of evidence is important in a case study because it increases the reliability of information collected and used [64]. Therefore, the evidence collected in this phase will be traceable to one or more of the research questions of this research (1.3 Research Questions).

2.2.5 Analyze

The data from the three sources of evidence (systematic literature review, semi-structured interviews, and desk research) will then be triangulated in order to find mismatches. These potential mismatches may provide areas where improvements are possible. In order to ensure a high quality analysis, the four principles proposed by Yin [64] will be taken into account:

1. Use all the evidence
2. Address all major rival explanations (if any)
3. Address the most significant aspect of the case study
4. Use own prior expert knowledge in case study

2.2.6 Share

Sharing the results will be done through this document, in which the process of the case study is described, the analysis is documented and the results and conclusions are stated. Also a scientific paper might be created in which the case study would be presented.

3 Theoretical Background

The theoretical background is comprised of three parts:

Enterprise Architecture – an overview of relevant literature regarding EA in an unstructured ad hoc way. No specific search method was used to accumulate the literature used and is intended to provide insight into what EA is considered to entail within this research. This will be discussed in two parts: the history of EA and the concept of EA.

Agile software development – similar to the literature on EA, this section gives an overview of relevant literature regarding agile software development, in an unstructured ad hoc way. No specific search method was used to accumulate the literature used and is intended to provide insight into what agile software development is considered to entail within this research.

Enterprise Architecture and agile software development – the literature found during the systematic literature review (SLR) will be discussed here. The process leading to the literature is described under 2.1. The literature found has been categorized in two parts, as the found literature discusses agile organizations (how to adapt to changes as an organization) and agile software development.

3.1 Enterprise Architecture

A vision paper written by Proper and Lankhorst [45] describes the origin of the EA field well. This part will use that vision paper extensively to describe the history of this field. Following the history of EA, the concept of EA will be explained, according to work done by Op 't Land et al. [42].

3.1.1 History of Enterprise Architecture

The origin of EA can be traced back to the mid 80's of the last century. In 1986 the PRISM project was conducted [45], and in Europe the ARIS framework was presented [45]. The Zachman framework [66] is often referred to as the founding of the field of EA, even though PRISM and ARIS predate it and that these three frameworks focus on information systems architecture [45]. In the 1990's authors realized that the focus should be on a broader enterprise context rather than information systems. Spewak [55] was one of the first authors to use the term enterprise architecture [45]. The EA field started to reach the conclusion that the business architecture should not be taken as a given anymore and that both should be created together, so that they could influence each other [45]. The strategic alignment model of Henderson and Venkatraman [23] played an important role in recognizing the importance of aligning these two. The IT architecture was mainly aligned to the business architecture however, resulting in a 'Business-to-IT-stack' [45]. Then, in the late 2000's, authors [18,20,61] argued that just aligning IT to the business was a weakness of EA approaches [45]. According to [61] EA should not just be concerned with Business-IT alignment, but rather with the alignment of all relevant aspects of an enterprise and introduces the term enterprise coherence instead of alignment. Just after this shift towards enterprise coherence, the field of EA tried to mitigate another risk: from blueprint thinking and "Big-Design-Up-Front" (where the entire organization should be defined and modelled), a shift towards 'just enough architecture' was made. Terms like "Fit-for-Purpose", "Just Enough, Just in Time" started to surface in the early 2010's, which should start to bring the field of EA and the agile community more together [45].

Finally, [45] describe three very recent trends, happening simultaneously. First, EA approaches are moving away from constructing towards a constraining perspective. "A shift from considering an architecture as being primarily concerned with constructing the (high level) design of an enterprise in terms of building blocks to being concerned with constraining the space of allowable/desirable constructions." [45] Second, a shift from building to integrating, where organizations move from designing and building their own information systems and applications from the ground up towards integrating out of the box solutions and thus limiting the design freedom of the architects [45]. The third trend is organizations moving away from one shot to iterative approaches. This is fueled by the adoption of agile software development methods and results in a friction between architects (who are perceived to have to think everything out up front) and software developers

(who state that stakeholders don't know everything beforehand and the project is bound to change while underway, rendering any design up front useless). Both are misguided regarding the role of architecture however, as EA helps position new developments in the existing context [45].

3.1.2 The Concept of Enterprise Architecture

The definition of EA is adopted from [42] who define EA as:

"A coherent set of descriptions, covering a regulations-oriented, design-oriented and patterns-oriented perspective on an enterprise, which provides indicators and controls that enable the informed governance of the enterprise's evolution and success."

This definition is based on other definitions, as described by amongst others IEEE, TOGAF, Netherlands Architecture Forum, ArchiMate Foundation, CapGemini and Gartner Group. While there is variety in these definitions indicating EA's infancy, they also have things in common, forming the base of the above definition. All these definitions revolve around structure, relationships and governing principles that provide guidance and support for directions and decisions [42].

Op 't Land et al. [42] incorporate these aspects into the definition through the notions of three perspectives:

1. *"A regulation-oriented perspective - which manifests itself as a prescriptive notion governing the design of an enterprise. When taking this perspective, one will focus on principles, leading to rules, guidelines, and standards, focusing the enterprise's design freedom in the direction of its success."* [42]
2. *"A design-oriented perspective - which emphasizes the comprehensive and cohesive specification of an enterprise in all its facets, as a high level design. This perspective focuses on essential design decisions, as well as its core structures. When taking this perspective, one typically produces models that describe the design of actual systemic artefacts and their interrelations."* [42]
3. *"A patterns-oriented perspective - which focuses on the use of design patterns. This perspective forms a bridge between the regulative and the design perspectives. To meet the regulations set out in the regulative perspective, during design activities, suitable patterns can be applied."* [42]

In order to help organizations in successfully executing their strategy, EA is an active planning and steering instrument, revolving around stakeholders and their concerns and four main components [42]:

Stakeholders and their concerns

"A stakeholder is an individual, team, or organization (or classes thereof) with interest in, or concerns relative to, a system [11] (such as an enterprise). Concerns are those interests, which pertain to the system's development, its operation or any other aspect that is critical or otherwise important to one or more stakeholders." [42]

Stakeholders make decisions regarding the direction of an enterprise. They do this on different levels within the organization: CxO's, managers, project leaders. These decisions need to be made with an understanding of the impact they will have. This is done with the help of EA and communicated at different levels across the organization [42].

Principles

Many different definitions of principles exist, but there are three general perspectives on principles.

1. Principles as inherent laws: "properties of (classes of) a system that can be observed and validated" [42].
2. Principles as imposed laws: "properties of (classes of) a system that can be validated", which typically address the concerns of stakeholders [42].

3. Guidelines: “properties of (classes of) a system that are specific enough to provide guidance to operational behavior that make it fit within the borders set out by imposed laws” [42].

Models

According to [42] “a multitude of graphical and nongraphic models are needed” in EA. These models usually span over multiple dimensions of focus, goals and purpose. Models referring to one specific version/alternative of an enterprise need to be coherent with models above that dimension.

Views

Views are a representation of the models, for the different stakeholders to act upon their concerns toward a system. This means that the views are derived from the models, to give an abstract view on reality without changing the model of the domain itself. This means that every model is a view, but not every view is a model as a view may have a narrower scope than a model, or span across various models [42].

Frameworks

Frameworks provide structure for architects to choose views. They “provide an ontology which uses different abstraction levels to map all kinds of information needed”. They position architecture results and enable a diverse communication. Best practices and tools may be included in frameworks as well. [42]

This section started by describing the history of EA. Since the mid 1980’s, EA has come a long way, but is still maturing. Moving away from the initial blueprint thinking, EA is becoming more iterative, focusing on integration rather than building and having a constraining instead of a constructing perspective. An explanation of the concept EA followed with a definition of the concept and a discussion of the most important components (stakeholders and their concerns, principles, models, views and frameworks).

3.2 Agile Software Development

It is widely acknowledged that the introduction of the extreme programming method (XP) [4,5] has been the starting point of agile software development methods. Reactions from practitioners regarding agile methods were positive from the very start, even though little was known about actual benefits [2]. Because agile software development was embraced so enthusiastically, agile practitioners and consultants came together to form the “Agile Manifesto” [2]. This agile manifesto has four main values which depict what agile is all about [19]:

1. “Individuals and interactions over processes and tools”
2. “Working software over comprehensive documentation”
3. “Customer collaboration over contract negotiation”
4. “Responding to change over following a plan”

Furthermore, the Agile Manifesto states “while there is value in the items on the right, we value the items on the left more.” [19]. Besides this Agile Manifesto, twelve principles are listed to make the Agile Manifesto a bit more explicit [19]:

1. “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”
2. “Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.”
3. “Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.”
4. “Business people and developers must work together daily throughout the project.”

5. "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."
6. "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."
7. "Working software is the primary measure of progress."
8. "Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely."
9. "Continuous attention to technical excellence and good design enhances agility."
10. "Simplicity--the art of maximizing the amount of work not done--is essential."
11. "The best architectures, requirements, and designs emerge from self-organizing teams."
12. "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly."

Looking at these values and principles it can be noted how they differ from the older plan-driven methods. These plan-driven methods assume that requirements for information systems can be explored and locked down before an actual project, but in reality this is often not the case. For this reason the call for more lightweight and agile methods arose [14,34,39], which eventually resulted in the Agile Manifesto.

After the introduction of the Agile Manifesto, organizations were keen on adopting agile software development methods, as in 2005 research showed that "14% of companies are using agile methods, and that 49% of the companies that are aware of agile methods are interested in adopting them" [15]. To show how fast agile methods were adopted: a 2006 survey conducted in Norway reports an adoption rate of 29% of pure agile methods and if Rational Unified Process is to be considered agile as well, this increases to 58% [8].

This section also started with the origin of the field and how it originated. It then went on to explain the four values and twelve principles of agile, characterizing the agile software development ideology. The main difference between plan-driven and agile software development is the realization that stakeholders cannot know what requirements should be developed beforehand, which is what the values and principles aim to solve.

3.3 Enterprise Architecture and Agile Software Development

While Enterprise Architecture and agile software development have been around for several decades and have been widely adopted [45], little research has been done regarding a combination of these fields [7,22]. Questions regarding such a combination might be along the lines such as "What is the impact of EA on agile software development?" (or vice versa), "Can (elements from) agile methods improve EA?". Also, literature on EA and in particular EA methods propagates to enable agility for the organization [27,53]. The literature found in the systematic literature review revolves around these two themes as well:

1. EA and agile organizations (note: this refers to agility of an organization, not agile software development).
2. Map/use agile principles and/or values to (improve) EA and possibly aim to introduce a new EA framework.

3.3.1 Enterprise Architecture and Agile Organizations

Greefhorst discusses the dependencies between EA and Service Oriented Architecture (SOA) [21]. He states that "business agility mostly depends on the flexibility of applications" and that therefore "an important part of an EA should revolve around applications". Since SOA aims to decouple applications by making a (reusable) service interface and assembling applications is faster than constructing them from scratch, which makes the application landscape more flexible and thus the business more agile. By combining EA and SOA the inflexibility of legacy applications should be

solved and reduce the time to market [21]. SOA to create enterprise flexibility has also been researched by Erol, Sauser & Boardman [17]. They state "SOA is a widely accepted software architecture approach, which promises the design and implementation of flexible systems and facilitates the change of business processes quickly". They aim to take SOA to an enterprise level instead of just a software architecture approach. According to Erol, Sauser & Boardman, SOA "simplifies the development of enterprise applications as modular, reusable business services that are easily integrated, changed and maintained (just like Greefhorst did [21]), which leads to an increased organizational agility. The two enablers of enterprise flexibility are: 1) the capability to connect people, processes and information and 2) the alignment of information technology and business goals [17].

Yu, Den & Sasmal aim to make organizations more adaptive, by involving Business Intelligence (BI) [65]. BI enables organizations to gain insights quickly and recognize needs for change, as well as the ability to take actions in response. To change, the ability to adapt is needed in an organization, however "EA has currently limited ability to model and reason about the adaptiveness that is available or desirable in various parts of an organization". Various types of change to which an organization needs to respond are for example a competitive threat or technological innovations (both external), or a strategic initiative (internal). EA, should have an overview of the entire organization and how IT systems may be adapted to contribute to the business-level adaptiveness, however: "current EA framework have no specifically addressed enterprise adaptiveness" [65]. Morabito, Sack, Stohr and Bhate discuss the flexibility of organizations [37]. They view the entire organization as a set of systems, which can be categorized as thin and thick systems. Thin systems are characterized by explicit, visible, well-defined attributes such as routine process workflows. Thick systems are more tacit: organizational culture and learning. While tacit systems generally responds more slowly than explicit systems, Morabito et al. argue that they are the most flexible system, if they are constructed to embrace change, learning and have respect for people [37].

A proposal for two different speeds of Enterprise Architecture Management (EAM) was put forward by Abraham, Aier and Winter [1]. They recognize two types of Enterprise Transformation (ET): planned, proactive transformation and emergent, reactive transformation. Because organizations are faced with both types of change projects, "EAM needs to support both strategic, proactive change while also being able to provide swift assistance when enterprises are forced to react to external pressures". This leads to their statement that EAM must be able to executed in two "speeds": 1) "as a dynamic capability, providing a complex set of artifacts to support a planned, time-consuming reconfiguration process", and 2) "as an improvisational capability, supporting fast, spontaneous reconfigurations with a simple set of artifacts". In order to enable an organization to achieve this, "governance and feedback mechanisms are critical" [1].

3.3.2 Enterprise Architecture and Agile Software Development

One of the first scholars to use the term Agile Enterprise Architecture (AEA) was Edwards in 2006 [16]. Considering the agile manifesto was written in 2001, this may be considered quite early, as the adoption of agile software developments started to really gain momentum around 2008 [16]. As the field of research regarding EA and agile software development was new, Edwards [16] did not use many scientific literature references. While this is understandable, he could have used references regarding EA or agile software development separately, but also these are scarce. Edwards does list a number of problems AEA solves, such as "an adaptive versus a predictive approach", "traceability between EA and software development", "separation from data and function" and "maturity of the EA practice". Edwards then explains how each of the identified problems and how AEA would solve them. Interesting is to see how Edwards describes the four agile principles and how AEA would incorporate those principles [16]. While Edwards explores and gives pointers for an AEA framework, Shirazi, Rouhani and Shirazi introduce a framework for AEA in 2009 [54]. They too use few scientific references however and the proposed AEA framework remains vague. They do mention an agile principle from the agile manifesto, but they do not explain how the values are incorporated in the framework. Furthermore, no validation is done on the framework [54]. Diving further into agile methods and EA, Buckl et al. [7] go on to map agile terminology and activities to the field of EA. EA challenges are stated first: "alignment with

stakeholders' interests", "early and periodical delivery of concrete EA products", "commitment and involvement of all parties" and "continuous adaptation to a volatile environment". Later on Buckl et al. interpret and translate scrum terminology to EA management. In their work, Buckl et al. also acknowledge that little scientific research has been done regarding Agile Software Development and Enterprise Architecture and state that more research is needed [7].

Lankhorst and Proper [29] discuss how (Enterprise) Architecture can help organizations to "design agile systems" as well as "control agility and keep a balance between stability and change". They also elaborate how architecture processes fit within an agile context. They state that architecture can help organizations be more agile in three ways: "it gives designers and developers insight into a system and its environment they need for making changes, "it provides a way of designing organization-level agility" and "it helps in focusing design effort on those points of variability or uncertainty that are important from a business perspective" [29]. Two capabilities are distinguished that may be architected for different kinds of agility: the execution system ("things needed for business as usual", focusing on system agility) and the innovation system ("the capability to innovate or change", focusing on process agility). Lankhorst and Proper furthermore stress the need for "separation of concerns": changes remain as local as possible, "low coupling and high cohesion": eases analysis and understanding of the system and thus increasing agility and "encapsulation" so that an implemented system can be changed without affecting the environment. They argue that use of standards is not limiting agility, but increases the rapid development of new solutions from existing building blocks [29]. Furthermore, they stress the need for decoupling and reuse of systems or system elements. Lengthwise standardization (standardizing steps taken in a process) increases agility as you can potentially reuse them in other processes, but that crosswise standardization (the same implementation of a process step across different process stream) may hurt agility because you can't tailor or change process steps for different processes, as the step would become too complicated to incorporate every aspect of both processes [29]. Decisions regarding architecture should be "just in time, just enough" and a big design up front should be avoided. Taking decisions too early is dangerous as the decision made may be the wrong one, resulting in rework and refactoring. An analysis phase before starting the actual project can help address risks and define the essence of the architecture. Furthermore, by providing boundaries of the design space, architecture can make sure that technical debt gets out of hand [29]. Finally, Proper and Lankhorst argue to use the process of enterprise architecting proposed by Op't Land et al. [42], which consists of 1) create 2) apply 3) maintain, but to run them continuously and in parallel with development [29]. These may be executed and manifest themselves in the form of project start architectures (PSA), project level architectures and so on and will be sources of information regarding (enterprise) architecture for projects and the organization [29].

Lankhorst and Van Gils [30] later propose an approach to a rapid realization of business value, based on [29]. While they focus on agile organizations mainly, AEA is discussed as well. In this model Lankhorst and Van Gils define five process agility levels, from initial, managed, defined, quantitatively managed to optimizing. Also five system agility levels are defined: silos, standardized technology, optimized core, business modularity and dynamic venturing. Organizations can measure their agility based on this two-part capability model [30]. After defining product and service aspects for an organization, four levels of abstraction are considered: 1) requirements level 2) design level 3) implementation level and 4) infrastructure level. It is recommended to address individual product and service aspects first, to maintain a grip on the complexity [30]. Focusing on individual systems alone is not enough however, because this may result in agile silos. So, an architecture approach is needed to ensure coherence. This should not be practiced in a "waterfall" or top-down style when being applied in an agile environment: decisions should be made as locally as possible, just-in-time and in a lightweight manner [30]. Architecture should be practiced in such a way that it helps run the organization, but also supports and enables change. Furthermore, it should support the operations side and the management of the organization, by using (enterprise) architecture as a feedback loop, enabling the organization to continuously improve [30].

According to Hauder et al. literature documenting the agile nature of EA management is scarce, but they also state that "From an empirical standpoint, we witness that EA management endeavors of our industry partners increasingly apply these agile practices" [22]. Several examples are given of research where AEA is introduced, but most of this research is not scientifically sound, according to [22]. It is argued that the cited literature has little to no validation, quantification and empirical relevance. It is also mainly rooted in practical work experience [22]. Based on previous literature on applying agile software development principles to enterprise architecture, Hauder et al. created a survey to gain empirical data on which principles are actually applied in organizations. Their survey (n=105) shows that "while most organizations perform retrospectives within their EA management team, only few value time over quality. Most EA management initiatives apply an iterative (~79%) and incremental (~87%) approach. About 93% of the organizations apply EA management in a self-organized manner". Some agile principles are not applied by a majority of organizations, mainly related to the quality and completeness of the developed EA products: "Around 38% of the organizations are struggling with outdated EA results", ~65% of the organizations struggle with reluctant information providers [22]. The reason for these issues is "the long range vision of EA management, while agile practices focus on incorporating findings from ongoing projects immediately in the process" [22].

An effort to apply agile in EA is made by Hensema [24]. A comprehensive literature review is done, followed by a survey. Three requirements for an AEA approach are proposed: 1) "it should be a process such that new requirements can be incorporated during development", 2) "a method should reflect on the work that has been done" and 3) "a system that enforces that only artifacts are produced that are essential" [24]. Furthermore four key points are discussed:

- "EA artifacts are not what delivers value. The process of creating the artifact is what delivers value to stakeholders. EA artifacts are a tool for documenting the process." [24].
- "Communication of artifacts is not solved by a development approach. It is a skill of the architect." [24].
- "Incremental and iterative development are possible but require rough borders: high-level principles and boundaries." [24].
- "Create a definition of done for EA artifacts." [24].

Finally, Lumor proposed an Agile Enterprise Architecture Management Method (Agile EAMM) [33]. This Agile EAMM is based on eight "essential elements" (EEs), which are derived from "factors that influence the agility of the enterprise, namely; agility drivers, agility providers and agility capabilities" [33]. The eight EEs are:

1. "An Agile EAM Method should provide a mechanism to develop and continuously refine EA vision, goals, purpose, principles, and scope to accommodate changes in enterprise competitive bases, and turbulence in the external and internal environment." [33]
2. "An Agile EAM Method should maintain appropriate balance between long-term and short-term orientation and satisfy EA stakeholders through early, continuous and incremental delivery of effective and efficient EA products towards achieving a (shifting) long term goal/strategies." [33]
3. "An Agile EAM Method should provide an appropriate decision making and governance mechanisms (e.g. collaborative decision making and consensus building) that result in participation and commitment of all stakeholders, and decisions that benefits the enterprise as a whole rather than individual domains." [33]
4. "An Agile EAM Method should promote the development of skilled teams and adaption of methods and tools, and provide mechanisms for defining EA requirements and criteria for project goal fulfillment." [33]
5. "An Agile EAM Method should support appropriate representation of architectural artifacts and communicate architectural documentations across the enterprise in a common language." [33]

6. "An Agile EAM Method should provide mechanism to develop and maintain an integrated architecture repository for knowledge management and enterprise awareness initiatives." [33]
7. "An Agile EAM Method should act as a dynamic capability (e.g. as sensing, learning, integration, and coordination capabilities), and support enhancement of agility capabilities and development of agility providers of the enterprise." [33]
8. "An Agile EAM Method should promote the anchoring/integration of its mechanisms and tools into the culture of the organization." [33]

This section starts with discussing how EA can make organizations as a whole more agile, through SOA, Business Intelligence and by implementing EA in such a way that it allows for two different speeds in order to allow an organization to transform reactively and proactively. After discussing organizational agility, the focus shifted towards agile software development. Shortly after the introduction of the Agile Manifesto, scholars started arguing for a form of Agile Enterprise Architecture. Early frameworks and EA methods remain vague or try to map agile aspects one on one to EA aspects, which don't always work or make sense. The overall consensus is that EA should become more agile however, enabling the organization to be more flexible and that the EA discipline should become more lightweight and pragmatic. Later research proposes three concrete requirements for an Agile Enterprise Architecture, supported by four points on how to realize this. The section concludes with eight essential elements regarding Agile Enterprise Architecture.

4 Case Study

The information gathered in the Case Study (interview data and internal documentation if available) will be the basis with which the sub research questions will be answered. Sub research questions one and two (SRQ1 and SRQ2) will be answered based on direct information from the case study results. Sub question three (SRQ3) will be answered based on direct information regarding the combination of EA and (agile) IT development gathered from the case study, in combination with an analysis of the information gathered for SRQ1 and SRQ2.

According to the research approach described in Section 2.2, the case study was conducted mainly within Rabobank WRR as the main case of the case study. At Rabobank WRR 13 interviews were conducted and various internal documents were acquired from the intranet and shared by interviewees. With the aim to be able to generalize findings from the main case study, three more case studies were conducted at other financial organizations (banks): three interviews were conducted at Rabobank IT Netherlands; one interview was conducted at ABN Amro Bank; and one interview was conducted at SNS Bank.

As Rabobank WRR is the main case, informal and preliminary meetings were held right after the research started in order to get acquainted with the organization. Because there was no time for a pilot case study, information needed for the semi-structured interview [32] protocol was gathered from these meetings. The interview protocol would revolve around the sub SRQs to ensure that the information needed to come to a conclusion would be gathered. This meant that the interview protocol would revolve around the subjects "Enterprise Architecture", "(agile) IT development" and "Friction between Enterprise Architecture and (agile) IT development". These subjects are still very broad, however, so more depth was needed.

During those preliminary informal meetings it became clear that Rabobank WRR distinguished three components of its EA: 1) Governance; 2) Process; and 3) Deliverables. (R. Zwart, Progress Review, December 12, 2015; J. Bekkers, FINAL Report IT Architecture 2012, internal document, July 30, 2012). These components were added to the interview protocol under the subject "Enterprise Architecture". Regarding (agile) IT development questions regarding the "Process" and "Deliverables" were devised. The third subject regarding Friction between EA and (agile) IT development asked the interviewee about "EA as an impediment or enabler for (agile) IT development". A fourth subject was added by request of Rabobank WRR and revolved around a more general "Agile way of working", so not specifically targeted at IT development. This concept was too vague for the interviewees, however and yielded very few results. The protocol, with the subjects and their breakdowns can be found in Appendix A Interview Protocol.

Interviews were recorded and later summarized using an Excel Worksheet so tags (see Appendix B Tags used for Interview Results Summary) could be added to information gathered through the interviews. A part of this excel sheet can be found in Appendix C Interview Results Summary Example to give an example of statements gathered and tags used.

This chapter will provide case descriptions for all four cases. They will similarly be structured, according to the interview protocol. They all have a short introduction, followed by a description of the organizational structure, focused on the placements architects in the organization. Next Enterprise Architecture will be discussed, based on information provided by the interviewees, so subsections may slightly differ between the cases. Following the EA description, the IT development will be discussed, also with information provided by the interviewees, resulting in varying subsections. Finally a section about friction between EA and IT development will end each of the cases.

The first case, regarding Rabobank WRR is more extensive, as it begins with sections about the history and general information regarding Rabobank WRR, in order to provide background information and context about the organization. An organogram is also created for Rabobank WRR, while it hasn't been created for the other cases. These extra sections are added and other sections are more elaborate because Rabobank WRR is the main focus point of the case study and therefore a more elaborate description of the organization was in order.

4.1 Case 1: Rabobank Wholesale, Rural & Retail

As the main focus of the case study research, Rabobank WRR, information was gathered extensively. This part will begin by discussing a short history of Rabobank WRR, followed by general information on the current organization. An overview of the company will be provided through an organogram, with a short description of the various departments and domains mentioned in the organogram. Following the organogram, the information gathered will be presented, structured through the main themes of the research: Enterprise Architecture, (Agile) Software Development, the agile way of working and conclude with information provided regarding friction between Enterprise Architecture and (Agile) Software Development.

Most of the information is gathered from interviews and, where possible, information on the same subject found on the intranet will be discussed. The interviews proved to be the main source of information, as only access to the intranet on staff level was provided while the intranet used in the domains was not accessible. Furthermore, information on the intranet was scarce: some (outdated) documents regarding processes and policies were found. It became clear that the intranet was mainly used for document storage for Enterprise Architecture deliverables (created by the domains) that needed to be discussed and saved for future reference.

4.1.1 History

At the end of the 19th century, the first Dutch local banks established two umbrella organisations: the Coöperatieve Centrale Raiffeisen-Bank in Utrecht and the Coöperatieve Boerenleenbank in Eindhoven. These central banks merge in 1972 and become Rabobank in which all local banks are members and shareholders, each having their own banking license. This remained so until January 2016, when the remaining local Rabobanks merged and started working with one central banking license [47].

During the second half of the 20th century, advancements in computing technology and ICT enable the Rabobank to cater to non-agricultural and even individuals as well. The product portfolio expanded from savings and loans to home mortgages, payment services, investments, insurances and leasing. Clients started to operate more internationally toward the end of the 20th century, so from the early 1980's Rabobank started to cater towards these needs as well. Initially for business clients from the Netherlands doing business internationally, but later on also for foreign business clients and even individuals [47].

4.1.2 General Information

As described, Rabobank offers its clients a lot more than just banking products and services and therefore has a set of several subsidiaries and associates such as Interpolis (insurances) and De Lage Landen (leasing). This entire group of banks, subsidiaries and associates is called Rabobank Group and operates in 40 countries, serves 8.6 million clients through 403 foreign places of business with 49,971 employees [49] (internal and external in FTEs). Its mission is to be a "leading customer-centric cooperative bank in the Netherlands and a leading food and agri bank in the world" [48]. Rabobank has split its banking efforts in two separate entities: Rabobank Netherlands, which is headquartered in Utrecht, and Rabobank Wholesale, Rural & Retail (Rabobank WRR), which is headquartered in Utrecht and London. Rabobank Netherlands services the Dutch market alone, Rabobank WRR services the rest of the world.

During the first six months of 2016, Rabobank WRR had 9,253 employees (8,505 internal and 748 external in FTEs), a total income of 1,752 million euros and realized a net profit of 387 million euros, resulting in 475.7 billion euros of total assets. Compared to a total of 686.6 billion euros of total assets for the entire Rabobank group with a total of 49,971 employees (internal and external in FTEs) [48], Rabobank WRR is an important part of Rabobank Group.

4.1.3 Organizational Structure

It is important to notice that this research started in 2015, before the latest organizational structure (effective June 2016 [46]) was in place. The information gathering ceased in May 2016 and

therefore will not take into account the latest and current, but rather the organizational structure in place during the case study research. The current organizational structure is completely different: Rabobank WRR and Rabobank IT Netherlands are now positioned as one entity directly under one of the Executive Board (EB) members. Architects from both organizations are placed under the same EB member. This is all irrelevant for the case description, but merely provided as extra information, as a few initiatives were already taken towards this merge (such as EA principles being merged between Rabobank WRR and Rabobank IT Netherlands).

Only the organizational departments discussed in this research are depicted in the organogram, in order to prevent an overly detailed organogram. It nevertheless provides the scope and relationships between the various discussed departments and places each in perspective of one another. The following sections will briefly discuss the departments mentioned in Figure 5 Organogram Rabobank.



Figure 5 Organogram Rabobank

Executive Board

The executive board is chaired by Wiebe Draijer and is responsible for defining and achieving the business objectives, the strategy and the profit performance. It is also responsible for ensuring compliance with all the relevant laws, regulations, controlling the risks connected with the

business activities and financing the activities. Ralf Dekker is also a member of the EB, as Chief Operational Officer. His portfolio consists of ITOPS, IT Netherlands and Operations Netherlands [48].

IT & Operations (ITOPS) Wholesale, Rural & Retail (WRR)

ITOPS WRR, commonly referred to as just ITOPS is responsible for supporting the international banking activities of Rabobank, by providing IT & Operations to all regions in which Rabobank is active, with the exception of the Netherlands (which are separate departments within Ralf Dekker's portfolio). Anton Rutten, who reports directly to Ralf Dekker in the Executive Board, chairs ITOPS. In time Rabobank intends to consolidate these three portfolios into one, thus merging the domestic and international oriented branches.

Business Architecture (BA)

Business Architecture is a staff department situated directly under ITOPS, as part of the Enterprise Architecture effort in place for Rabobank WRR. The department consists of solely Business Architects and reports directly to the ITOPS Management Team (MT). The department was formed in 2015 by detaching the Business Architects from the IT Architects, who remained positioned as a staff department within IT Systems & Development.

The Business Architects aim to provide the ITOPS MT with a proper understanding of the business needs and reduce the system landscape complexity. Furthermore they aim to ensure a cost efficient ITOPS operation and challenge initiatives on their added value for the organization. Other tasks are to improve transparency and communication: how does each program/project fit into the bigger picture; projects need to be viewed as business change projects not mere technical initiatives; and finally ensure to improve the realization rate of large projects. (J. Bekkers, FINAL Report IT Architecture 2012, internal document, July 30, 2012)

IT Infrastructure (ITI)

IT Infrastructure is the department within ITOPS and is responsible for the infrastructure component of IT, chaired by Rob Klomps. It is on the same hierarchical level as IT Systems & Development and together they provide the complete IT portfolio needed by the other departments within ITOPS in order to conduct their own business (which are amongst others providing banking, financial, leasing and agriculture solutions to customers around the world).

Departments in ITI are Global Core Services (GCS) and DataCenter Technology Services (DCTS). (M. A. van Grondelle, interview, March 10, 2016) For the remainder of this research these departments are not treated separately, rather ITI will be treated as one entity (along with its architects), as it interfaces with IS&D, which is the main subject of this research.

Global Architecture

Global Architecture is a staff department within ITI, in which infrastructure architects are situated. (J. A. A. Klunder, interview, February 17, 2016) This department is responsible for the overarching architecture regarding the infrastructure component.

IT Systems & Development (IS&D)

IT Systems & Development is the department responsible for the IT Systems (e.g. applications and software) within ITOPS and is chaired by Dick Vlot. The department mainly works in a supply demand way, where the business is the client, needing solutions, services and support for their business and IS&D is the supplier of said solutions, services and support. In rare occasions IS&D is the initiator of projects, in order to optimize the IT landscape.

IS&D is responsible for the entire IT application landscape, which the international branches (regions) of Rabobank can use. IS&D has several internal domains defined in order to supply the regions with their needs. These domains are 1) Financial Markets, 2) Corporate Banking & E-Commerce and 3) Risk & Finance. Various staff departments support these domains: 1) APS, 2) Programme Management and 3) Business Management.

Under IS&D three domains exist to cater the regions' needs: Financial Markets (FM), Corporate Banking & E-Commerce (CB) and Risk & Finance (RF). These domains each focus on specific types of

IT solutions, such as trading solutions (FM), banking solutions (CB) and auditing and reporting solutions (RF).

Architecture & Professional Services (APS)

Architecture & Professional Services is a staff department with several disciplines in order to support the domains, ensure alignment and high quality of services and products. Disciplines within APS are: 1) Process Management & Quality Assurance 2) Security 3) Functional Application Management & IT Tooling and 4) Architecture.

Process Management & Quality Assurance is responsible for aligned and coherent processes across IS&D and ensuring high quality of products delivered by IS&D, while Security is responsible for ensuring secure products through predefined requirements set for products and auditing products before they are launched. Functional Application Management & IT Tooling is concerned with providing Product Owners for products used by the domains, so internal clients rather than the business as a client. Finally Architecture is responsible for the entire application landscape within ITOPS, which is mainly done through in an Enterprise Architecture kind of way where new projects are discussed and reviewed before they start to ensure that they deliver a product that fits the envisioned future state of the landscape.

Programme Management

This staff department is responsible for alignment of bigger programs within the regions. It also makes sure that IS&D follows the course set for the future by the Executive Board and the ITOPS MT. The vision, mission and strategy of Rabobank is first translated into a vision, mission and strategy for ITOPS, which is then translated into a vision, mission and strategy for IS&D. These need to be in line and traceable both downwards as upwards. The programs in the domains then need to adhere to these and make sure they are not in conflict with them.

Business Management

Business Management is responsible for every day operations. There's a secretary, a communication officer and business manager, providing services to APS, so they can provide their services to IS&D.

As can be seen from the departmental descriptions, the architects are placed decentralized in the organization. Business Architects are placed directly under the MT of Rabobank WRR, the IT/Application Architects are placed directly under the MT of IS&D, Infrastructure Architects directly under the MT of ITI and the Domain Architects are placed in their respective domains (FM, CB, R&F). This results in functional reporting lines instead of hierarchical ones.

4.1.4 Enterprise Architecture

While each of the domains (FM, CB and R&F) is individually big enough to be able to be seen as an organization on its own and would therefore be able to organize its own EA if so desired (P.J. Morssink, interview February 2, 2016) Rabobank WRR aims to manage EA across the entire enterprise (Rabobank WRR) and focus the EA effort in an overarching way. (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015) The discipline is relatively new within Rabobank WRR (the development of the Rabobank WRR framework started in 2009) and started out with the aim to guide the organization using a set of processes and deliverables. (J. Bekkers, FINAL Report IT Architecture 2012, internal document, July 30, 2012)

Lately (2014) the MT of ITOPS realized it needed a more strategic EA discipline, so the Business Architecture Team was initiated. (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015) EA within Rabobank WRR is regarded of consisting of three components: "Governance", "EA Process" and "Deliverables" (R. Zwart, Progress Review, December 12, 2015; J. Bekkers, FINAL Report IT Architecture 2012, internal document, July 30, 2012 & ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015):

Governance sets the **boundaries for the EA Process**.

The **EA Process** describes how the **Vision** of the bank **leads to a Business Strategy** (e.g. Business Line Objectives, Business Cases), which **is translated to a Business Architecture** (e.g. descriptions of Processes, Organization), which in turn **is translated into Data, Application and IT Architectures** (e.g. Data Models, IT RAs), which **finally lead to Programmes/Projects** (PSAs).

Deliverables are **instantiations of architectural documents** that support the EA Process.

Each of the domains within ITOPS is governed under EA (which means they need to adhere to the artefacts (bricks and principles) as well as the governing processes), but is left free to fill in the specific details of EA for its own domain. (P.J. Morssink, interview February 2, 2016)

4.1.4.1 Governance

Governance enables Enterprise Architecture at Rabobank WRR, but also sets boundaries within which teams are allowed to maneuver. This is done through various means: 1) the way architects are organizationally placed in the organization; 2) Roles and responsibilities of architects; 3) EA artifacts 4) governing processes 5) architecture meetings

Organizational placement of architects

Section 4.1.3 described how the architects are placed decentralized in the organization: the Business Architects are placed just under ITOPS MT. The IT/Application Architects are placed as a staff department under IS&D MT, the Domain Architects are placed in their respective domains. The Infrastructure Architects are placed as a staff department under ITI.

The Business Architects are positioned so far up the hierarchical chain that the distance to the operational workplace is huge. Because of this, the operational workplace doesn't notice Business Architecture. Of course they are at enterprise level, they can't actively be involved in all layers, but the distance is too big this way. You'd expect them to have a mandate regarding the architecture of solutions this high up the hierarchy, but they don't. Business demands may outweigh the architectural issues from time to time, but there should be a balance. (R. Zwart, interview, March 1, 2016)

Roles and responsibilities of architects

The Business Architects are responsible for the general Business Architecture (e.g. Business Domain Reference Architecture, Target Operating Models) and for Data Architecture (governance, ownership, dictionary definitions). (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015)

The IT/Application Architects' responsibilities are the Data models, Data warehouses, ETL, IT Reference Architectures, PSAs, Technology Bricks (applications) and the Application life cycle management. (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015) furthermore they provide the IS&D MT with advice, govern the Principles and be the linking pin between the domains (R. Zwart, interview, March 1, 2016). They also function as the main point of contact for each of the various regions. They take place in the architecture meetings within each of the domains that they're assigned to, to stay informed of developments within the domain. (E. de Nies, interview, February 15, 2016)

The Infrastructure Architects are responsible for the Technology Bricks and the Technology life cycle management. (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015)

In each architecture department and domain there is one architect appointed as Lead Architect. This role gives the architect a few extra organizational responsibilities, such as HR tasks (hiring, firing and evaluation) and being the main point of contact towards the rest of the organization. (P. J. A. Morssink, interview, February 23, 2016)

Artefacts

- *Principles*

Principles are a set of guidelines/rules, which describe the preferable choice in certain situations to help make decisions when creating a PSA. (P. J. A. Morssink, interview, February 23, 2016)

They are being revised, aligned and merged at the moment. First the principles of IS&D and ITI were made into one set, now they're even moving to one single set of principles for the entire Rabogroup: ITOPS and Rabobank Netherlands will have a joint set of principles. There can be negative effects regarding the merging of principles: they are going to be very general, in order to be fit for the entire Rabogroup, no one has ownership of the principles, as they are a joint undertaking. This is a risk because principles should be accepted and adopted by the team that has to adhere to them. (J. A. A. Klunder, interview, February 17, 2016)

- *Technology Bricks*

Technology Bricks are a set of technology standards, describing which technology is being preferred over others, keeping long term usability in mind: are they strategic (will last 5+ years), tactical (to be used up to 3 years), containment (only applicable in certain settings/areas) or retirement (to be replaced in 2-3 years). These also help put together a PSA, as they guide the projects regarding preferred technology. (Global ITI or IS&D Architect, Technology Bricks Process description, internal document, June 2012)

Governing Processes

Governing processes are processes in place with the goal to uphold the integrity of the architecture deliverables and artefacts as the EA activities are carried out.

- *Technology Bricks*

The Technology Bricks process was devised in 2012. It describes the process around technology bricks, which are essentially technology standards. The policy, activities and responsibilities involved in the process of maintaining and applying technology standards are formalized. (Global ITI or IS&D Architect, Technology Bricks Process description, internal document, June 2012)

Nobody seems to want to actually maintain the Technology Bricks. There is not a single owner of all Technology Bricks; rather teams are responsible for them. (P. J. A. Morssink, interview, February 23, 2016 & H. Dijkstra, interview, February 25, 2016) Revising them should be done once every 18 months, architects have the assumption that it has to be once every 12 months. But in an audit conducted in 2013 within ITOPS regarding architecture, it was discovered that 80 out of the 98 Technology Bricks were not reviewed in the last 18 months. (J. Bekkers, FINAL Report IT Architecture 2012, internal document, July 30, 2012) This may be caused by the fact that the bricks 'belong to everyone' at the moment, there is not a single owner of them, that makes keeping the bricks up to date and pragmatic harder: there's a lot of bureaucracy around the bricks.

- *Application Lifecycle Management*

A process devised in 2010, describing the Application Life Cycle Management (LCM) process of Rabobank WRR, so that a better control of the state of their application landscape can be achieved. (Global ITI or IS&D Architect, Application LCM Process Description, internal document, September 2010)

- *Project Start Architecture*

Project Start Architecture is a process finalized in 2013, describing the process followed for Project Start Architectures (PSAs), or the exempt form for PSAs. It describes how PSAs are reviewed, discussed and approved. It revolves around the WRR Architecture Board, which is a weekly meeting chaired by the IT/Application Architects, and attended by Infrastructure

Architects and the Domain Architects (the business is not represented). (Global Head IS&D S&A, PSA Process Description, internal document, October 2014)

A PSA has to be constructed if a Project Initiation Document (PID) is needed. This is the case if a project costs more than €500.000,=. Once such a PSA is approved, the project can officially start. If the project is for example under a lot of time pressure, a PSA exempt form can be submitted, which allows the project to go ahead and deliver a PSA in a later stadium, to ensure that the architecture models are updated (Global Head IS&D S&A, PSA Process Description, internal document, October 2014)

- *Deviations*

Deviations is a process finalized and approved in 2013, which describes how deviations in architecture are handled. It elaborates on capturing and maintaining deviations, how they are fed into the budget process and how they are reported upon. (IS&D Strategy & Architecture department, WRR Architecture Deviations Process, internal document, June 2015)

- *Application Register Process*

A draft version, from 2011, a process describing how applications will be registered in a standardized naming register containing unique names for applications. As authors of PSAs and RAs which deal with positioning of IT in the IT landscape, the IS&D architecture community is the organizational body to own the naming of the business applications. (IS&D Strategy & Architecture, Global Application Register Process Description, internal document, November 2011)

- *Architectural Design Decisions*

Architectural Design Decisions is a process from 2014, which formally describes the global process of Rabobank WRR for design decisions with architectural impact, delivery teams may make after the PSA has been approved. If during the course of the project it becomes apparent that the end result is going to deviate significantly from the originally proposed solution in the PSA, the PSA should be revisited, adjusted to reflect the current expectation of the end result and resubmitted to the board. This process leaves room for the architects to some level of own judgement whether or not the change is significant enough to have the PSA changed and be resubmitted. (IS&D Strategy & Architecture department, WRR Architecture Deviations Process, internal document, June 2015)

A process that checks whether projects are developing according to the approved PSA or not is missing. (M. A. van Grondelle, interview, March 10, 2016) Officially there is a checklist (the process Architectural Design Decisions) to check whether deviations are significant enough to be obliged to adjust and resubmit a PSA, but there are no controlling processes that checks whether or not the projects (or individual developers) are violating this checklist or not. (R. Zwart, interview, March 1, 2016) Also a Project End Architecture process and template are being devised at the moment, but this has not been finalized or implemented yet. (<author unknown>, Project End Architecture Template, internal document, April 2016)

Another process that might be relevant is some sort of Request For Change process regarding architecture: in this board you would be able to look at changes in the architecture landscape (e.g. direct connections/links between applications instead of using API's) that need to be addressed. Then you can make a roadmap on what changes are needed in order to improve the architecture landscape of the organization. (M. A. van Grondelle, interview, March 10, 2016)

Architects mention that knowledge sharing is important within the architectural domain in Rabobank WRR, but it is not a specified process. (J. B. Landman, interview, February 24, 2016) The new CAT meeting fits this process of knowledge sharing: it is chaired by the IT/Application Architects who provide information regarding various subjects and issues, but also information from the domains is shared regarding developments and insights. (E. Woudt, interview, March 4, 2016 & H. W. C. Thoonen, interview, March 8, 2016)

Architecture Meetings

To execute and facilitate these processes and general knowledge sharing within Rabobank WRR among architects, several meetings are scheduled on a regular basis: (<author unknown>, Meetings, internal document, <date unknown>)

- WRR Architecture Board: a weekly meeting, responsible for reviewing and approving architecture deliverables with global or HQ relevance
- Central Architecture Team: a bi weekly meeting, which focusses on startup and coordination of architecture initiatives. It consists of the IT/Application Architects, Lead Domain Architects and the Lead ITI Architect. There are two alternating objectives of this meeting: the first is to align the domains: what programs/projects are happening in the domains, how are they impacting on each other and what do the other domains need to know? The second is to share knowledge: one domain gives an extensive presentation on internal developments within the domain.
- Extended Central Architecture Team: a monthly meeting of all architects focusing on knowledge sharing.
- Regional architecture calls: monthly calls between the IT/Application Architects and regional architects to attune on architecture topics.
- WRR Global Architecture Conference: a biannual conference of the Rabobank WRR Architecture Community.

It is often mentioned that the placement of architects throughout the organization is not optimal and sometimes even a restricting factor, as the different architectural departments are not interwoven, which would help alignment between them. (R. Zwart, interview, March 1, 2016 & M. A. van Grondelle, interview, March 10, 2016) The current situation places the architects in silos, which hinders communication, as there are no formal reporting lines (be it hierarchical or functional) between them. (H. W. C. Thoonen, interview, March 8, 2016 & M. de Haan, interview, March 10, 2016)

There is a general consensus that all architects (Business, IT/Application, Domain and Infrastructure) should be in one single department, in order to be able to maintain objectivity. (H. W. C. Thoonen, interview, March 8, 2016 & E. Woudt, interview, March 4, 2016)) This means that there would be a hierarchical reporting line between the architects instead of a functional one, so the architects do not need to please their manager, but can act in the best interest of the Enterprise Architecture. (H. W. C. Thoonen, interview, March 8, 2016)

There is no governance process regarding principles, which shouldn't be a problem as they are set for a longer period of time. An annual review and revision shouldn't be needed, rather they should be updated whenever the need for it is felt. Perhaps this could be a recurring subject during the WRR Global Architecture Conference (which is held once every two years), which all Rabobank WRR architects attend.

The Technology Bricks are posing more of a problem: technologies change more regularly: e.g. updates, retirement of technologies/solutions/applications, advancing insights (lessons learned) into what solutions are desirable, changing Reference Architectures. The revision of each Technology Brick should be done every 12 months, but this process takes place irregularly, nobody feels ownership for them: each Technology Bricks is owned by a team. This is not a very pressing issue, however, as interviewees mention that the Technology Bricks are adequate. (P. J. A. Morssink, interview, February 23, 2016)

Several of the governing processes are not even finalized (and in this state for a few years already), no updates to them are being made to them, no controlling processes, activities or entities are in place to check whether they are actually being carried out regularly and/or when needed. Some processes (Project End Architecture, a Request for Change process and an official Knowledge Sharing process) are being missed by architects.

Overall, the governance of EA (organizational placement, responsibilities, artefacts, processes and architectural meetings) are present. However, some components are missing and some can be improved: especially the placement of architects within the organization, controlling/policing activities and the missing processes should be addressed.

4.1.4.2 Process

The actual process description of Enterprise Architecture is not very elaborate and detailed yet, as the discipline is still very young within Rabobank WRR. Before 2014 the EA discipline within Rabobank WRR was actually an IT Architecture effort, designed by IS&D and ITI. (J. Bekkers, FINAL Report IT Architecture 2012, internal document, July 30, 2012) Around 2013/2014, the ITOPS MT started recognizing “the need for more upfront business problem and solution analysis and proper understanding what the business challenges and underlying problems are.” This realization led to the initiation of the Business Architecture department, in order to “support the ITOPS MT in assessing business challenges, underlying problems and new initiatives, but also to determine and validate required solution directions.” (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015)

The challenges for the Business Architects are the following (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015):

1. “Ensure an effective ITOPS operation with a proper understanding of the real business needs, information/data flows, organization and strategy, where operating models are more standardized across the international operation, and a clear roadmap to reduce the system landscape complexity.”
2. “Ensure a cost efficient ITOPS operation where new business or support initiatives are challenged on their value add and potential alternative solutions, where existing businesses or systems are reviewed on their total cost of ownership and where more consistent cost/benefit analyses across business lines are developed.”
3. “Ensure improvement of transparency and communications to the business and individual locations on how each program/project fits into the bigger picture and how these interact with local schedules.”
4. “Ensure that projects are not viewed as technical initiatives but more as business change projects with a clear emphasis on problem and feasibility analysis, solution design(s), objectives and measures (the WHY).”
5. “Ensure improvement of the large project realization rate.”

Business Architecture defined objectives for Business Architecture (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015):

- Support ITOPS Management in establishing a structural cost reduction
- Understanding business needs, regulatory requirements and technical possibilities
- Building bridges across business lines/ regions/ functions
- Improved resource planning and expectations management
- Developing solutions first time right –delivering projects within agreed scope, time and financial budgets.
- Consistent, efficient and effective operating models

Business Architecture aims to deliver an “integrated & cross domain functionality model”, “consistent Target Operating Models” and “Business Fact Sheets”. The Business Architecture Roadmap is still empty, but the canvas shows “Target Operation Models”, “Business Reference Architecture” and “Business Fact Sheets”. (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015)

For 2015 they had formulated several goals in four topics (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015):

- *Client Focus*

Goals:

1. Work towards ITOPS environment that is transparent and predictable for clients
2. Create business-led architecture

Actions:

1. Support in setting out the business requirements, the definition, implementation and maintenance of the ITOPS strategy in relation to the overall business strategy

- *People*

Goals:

1. Increase knowledge and insight in ITOPS landscape (across the chain) and its opportunities to improve
2. Encourage interaction between people, departments and locations for optimal input for Business Reference Architecture
3. Encourage people to be part of solutions, participate in continuous improvement

Actions:

1. Pro-actively inform and involve people (and use network) to create understanding and support for need for improvement.
2. Show genuine interest in each other's drivers and viewpoints

- *Run the Bank*

Goals:

1. Give insight in current global WRR landscape
2. Create sense of urgency to improve current landscape

Actions:

1. Create total overview current global landscape (including Operating Models)

- *Change the Bank*

Goals:

1. Create insight in external trends, client trends, market trends and business needs to determine the way to improve.
2. Create insight in ITOPS landscape (across the chain) and its opportunities to improve
3. Translate strategy into initiatives
4. Give input for ITOPS strategy decision making

Actions:

1. Providing support in assessing business challenges, underlying problems and new initiatives
2. Determine and validate (against strategy) required solution directions.,
3. Create global operating models (TOM), the business and data/information architectures and roadmap planning for the bank's international operation

Business Architecture (and the other architecture disciplines) is (are) related as followed to the organization: the bank has a certain Vision: “what do we want to achieve as a bank”, the Business takes this Vision and turns it into a (Business) Strategy by defining for example specific Business Line Objectives and Business Cases. This Strategy is transformed by the Business Architects into a Business Architecture, which explains how Services, Processes, People, Organization, Technology/Data and Management Information help the Business execute their (Business) Strategy. The Business Architecture is input for the IT Architecture, which describes what Systems and Infrastructure is needed in order to support the Business Architecture. The IT Architecture, finally is input for the Projects/Programmes. (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015)

Finally, an Enterprise Architecture is defined and the various architects are mapped onto it (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015):

- Business Architects focus on “Business Architecture” and “Data Architecture” (more towards the governance);
- IS&D Architects focus on the rest of “Data Architecture” (the Data models, Data warehouses and ETL), as well as “Application Architecture”;
- ITI Architects take on “IT Architecture”.

Experiences with the EA Process

ITOPS MT views the Business Architects as a link to the business, not as a first step in the EA. This results in the fact that Business Architects focus on the ITOPS level, they don’t go into more detail by defining an applications or infrastructure landscape. The TOM of the business and the TOM of ITI should be linked through some kind of dependency: there needs to be a route from top (business) to bottom (infrastructure), but also the other way around. You don’t see that link. This results in Project Architecture not being based on a general EA. There’s no cooperation when defining an application architecture and an infrastructure architecture, this should be aligned, through components and building blocks. Communication and alignment is important, you shouldn’t work in silos (business-application-infra). (M. A. van Grondelle, interview, March 10, 2016) Business Architects, IT Architects and Infrastructure Architects never meet: how should knowledge sharing and alignment happen that way? (M. de Haan, interview, March 10, 2016)

The architecture at Rabobank WRR is not Enterprise Architecture: there’s not a good link between the business and the architecture effort. This is also shown in the Enterprise Architecture board (chaired by APS): the business is not represented. (E. de Nies, interview, February 15, 2016) the architecture is more about application level. A mapping from Business Architecture to Information Architecture is not present. We’re working more IT focused, than business focused. (R. R. Mammen, interview, February 19, 2016) You should think about architecture starting from processes and the business; and devise solutions while working from those two. You can’t devise a solution based on IT and then try to fit the business into it. The non-IT side is too weak at EA at the moment. You don’t really see a lot architectures from the Business Architects, it’s mainly a capacity issue. (H. W. C. Thoonen, interview, March 8, 2016) The domains don’t get from the Business Architects what you’d expect from them, so the R&F business is own Reference Architectures. (K. P. Boerema, interview, March 15, 2016) Within domains there are no Business Architects either, so Lead Architects sometimes take on the role and activities of Business Architects: for example define products or model processes. The rest of the architectural processes in domains are very operational; they focus on design architecture for projects. That’s not Enterprise Architecture. (R. Zwart, interview, March 1, 2016)

Perhaps the Business Architects should be placed even higher than ITOPS to have mandate: place them under the COO, so that the EA discipline will span Rabobank WRR, and Rabobank IT Netherlands. (K. P. Boerema, interview, March 15, 2016) The importance of architecture is recognized within the organization, at least within IT. The business doesn’t really acknowledges the importance of architecture. (E. Woudt, interview, March 4, 2016) You should get the business on board by making them understand why you’re organizing the architects like this: it is important that the business knows that EA needs them as well. (E. de Nies, interview, February 15, 2016) It

would also be best if all architects (Business, IT/Application and ITI) are pooled together and be permanently sourced to the domains/programs. Maybe even merge Rabobank WRR and Rabobank IT Netherlands. (E. Woudt, interview, March 4, 2016)

The experiences regarding the EA process report about the missing link between the business and IT. EA should start with the business strategies and goals, translate them to TOMs and RAs (target states), create information and IT architectures based on them (current state and target state), which should lead to a roadmap towards the target EA landscape. This roadmap should be the basis for projects. This way you would get Now it's the business initiating projects by themselves, without consideration for architecture, which leads to a scattered architectural landscape. (P. J. A. Morssink, interview, February 23, 2016)

Often the situation at Rabobank Netherlands is referred to: have the architects sourced to the domains, where they take place in a triangle meeting with the business change manager and program/domain manager. (R. Zwart, interview, March 1, 2016 & E. Woudt, interview, March 4, 2016) This way you could position yourself as an equal discussion partner in order to create a balance between the IT strategy and where the domain is actually moving towards. (R. Zwart, interview, March 1, 2016) If such a construction is desirable for Rabobank WRR as well, the business will have to be informed, convinced of the importance to such an organization of architects and to align each other on the strategical and tactical level. (E. de Nies, interview, February 15, 2016 & C. P. W. van Zuijlen, interview, March 4, 2016) This would greatly help if the architects get acknowledged by and a mandate from the MT, ideally the ITOPS MT, perhaps even under the COO, in order to really create an EA discipline at Rabobank (WRR, ITN and Operations NL combined) level. (K. P. Boerema, interview, March 15, 2016)

4.1.4.3 Deliverables

Business Architecture

Delivered by the Business Architects; describes the business strategy, processes, products, organization and what the business functions of the organization are (or should be), independent of the organizational structure, in order for the business to achieve its strategy and achieve its goals. (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015)

Target Operating Models

Target Operating Models (TOMs) for Rabobank WRR, in which the services, processes, people, organization, technology/data and management information that are needed to be able to serve ITOPS' clients, are being constructed by the Business Architects. TOMs explain how the Business These TOMs are announced, and in a very preliminary phase. These TOMs will be based on the (Business) Strategy, devised by the business. (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015)

The business side of R&F creates their own TOMs (E. Woudt, interview, March 4, 2016). IDBR is also working on its own (living) TOMs; they are connecting those to the Reference Architectures. (J. B. Landman, interview, February 24, 2016) ITI is also missing a link in TOMs: the Business TOM and the ITI TOM should be linked and traceable, but they are not. (M. A. van Grondelle, interview, March 10, 2016)

Information Architecture

Information Architecture describes the information needed for the TOMs (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015) and how it is treated and placed within the organization: Common Data Standards; Master Data Management; and Management Information. (IS&D Strategy & Architecture, WRR Reference Architectures Introduction, internal memo, October 2011)

Even though the IT/Application Architects are responsible for the Data Models, Data Warehouses and ETL, they are hardly mentioned during interviews. This is because Data Architecture never really took off: the Data Governance Board that is established hardly produces results, which leads to the fact that the problems remain evident within the domains. (H. W. C. Thoonen, interview,

March 8, 2016) Data management proves to be difficult to embed in the organization, because of the way it is delegated: no one is owner of the data. (P. J. A. Morssink, interview, February 23, 2016) Data Architecture is a joint responsibility for the Business Architects and the IT/Application Architects, but the implementation of this architecture discipline is lacking, as is evidenced by these statements.

Application Architecture

The Applications Architecture is the current state of the application landscape within WRR. The Application Architecture enables the organization to execute it's processes (defined in the TOMs) by providing them with (information) systems. (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015) The Application Architecture comprises, amongst others the application landscapes and roadmaps (IS&D Strategy & Architecture, WRR Reference Architectures Introduction, internal memo, October 2011)

Infrastructure Architecture

The Infrastructure Architecture describes how the hardware is organized in order to support the Application Architecture, e.g. through Technology Lifecycle Management, Infrastructure Landscape Models. (ITOPS Business Architecture, Business Architecture Booklet, internal document, 2015)

Project Start Architecture

Project Start Architectures (PSA) are the default deliverable regarding EA for projects. It describes the basis of a project (Global Head IS&D S&A, PSA Process Description, internal document, October 2014):

1. Introduction: the goal and the scope of the project, the impacted functional domains and an assessment of the proposed solution against the principles.
2. Functional Architecture: the functional architecture and the processes are described. Also how the project relates to the Reference Architecture (RA) is explained.
3. Application Architecture: the application architecture current state and project end state are depicted, the changes described and design decisions explained.
4. Technical Architecture: describes the technology choices that have been made, the security attention points and the infrastructure requirements and attention points.
5. Documentation control: information regarding the history of the document, the sign offs and references.

The PSA is the bare minimum to start a project, it is expected that projects create more detailed documents during the course of a project. Software Architecture Documents, Design Documents and Infrastructure Documents. This is often done in an unstructured way, different for each domain. (R. Zwart, interview, March 1, 2016)

Project End Architecture

After a project finishes, there are no architectural deliverables that need to be created (like a Project End Architecture for example. (P. J. A. Morssink, interview, February 23, 2016)

As mentioned (in Section 4.1.4.1 Governance), this is in the process of being devised; a draft version is already available, but not yet approved. It will address deviations made by the project regarding the solution in the PSA. This Project End State should also contain a list of design decisions made within the project: assumptions and rationales that lead to these decisions. Lessons learned would also be a good addition to such a Project End State. (P. J. A. Morssink, interview, February 23, 2016)

Domain Reference Architectures

The Domain Reference Architectures describe the (current & future) Business and IT architecture in Reference Architecture documents. These RA's serve as a framework to provide guidance to project portfolio management and to the Project Start Architectures. They describe the domains through

Business Architecture descriptions; Information Architecture descriptions; Application Architecture descriptions; and Infrastructure Architecture descriptions. (IS&D Strategy & Architecture, WRR Reference Architectures Introduction, internal memo, October 2011)

Other models of domains are devised in a federative way: everyone has their own models, landscape views, and so on, they are linked loosely together. (J. A. A. Klunder, interview, February 17, 2016) There are some sketches of current states for R&F and CB. Of course these current states are for a large part present in the CMDB: applications are registered here, interfaces to a limited extent, perhaps you can make or acquire a tool to be able to see an up to date current state based on this CMDB. (P. J. A. Morssink, interview, February 23, 2016)

While a lot of architectural deliverables are defined and described, not all are present, updated or (publicly) available in the organization. The overall Business Architecture and TOMs are not yet devised or unavailable for ITOPS, those are partially present on the IS&D level, but no clear link to actual business strategies is present. Projects are not initiated based on a roadmap created by comparing the current and target state architectures, but rather through individual efforts and business requests. Projects should be initiated based on the roadmap regarding closing the gaps between current and target state architecture. (M. A. van Grondelle, interview, March 10, 2016)

The main missing deliverable (for now) is a Project End Architecture, which describes the actual developed solution, which can be used to update the (new) current architectural state, in order to keep the gap analysis and roadmap up-to-date. Other reported missing deliverables are more project based deliverables: technical and functional solution designs, decision documents, etcetera. (H. W. C. Thoonen, interview, March 8, 2016 & J. B. Landman, interview, February 24, 2016)

4.1.5 Software Development

Rabobank WRR aims to develop its software in an agile way. There is no official process the development teams have to follow; as the agile philosophy is that no team is the same. This means that what works for one team, doesn't necessarily have to work for another team. (M. de Haan, interview, March 10, 2016) A lot of variance therefore exists between domains and teams. Perhaps there should be more alignment between teams by standardizing some processes to provide teams with handles on how to approach Agile Software Development. (C. P. W. van Zuijlen, interview, March 4, 2016) The MT only hears success stories of Agile Software Development though, so they think everything is going well. (R. Zwart, interview, March 1, 2016)

To enable the organization to adopt agile software development, agile coaches are present within the organization. These coaches support the individual teams, educate the employees (not just developers) on general principles of agile software development and how Rabobank WRR has adopted and adapted agile software development within the organization.

Roles within Agile Software IT Development

Several roles can be distinguished when working in an agile way. The most notable ones are Product Owners, Scrum Masters and Stakeholders. Project Leaders are officially not a role within and agile way of working, but in Rabobank WRR these are still commonly used. (K. P. Boerema, interview, March 15, 2016)

The PO has to have a mandate from the business to act on their behalf. Product Owners should come from the business, to act as a representative from the business. Prioritizing often still happens in accordance with the business stakeholders through a Change Board for example. Rabobank WRR is still struggling with the Product Owner role though: one of the tasks of the Product Owner should be to create epics (large user stories that are not yet broken down into small user stories that can be used as work packages), but in reality it's often a Business Analyst that creates these epics. (C. P. W. van Zuijlen, interview, March 4, 2016; H. W. C. Thoonen, interview, March 8, 2016 & K. P. Boerema, interview, March 15, 2016)

Officially there is no need for Project Leaders in agile working organizations and teams. This is because you should not have a certain time limit for the project. The teams are continuously working in sprints of 2-4 weeks and have a Definition of Done regarding user stories. When the work is done, that user story is done and there is a software increment. When all user stories are done (or when the application is sufficient), the project is done, it should not be time managed. If you have a good Product Owner and a good Scrum Master you wouldn't have the need for a Project Leader. (K. P. Boerema, interview, March 15, 2016)

In reality projects often still have certain budgets, milestones, dates and a moment when the project should be finished. You should work within a certain budget and hire people based on that budget. The bigger the budget, the bigger the team and the more you can do in the same amount of time. (R. R. Mammen, interview, February 19, 2016; H. Dijkstra, interview, February 25, 2016; E. Woudt, interview, March 4, 2016 & H. W. C. Thoonen, interview, March 8, 2016)

Agile Teams

Teams usually remain quite stable; people don't get shuffled around much. The philosophy is: if a team is working well together, why break it up? People know each other's strengths and weaknesses and can compensate for them. In some cases people move to other teams, in an effort to transfer knowledge, but not to break up entire teams. (C. P. W. van Zuijlen, interview, March 4, 2016)

If you're in a product team (maintenance teams, developing changes for running application), you have a PO who prioritizes the backlog with Request For Changes (RFCs). If you're in a project team, POs prioritize user stories in the backlog based on the project targets, where he acts on behalf of the project board or steering committee. The backlog of a project team is filled with project items, formulated as user stories. Finally there are also hybrid teams, who are working as product teams and project team. They have a backlog filled with project items (additions to an application) and maintenance items (RFCs). (C. P. W. van Zuijlen, interview, March 4, 2016)

In some cases applications are so big, there are several teams working on it (be it project teams or maintenance teams), in that case there's an extra team that focusses on integrating and implementing the increments of the different teams. It proves difficult to actually align those team efforts with each other: teams mainly focus on their own development process, alignment between teams is seen as overhead and complicates the Agile SDM. This results in poor architectural alignment and makes integrating the different elements more difficult. (C. P. W. van Zuijlen, interview, March 4, 2016)

Business Involvement

The cooperation and alignment between business and IT is still not happening. Rabobank WRR seems stuck in some sort of supply-demand setting: the business has needs/demands; IT should cater to these needs/demands. An equal partnership where business and IT work together to develop new ideas is an utopia for now. (R. Zwart, interview, March 1, 2016)

In general Agile SDM enables you to see early on if the application you're developing is going to serve the Stakeholders the right way. Within Rabobank WRR this feedback loop is quite weak: Stakeholders (users) should be more involved in sprints (or at least when the sprint is done and the increment is presented). Creating such a feedback loop would also enable you to put checks in place to see whether or not you're deviating from the PSA too much or not. Getting Stakeholders and users involved to a higher degree, enables you to manage their expectations and check if you're going to fulfill the high level planning/roadmap. Managing these risks could be done better within Rabobank WRR. (P. J. A. Morssink, interview, February 23, 2016 & J. A. A. Klunder, interview, February 17, 2016)

Agile or Plan-Driven IT Development

Rabobank WRR is not capable of doing big programs in an agile way. These programs are therefore done in either a plan-driven way, or a hybrid way where agile and plan-driven are combined. What you see for example when buying a big enterprise system is plan-driven in the beginning: massive workshops to get an idea of how to set it up and get an how it should be implemented. When the

basis is there, teams start working on it in an agile way: work on small bits and pieces that can be delivered fast and then be tested. The most important thing is that not everything is created at once, then tested and finally implemented as a whole (pure plan-driven). (C. P. W. van Zuijlen, interview, March 4, 2016)

Hybrid IT Development

Two-speed IT is a possibility in areas where you can't fully develop agile: if large backend mainframes with legacy software are at the base of your systems you don't want to develop on them in an agile way. These mainframes are in maintenance mode anyway, no new code for them has to be written, it just has to be made sure that they don't break down. If you have a virtual layer, or API layer around that mainframe you can develop agile at the frontend. Hence the notion two-speed IT: backend is Plan-Driven and thorough, so slow; front end is agile and iterative, so fast. Alignment is important when working with two different speeds in order to keep everything working. (H. Dijkstra, interview, February 25, 2016)

You really need to consider if you need agile applied to all development efforts: transaction systems aren't continuously developed upon and don't have features added all the time. You don't even want that: they need to stay robust, thoroughly tested and never break down. Extensive requirements, functional and technical designs etcetera aren't a bad thing here. (H. Dijkstra, interview, February 25, 2016 & J. B. Landman, interview, February 24, 2016)

The backend systems aren't the place where Rabobank WRR can distinguish itself from its competitors anyway. This can be done in the middle layer and frontend of the applications. There is a lot more stakeholder and client interaction. Being agile in those layers gives you the ability to cater to their wishes fast. (H. Dijkstra, interview, February 25, 2016)

Documentation

Finally, now that teams are working in an agile way, it seems that software designs are not being created consistently anymore. The most important use cases and typical solutions you created in an application should be modelled, to show the most important functions of an application. Of course you don't have to create models of every single use case and create technical designs of the entire application, but the most important parts should still be documented. This would also include the most important decisions and their rationale as well as patterns and solutions created to ensure the application is working. (R. Zwart, interview, March 1, 2016; P. J. A. Morssink, interview, February 23, 2016; E. Woudt, interview, March 4, 2016; K. P. Boerema, interview, March 15, 2016 & H. Dijkstra, interview, February 25, 2016)

Rabobank WRR has embraced agile IT development, but it is mainly implemented within the software development teams, the infrastructure teams are not working very agile yet. The most important roles are introduced, but from time to time the way these roles are fulfilled or are delegated isn't optimal yet. Coaches are present in the organization to provide advice and guidance towards the teams, stakeholders and the rest of the organization regarding agile practices.

The teams are quite steady, which improves efficiency and team spirit. It also improves how well the team members know each other's strengths and weaknesses. It doesn't improve knowledge sharing between teams, however. Maybe some (junior) team members should be rotated between teams, in order for them to learn from different teams and create automatic knowledge sharing between teams.

One of the main issues with agile IT development revolves around the business involvement. The way IT projects are initiated is still in a supply-demand kind of way, not architecture driven through alignment between the business strategy/ goals and the architectural state of the organization (including its IT landscape). Also the stakeholder involvement in agile IT development projects (be it the product owner or direct stakeholder involvement) is quite weak.

Another issue is the ability of Rabobank WRR to successfully execute big programs. They are struggling with this, which could be solved through a better alignment of architectural models,

analyzing current and target states and creating roadmaps based on this. That would guide you towards a focused IT programme.

Finally documentation seems lacking: e.g. solution designs (functional and technical), deviations regarding the PSA are hardly delivered. It has to be noted however that this is not only the fault of agile IT development: there are no controlling/policing processes in place either that check up on projects to see if they're still working under architecture or not.

4.1.6 Friction between Enterprise Architecture and (Agile) Software Development

There is a natural tension between EA and agile software development is that EA has a long-term focus and wants to standardize/centralize designs, decisions and so forth, while agile prefers a short-term focus and wants to be self-regulating. (C. P. W. van Zuijlen, interview, March 4, 2016, E. Woudt, interview, March 4, 2016 & R. Zwart, interview, March 1, 2016) Those self-regulating teams need space to do so, but from a complexity reducing and efficiency point of view you don't want everyone and everything to be self-regulating. There should be a balance between structure and chaos: if there's too much structure, it's a bureaucracy; if there's too much chaos, you will drown in directionlessness. (C. P. W. van Zuijlen, interview, March 4, 2016) If everyone is self-regulating, you are at risk that everyone just does their own thing and loses sight of context. If those teams become self-regulating and it becomes a free-for-all situation, you will get isolated products, without awareness of context and environment and get a 'spaghetti' of systems. (E. de Nies, interview, February 15, 2016) The challenge with self-regulating teams is to make them aware of the importance of working under architecture. (M. de Haan, interview, March 10, 2016)

At the moment the pressure to deliver is bigger than to improve the organization. (C. P. W. van Zuijlen, interview, March 4, 2016) The business wants to see fast deliveries, while the governing IT side wants to give attention to architecture, security, etcetera. (M. de Haan, interview, March 10, 2016) This may lead to isolated projects, not looking at the context, resulting in bad alignment of projects. (E. de Nies, interview, February 15, 2016 & K. P. Boerema, interview, March 15, 2016) This may also lead to software which can't be generalized because of refactoring. In that situation it'll look as if you're developing fast and efficient, but in the end you get complicated software without much documentation (i.e. technical debt). (R. Zwart, interview, March 1, 2016) If architects approach Product Owners (POs) that an application or system needs maintenance, it might be given a low priority: it is more important to develop new features and functionalities, which leads to pressure on upgrades and middleware. Sometimes POs do listen to input and advice regarding prioritization though. (K. P. Boerema, interview, March 15, 2016 & P. J. A. Morssink, interview, February 23, 2016)

There's a shift in the way of working: it used to be that the teams were facilitating the architects (by providing PSAs for example), now the architects are more and more facilitating the teams (by providing general solutions, methods, boundaries within which they are allowed to roam free, while still working towards a future landscape). (E. Woudt, interview, March 4, 2016, H. Dijkstra, interview, February 25, 2016, J. B. Landman, interview, February 24, 2016, M. de Haan, interview, March 10, 2016 & R. R. Mammen, interview, February 19, 2016) Too many rules won't work either, you can't be too restricting and bureaucratic. (P. J. A. Morssink, interview, February 23, 2016 & R. R. Mammen, interview, February 19, 2016) Projects demand a bigger involvement of architects during the course of a project (sometimes even a dedicated architect), rather than at the beginning, but the question is to what degree you would like to be involved in the daily operation of a development team: you want to know the risks and issues, but you don't want to interfere with the content of a specific sprint. (E. Woudt, interview, March 4, 2016, M. de Haan, interview, March 10, 2016 & P. J. A. Morssink, interview, February 23, 2016) Teams usually approach architects with questions regarding architecture though, but some teams think they can do it just fine themselves, so they start making architectural decisions without notifying the architects. (E. Woudt, interview, March 4, 2016, M. de Haan, interview, March 10, 2016 & H. W. C. Thoonen, interview, March 8, 2016)

The focus of agile software development is on delivery, EA wants to ensure a good architectural landscape to prevent a high IT debt. (J. B. Landman, interview, February 24, 2016 & P. J. A. Morssink, interview, February 23, 2016) While working in an agile way, you don't know the exact end state of

a solution, the furthest you can predict is two-three weeks (the duration of a sprint), so developing an entire architecture for a proposed end state doesn't really fit agile SDM. Of course it is important to have a dot on the horizon/a general idea of what you're working towards, being completely rudderless is undesirable as well. (R. R. Mammen, interview, February 19, 2016) If you don't have such a dot on the horizon and you start deviating from a PSA early on in the project, you are at risk that the final solution is completely different from the approved proposed solution, which should be signaled in an early stage: if you discover at the end of the project that the deviation is too big, you're too late. (J. B. Landman, interview, February 24, 2016, K. P. Boerema, interview, March 15, 2016 & H. W. C. Thoonen, interview, March 8, 2016) It is important to start with a high level overview, general solution, what do you think it will look like, etcetera, but during the course of the project you should start to work much more towards specific designs. (P. J. A. Morssink, interview, February 23, 2016, K. P. Boerema, interview, March 15, 2016 & R. R. Mammen, interview, February 19, 2016)

Documentation has always been a challenge at Rabobank WRR, but with agile SDM it seems even worse than ever. PSAs are being created, but after that documentation (system and infrastructure designs) seems lacking. (R. Zwart, interview, March 1, 2016) There's always a discussion whether a document is project documentation or EA documentation: does the PSA describe the project or its environment? Within Rabobank WRR it is mainly a project document, resulting in every project making a new PSA, which is not very iterative. (R. R. Mammen, interview, February 19, 2016) Decision documentation isn't created during projects, so you can't learn from them when something isn't working out. (P. J. A. Morssink, interview, February 23, 2016)

There is a sort of natural tension between EA and agile IT development: EA revolves around long-term thinking while agile IT development revolves around short-term: it doesn't want to plan too far ahead because the future is unsure: the organization, requirements and insights are always changing. This does not have to be a problem, however: EA can still develop a future state which focusses on the long-term, while they create roadmaps for agile IT development with (small) increments regarding the architectural landscape. This way agile IT development would complement the EA by helping them transform the IT landscape towards the desired state. For this there would need to be guidelines and standards, in order to ensure the teams are working in such a way that the landscape is coherent and durable, but it has to be safeguarded that those guidelines and standards are not too strict: teams need to have a degree of freedom to develop the way they see fit.

There's a bigger pressure on delivering fast than to change the organization in a structured way, which dilutes the architectural landscape. It can be observed that architects are moving from a prescribing role (just provide PSAs and let the teams do the rest), to a facilitating role where they are involved with the teams, check up on them, give advice regarding architectural issues, etcetera. It is important that teams have an idea where they are going and are working under architecture (adhering to the guidelines and standards) to prevent a landscape which isn't agile because of dependencies and interfaces.

Documentation is also mentioned as a problem: it has always been a problem within Rabobank WRR. PSAs are being created, but solution documentation is lacking. The deliverables from agile IT development should be (re) considered and teams should be made aware of their importance.

4.2 Case 2: Rabobank IT Netherlands

Within IT Netherlands (ITN) three interviews were conducted. Three (out of five total) Lead Architects were available to provide information on how ITN was conducting EA and developing software. Each interview lasted between an hour and an hour and a half and was conducted in a semi-structured way [32], just like the interviews at Rabobank WRR. The structure was even more loosely set up at ITN when compared to Rabobank WRR, in order to extract as much relevant information from the Architects as possible, without limiting them to a certain predefined mindset.

4.2.1 Organizational Structure

Strategy and Architecture is a staff department at ITN. Disciplines within this staff department are Security, Risk Management and all the Business Architects. The Business Architects are positioned at the IT side of the organization and while they are called BAs, they are responsible for the IT landscape and its alignment with the Strategy of the business. They are all positioned in the same organizational unit, but are permanently sourced to the domains, in order to ensure that the BAs remain connected and grounded within the domains they are responsible for. ITN prefers pooling the BAs, despite a risk on an 'ivory tower', because they experience that the BAs cooperate more easily in this situation, which improves synergy and contexts. Five Lead Architects manage the BAs and report to René Steenvoorden, MT member of ITN. (E. van Gorp, interview, March 8, 2016)

ITN has functional domains: "Distribution", "Payments, Savings and Investments" (PSI), "Lending", "Organization" and "Infrastructure". The domain Distribution is an ITN wide domain, as a layer, which provides common processes and solutions to the rest of the domains: it addresses and provides client portals, the sales process, client service processes. The product-domains PSI and Lending (Insurance too, but it is not part of ITN: it belongs to Achmea) use these common processes and solutions. The domain Organization focuses on any processes that are not primary banking processes, but which are needed for managing the organization (e.g. Finance, the general ledger, HR). Finally the Infrastructure domain is responsible for all infrastructure used by the other domains. (E. van Gorp, interview, March 8, 2016 & M. Verlinden-van den Berg, interview, March 11, 2016)

4.2.2 Enterprise Architecture

Lead Architects (LAs) have an Enterprise Responsibility: they are not just focusing on their domain, but the five LAs together form the EA team. (E. van Gorp, interview, March 8, 2016 & M. Verlinden-van den Berg, interview, March 11, 2016) LAs use the Tapcott model as a basis to look at the EA aspects: Business, Processes, Information, Application and Technical. Time horizons for each of the EA components/artefacts were defined, as well as levels of detail. EA is high level and has a time horizon of 10-15 years. Domain Architectures are positioned below the EA, have a little more detail, with a time horizon of about 5-8 years, which are updates every 1-2 years. Within domains you have certain segments/programs, who focus on even more detail with a time horizon of 1-2 years. Finally you get the individual PSAs (projects), which are very detailed and have a short time horizon. To be able to achieve the business goals set, roadmaps are created based on the above EA components/artefacts. These roadmaps are used to initiate programs and projects. (M. Verlinden-van den Berg, interview, March 11, 2016)

The LAs have several themes they are focusing on, such as "Data", "Service Architecture", "Cloud" and "Integration". These themes are generic themes for the entire organization, for all domains on which standards need to be devised to guide the organization within those themes. The Lead Architects take on those themes. (E. van Gorp, interview, March 8, 2016 & M. Verlinden-van den Berg, interview, March 11, 2016) For example Rob Douwes said: "the Service Architecture" within ITN is outdated, so I'm going to update it. Formerly ITN had Enterprise Architects that would try to do these kinds of things, but they were in an 'ivory tower', missing connection with the operations of ITN and therefore those initiatives never really took off. Now that the Lead Architects initiate those themes, they know what's going on within the domains, the domains know and respect the Lead Architects and are easier to get on board regarding changes. You can't do this all by yourself though, the themes are way too extensive for that, so you need to make sure you get budget for it

in order to create teams to take action on these themes, start changing and updating the organization, but under your responsibility and direction. (E. van Gorp, interview, March 8, 2016)

The LAs meet every week to keep each other updated on developments within the domains and themes. Once per month Michel Angevare (Manager APS) and Fabian van Altena (Enterprise Architect of Rabogroup; the only one we have; focusing on Data) join the meeting in order to align RN and Rabobank WRR. (E. van Gorp, interview, March 8, 2016)

In order to be able to steer the organization towards a desirable and sustainable architecture, EA creates models. ArchiMate is being used as a tool for this. In order to make EA transparent and explicit, models are created. Ideally you want models that require little maintenance, which is achieved if they are constant (steady) with little changes being made. The steadiest models within ITN are the logical business models: those describe general functions of the organization (e.g. "additional payments for mortgages"), without any mention of processes, departments, etcetera. These functions hardly change and remain for years if not decades. Logical models can be made for the entire organization; applications, functions, information and entities can be modeled this way. When you make them concrete (you link them to instantiations within the organization) and start assigning the organizational/business functions to departments, applications, etcetera they are getting more volatile and maintenance heavy. Logical models for the organizational functions are in place, for the other layers they are not always in place, only when it's sensible to do so. When done in a tool you can link those together, so you can delve deeper into the organization or have more abstraction. Also consistency is easier to ensure in a tool, rather than for example PowerPoint. (M. Verlinden-van den Berg, interview, March 11, 2016)

The LAs organize architecture within their own domain, because one ITN Architecture Board (like the WRR Architecture Board) would become too big to organize. They have Domain Architecture Boards in which disciplines from the domains are represented: BAs, the actual business, Risk, etcetera. The subdomains (programs) each have their own smaller Program Architecture Board in which they discuss their PSAs. At first this was done at the level of Domain Architecture Board within the domains, but even then there were too many PSAs. Giving the programs their own Program Architecture Board also gives them more freedom. PSAs that are in a final state and have to be approved and PSAs that are impacting multiple programs are discussed in the Domain Architecture Board. PSAs that are impacting one of the other 4 domains within ITN (for example there is a project from the PSI domain is impacting the Distribution domain, the BA responsible for the PSA has to get the PSA approved in both domains to ensure alignment, compatibility and knowledge/information sharing between domains. (E. van Gorp, interview, March 8, 2016)

Sometimes this process of creating PSA takes quite some time, projects may find this as an impediment for their projects, and it makes the process slower. They experience positive effects as well though: alignment with other teams and domains is perceived as a good thing. (E. van Gorp, interview, March 8, 2016)

4.2.2.1 Governing Programs (Domains)

Governing the programs within ITN is organized based on equality between the BA, the Program Manager (PM) and the Business Change Manager (BCM); recently a fourth manager was added to this triangle: the Continuity Manager (CM). The BCM is responsible for business priorities and budgeting. The BA is responsible for the contents (of the program/project) and thus for the IT landscape. The PM is responsible for timely delivery and staying within budget (of the program/project). This triangle is implemented at several layers in the organization: the Lead BAs are in a triangle with the Domain Manager and Lead BCM. They are the "lead triangle" and manage seven "program triangles" (comprising BAs, PMs and BCMs), working within the domain PCM. (E. van Gorp, interview, March 8, 2016. M. Verlinden-van den Berg, interview, March 11, 2016 & R. Douwes, interview, March 11, 2016)

There is also a content triangle, which consists of a BA, an Application Engineer (AE) and a Business Analyst (BuAn). BAs are responsible for domains (e.g. Current Accounts); the AE is responsible for the engineering (technical) side of development and the BuAn makes sure that alignment with the business side is done (e.g. requirements). The BA manages multiple of these triangles, so a 1-n

relation. This is of course in a functional hierarchy, as the BA is part of the staff department Strategy and Architecture and the AE and BuAn are in the different programs. (E. van Gorp, interview, March 8, 2016. M. Verlinden-van den Berg, interview, March 11, 2016 & R. Douwes, interview, March 11, 2016)

You could focus on architecting the entire organization and modeling out everything, but that's pointless. The current state is the operational state: if you develop in such a way that the pieces in your architecture are able to communicate with tools, you can have your current landscape visible at any given moment, in real-time. Having the big picture, knowing what has to be done and changed, you can use communicating and coaching skills to guide the organization and teams towards a desirable solution. Things need to be decentralized a lot, responsibilities need to be addressed as low in the hierarchy as possible. You just need to have the right people: the ones that take pride in their work and want to do as good a job as possible. (R. Douwes, interview, March 11, 2016) Set boundaries, but forget about KPIs, processes, procedures, controls, etcetera. Of course you need to check up from time to time, but start trusting people. The time of planning everything in detail is gone, you need to be agile: if it takes 1 year, it takes 1 year, but you can't plan it all in detail. Even on core banking and legacy systems that are business critical you can develop in an agile way: create a shadow system, start developing and testing; if it works and proves to be non-breaking, you can implement it.

Projects are initiated based on the Architecture roadmaps created within the EA discipline. You already know what general direction you're headed to, because you have created the EA, DA, segments, so "developing without knowing where you're going" won't happen in these situations: you know the dot on the horizon you're heading to. Principles (which are flexible and you don't really need to adhere to) and standards (obligatory, unless you get an exemption for it) provide the boundaries within which you can maneuver, tell you what is allowed and what is not: they are there to help you guide your project towards the solution while working under architecture. How you get there in detail is irrelevant. (M. Verlinden-van den Berg, interview, March 11, 2016)

The AEs and the BuAns, not the BAs, create PSAs within ITN. The PSAs are quite high-level design. In the end the BAs are there to architect the IT landscape, but it starts with the business. The PSAs are constructed based on the idea that you are doing projects for the business. Therefore you start with business strategy and high-level business requirements. These lead to business processes to achieve the strategy, which in turn lead to information objects needed in order to be able to execute the processes. This forms an application on a logical level: therefore logical components architecture is important within ITN. For example: the application "current accounts" performs certain actions on certain data. This logical application is finally translated into a technical application (whether you build it yourself or you acquire it is irrelevant for the logical application). (E. van Gorp, interview, March 8, 2016)

The BAs within programs (for example the current accounts domain/program) are responsible for parts of these logical application components. They need to make sure that those logical components are done at one point in the architecture landscape and not have 10 of the same applications running that all do about the same. (E. van Gorp, interview, March 8, 2016)

If projects are deviating from the proposed solution, they need to state how it's going to deviate and how it's going to impact the architecture. The BA then has to approve of the deviation (possible after discussing this in the board). This is quite frustrating for development teams, because they have to deliver in time. They aren't allowed to continue the deviation before architecture approves of it though. (E. van Gorp, interview, March 8, 2016)

BAs within programs also need to show that they are in control of their program. They need to stay in touch with the developers during projects and not only review at the end of a project. (E. van Gorp, interview, March 8, 2016 & R. Douwes, interview, March 11, 2016) At the end of projects, a document called "Project End Architecture" is created, in which the architecture of the solution is reviewed and compared to the proposed solution and architectural landscape guidelines. This is done in order to determine whether the project delivered under architecture or not. It is not an extensive document: just a few sheets to elaborate on the final solution. (E. van Gorp, interview, March 8, 2016)

From the above description of how EA is implemented within ITN it can be concluded that EA is used as a strategic discipline, which aims to help the business to execute its strategy, support its vision and mission and reach its goals.

4.2.3 Software Development

The domain PSI is not yet developing in an agile way, but has had discussions about it already. The general consensus is that PSAs should not be limiting agile software development teams too much. The PSA does not require fully detailed designs of solutions, they only set boundaries within which the projects should develop. The PSAs are quite high-level abstractions of the solutions, such a high-level abstraction that not even all interfaces are required to be put into the PSA. The PSA process itself is not part of the project, it is a preliminary phase (at least at ITN), setting the boundaries of the solution and project are not dependent on the way you are developing your software. (E. van Gorp, interview, March 8, 2016)

The domain Lending is working agile to a large degree. Most teams have adopted agile SDM. (M. Verlinden-van den Berg, interview, March 11, 2016)

Distribution has organized itself as DevOps and it mainly working agile within those teams. You have all-round teams working on various projects. This requires a lot of communication, but also understanding of each other's work, responsibilities and activities. As architect you don't have to be "standby" (whenever something breaks down, you need to be online within 15 minutes, you can't drink, etcetera), but you do need to know what kind of impact it has on developers that do. You don't need to think about changing the way or working extensively; you just need to start doing it and make it work. Just like switching from plan-driven to agile. In the end everything will work out as long as you act sensibly, communicate and together. (R. Douwes, interview, March 11, 2016)

It is not always possible to work in an agile way. There are standards and principles for projects that state "reuse, before buy, before build". This means that ideally you reuse your applications, if you can't and you need to buy an application, you're dependent on the vendor. If the vendor isn't working in an agile way you can hardly force them to deliver in such a way. The same goes for big infrastructure projects: you can't buy and deploy hardware iteratively. For those kinds of projects you need to come up with an extensive analysis and plan on what you need and how you're doing to implement it. In such cases you often see a combination of plan-driven and agile: the big pieces are done plan-driven, while interfaces, configurations, etcetera are done in an agile way. (E. van Gorp, interview, March 8, 2016)

Developing on mainframes happens in plan-driven projects. They are so critical for the organization and its processes that you can't afford to take any risks by developing agile on them. The core infrastructure does not need much agility. Changes, to this infrastructure rarely happen; and when these changes happen you'll know a long time beforehand that it'll have to change. (E. van Gorp, interview, March 8, 2016)

ITN is working agile in various degrees. Some domains are still working according to plan-driven development, while other domains are working (fully) agile. There is even a domain that has adopted DevOps along with agile software development. The domains do not see the current EA discipline within ITN as a limiting factor for agile software development.

4.2.1 Friction between Enterprise Architecture and (Agile) Software Development

Some teams are agile to such a degree that they want to make the own decisions regarding everything (completely self-regulating), for example regarding tooling, languages, databases, hardware, etcetera. ITN has standards in place to minimize diversity, which could mean that the language or tool the team wishes to use is not part of the set of standards. Then you have discussions between the architects and the teams about making an exception or not. (M. Verlinden-van den Berg, interview, March 11, 2016)

Teams sometimes embrace all changes and want to react/develop in an agile way, without keeping architecture in mind or in the loop. There are foreseeable changes and unforeseeable changes. The

foreseeable changes are usually taking into account in standards and architecture and if not, they should notify architecture. Some teams don't want to distinguish between those two. (M. Verlinden-van den Berg, interview, March 11, 2016)

Developers experience a lot of pressure from projects to deliver as much as possible, as fast as possible by their Project Leaders. EA wants to ensure applications that are architecturally sound and of high quality, because this will improve the quality of your architecture landscape. So the BAs have to check up on the architecture teams to make sure that no corners are being skipped, because there are some deadlines to be met.

4.2.1.1 Tips for Enterprise Architecture

It is very important that the BAs are seen as equal partners regarding the organizational strategy (the triangles). (E. van Gorp, interview, March 8, 2016. M. Verlinden-van den Berg, interview, March 11, 2016 & R. Douwes, interview, March 11, 2016) If you're talking about real EAs, you need to be such a partner of even the MT or even the EB in order to give the organization towards a good strategic organizational and architectural landscape. (R. Douwes, interview, March 11, 2016)

The process used to start projects (the PSAs, which start with business and work towards technology) is addressing EA from beginning to end and is therefore really strong. The BAs are positioned really well within the organization: pooled together, just under the MT, but permanently sourced to the domains to ensure a connection with them and prevent 'ivory tower' thinking. It also prevents the inability to stay objective: if you're hierarchically part of a domain, you risk being influenced by them and follow the lead of the business, rather than a sustainable architectural landscape. In that case you'd be a solution architect: you're creating what the business wants, rather than creating a constructive discussion with the business. (E. van Gorp, interview, March 8, 2016) BAs should therefore not only have analytical and modeling skills, but also communicative and consulting skills. (E. van Gorp, interview, March 8, 2016 & M. Verlinden-van den Berg, interview, March 11, 2016) RN aims to have BAs who are T-shaped: broad knowledge about (almost) everything, and deep knowledge about one discipline. (E. van Gorp, interview, March 8, 2016)

Positioning of BAs alone is not enough: your governance and management approach of the organization has to support the BAs too. The organization (and business) has to acknowledge the BAs as equal partners, acknowledge the importance of EA and organize itself into such a position that enables this setup. There is a correlation between the BAs and organizational management (MT): proven successes of BAs aren't enough; the MT has to have a positive attitude about BAs as well (and vice versa). (E. van Gorp, interview, March 8, 2016)

If you create models sensibly by looking at what models are hardly changing and which ones are needed for projects, you can model your organization to make it transparent. High-level models of your application landscape etcetera are very useful too. For this you need to acknowledge different levels in the organization and move towards involving the business, rather than focus on IT. (M. Verlinden-van den Berg, interview, March 11, 2016)

You need to move away from blueprint thinking and trying to record everything in models (current states, future states, etcetera): you don't live in a utopia where you can model everything perfectly before changing the organization towards it. Furthermore, reality is ever changing, projects are always changing the current state, and it is simply impossible to maintain all your models. You need to create boundaries and have ideas on where you want to go. (E. van Gorp, interview, March 8, 2016)

As EAs you should also make sure that people are communicating with each other: the BAs, the developers, the business; the entire lineage needs to communicate. If communication is done right, you are a lot more proactive and coaching, rather than prescribing and policing. (R. Douwes, interview, March 11, 2016)

4.3 Case 3: ABN Amro Bank

At ABN Amro Bank one interview was conducted, with a Principal Architect, who has been active in the banking domain since 1994, and working at Fortis since 2003, which became ABN Amro Bank in 2008. Being a Principal Architect means you're responsible for a specific architecture domain. He is Principal Architect International and is responsible for the International Enterprise Architecture. (J. Jedema, interview, March 23, 2016)

ABN Amro Bank has its international activities carried out by independent local banks, located in 28 countries. ABN Amro Bank Headquarters provides those local banks with IT, processes and organization, through shared services. International Enterprise Architecture only deals directly with about 10 countries by focusing on the global business lines such as Private banking and Corporate Banking. (J. Jedema, interview, March 23, 2016)

EA has recently undergone organizational changes within ABN Amro Bank. It is a young discipline and they are still searching for the optimal way to organize it within the organization. This is why the structure of this case is a bit different: it describes the thoughts ABN Amro Bank is going through, the ideas they are having and how they plan to organize it. Part of this description is the history of software development and how the realization came to be that things had to change. (J. Jedema, interview, March 23, 2016)

4.3.1 Organizational Structure

The Enterprise Architecture group is situated at the headquarters, all architects are positioned in the same department. In the past they had different teams of architects, within domains (IT Netherlands, IT International, IT Infrastructure and Security), all with their own architecture disciplines. While there was a community and alignment took place, the MT felt that the architecture departments had capable people, but were positioned too low in the hierarchical chain. This resulted in the fact that architects missed authority within the organization: the normal management could get past architecture quite easily by escalating issues. (J. Jedema, interview, March 23, 2016)

When the strategy towards 2020 was presented, it was also announced that architecture should become more important within ABN Amro Bank by pooling the Lead Architects together and place them directly under the board of directors. This means that the head of Enterprise Architecture is a member of the board of directors: Chief Architecture Data Management. He manages the Lead Architects (LAs) and Data Managers in ABN Amro Bank. The architects have different attention areas: for example International. The architects are organized from the business towards operations and IT through chains. This means that they are focusing on the chain "credit", or "payments", etcetera, which go through the organization: there's a bit of business, operations, IT and some infrastructure. For each chain there is a LA, which manages the other architects within that chain. Because the LAs are positioned together in one team, alignment between domains automatically becomes easier and stronger. (J. Jedema, interview, March 23, 2016)

The other architects remained at the same position in the organization and still report to their line manager. While there are no hierarchical reporting lines between the Domain Architects (DAs) and the LAs, the LAs still have mandate and authority within the organization, as they report to the board of directors. (J. Jedema, interview, March 23, 2016)

4.3.2 Enterprise Architecture

4.3.2.1 Reasons Leading to the Current Enterprise Architecture Organization

In the past projects developed with a short-term focus: delivery had to be fast, short cuts were taken and not much thought went into maintainability. Systems and applications (and the application landscape) grew organically, creating a lot of dependencies between them. In these projects, architecture only got involved after the solution was already decided upon. Architects always gave advice regarding the best solutions, but those were ignored. (J. Jedema, interview, March 23, 2016)

Every time you take such shortcuts and easy ways out, you create a little bit of technical debt: you need to spend money to rectify the shortcut later on. Over the years this technical debt accumulates quickly and gets more expensive to rectify every time, as you cannot do it piece by piece, as the spaghetti of systems is all connected and dependent on each other. (J. Jedema, interview, March 23, 2016)

Within the change program towards 2020, the board of directors decided to address this technical debt and reorganize the IT landscape. At the same time they decided to change the architecture discipline, to prevent this same situation from happening again in the future, which is why the Lead Architects were placed directly under the board of directors. (J. Jedema, interview, March 23, 2016)

The executive board gave the EA department a large budget to reorganize the architecture landscape, which placed EA almost immediately at the same level as the business: EA (or IT) didn't need the business anymore for budgets. While EA is reorganizing the landscape, the business is also strategizing and developing ideas. They need the IT to realize those, so EA is invited to the discussions more and more, to ensure that the architectural landscape will have space to accommodate for the business solutions. (J. Jedema, interview, March 23, 2016)

4.3.2.2 Goals of Enterprise Architecture

The first aim is to change from project driven architecture (projects are started, architecture gets involved later on as a sort of project/solution architecture) towards architecture driven projects (projects initiated because of identified gaps in architecture). In the latter case you would communicate with the business about the goals and strategies and how to get there. Then you devise a future architecture in which the business goals can or will be realized and initiate projects based on the delta in the architecture. (J. Jedema, interview, March 23, 2016)

Those projects should develop their software products within predefined boundaries and adhere to architectural rules. Whether they are developing software in a plan-driven or agile is then irrelevant. While you cannot and should not define all boundaries and rules very strictly (as this limits the organizational agility), you do know general boundaries and general solutions, which you want to head towards, for example regulators enforcing access of external parties to your data within the next few coming years. You can already put boundaries, rules and ideas forward on how you want to organize that. How teams eventually develop the services that provide such access doesn't have to be designed in detail beforehand. (J. Jedema, interview, March 23, 2016)

You can already work out general solutions towards such themes, without actually designing specific solutions. If you move towards agile ideas with EA, you connect a lot more with the development teams. Perhaps you should define 'epics' on EA level, such as: "we want cloud solutions". (J. Jedema, interview, March 23, 2016)

EA should focus on decoupling applications within the landscape. If you connect them through services (or APIs), you can replace one of them, or even both at the same time, if so desired. If you make point to point connections this is a lot harder. Agile software development might make such decoupling easier for EA. (J. Jedema, interview, March 23, 2016)

EA has to make sure that the business side (at a higher management level) is aware of the added value of EA. If the business is convinced of the added value, EA will be involved in strategic planning sessions regarding problems. Problems shouldn't be chopped up in pieces and each individual piece addressed and solved, but the bigger picture should be kept in mind, EA can help with that. Problem solving by chopping them up in bits and pieces is short-term thinking, EA can help with a long-term vision. (J. Jedema, interview, March 23, 2016)

4.3.2.3 Current Enterprise Architecture Situation

The Chief Architecture Data Management wants to be kept in the loop regarding project boundaries, rules etcetera. He doesn't want to know what projects are about to start, because the boundaries and rules are set beforehand should be enough to ensure that a project is working under architecture. This resulted in the fact that projects don't have to go through an Architecture

Board anymore to get approved. It is up to the Lead Architects to approve or decline them, so now you see that the LA is being involved in the preliminary phases of projects early on, so that he knows what is happening, what the ideas are and therefore can steer towards a desired solution, which in the end results in a fast approval. (J. Jedema, interview, March 23, 2016)

The Lead Architect is also accountable that the projects deliver products under architecture. So a process is being created that ensures checkups on teams so that architects will be aware when projects start to deviate from the Proposed Solution or start to cross the boundaries set forward for projects. (J. Jedema, interview, March 23, 2016)

You already notice a higher level focus: from strategic discussions about what direction to head to, project briefs are created. These project briefs eventually lead to IT architecture. The IT Architects stay involved within those projects, as they are the ones responsible for delivery. IT Architects start designing software deliverables into more detail as time goes on. (J. Jedema, interview, March 23, 2016)

The board of directors recognizes the importance of architecture, which is why they placed the Principal Architects as a staff department directly under their control. EA is seen as a strategic discipline, which can help the business in executing the strategies and achieving their goals.

4.3.3 Software Development

ABN Amro Bank is hardly working with agile teams for software development. The way architecture was used by projects makes the organization wary of adopting the agile way of software development. There are some pilots with projects, but they are very scarce within the organization. They are searching for a way to start agile software development, as agility is one of the IT drivers towards 2020. The teams that are developing in an agile way already are experiencing problems with alignment between teams: the portal and the backend for example. Also infrastructure development for the agile software projects is hard to do in an agile way. (J. Jedema, interview, March 23, 2016)

The risk with agile software development is that you just start developing and choose the easiest solution every time. This solution may not be the best though and may not be future proof. You have to make sure you don't keep refactoring and adding functionalities without planning, this would become unmanageable and no one would know the technical design of the application anymore. If you change one small piece of code, it impacts a lot of functionalities that are dependent on that piece of code or on each other. (J. Jedema, interview, March 23, 2016)

ABN Amro is mainly developing IT through plan-driven software development. Some pilots are taking place. They are wary of agile development as it might lead to taking shortcuts when focusing on fast and cheap delivery.

4.3.4 Friction between Enterprise Architecture and (Agile) Software Development

It is going to be challenging to have an idea as to where you're heading, while keeping it abstract and general enough for agile software development teams to work comfortably and agile towards it. (J. Hendriks, interview, March 23, 2016)

The main friction will not be between EA and agile software development, but between the Solution Architecture and agile software development: EA should have boundaries, rules and general solutions in place before the project starts. Solution Architecture is done in concurrence and during agile projects, so that might be perceived as a limiting or slowing factor. (J. Jedema, interview, March 23, 2016)

When you're developing in an agile way, but not have an EA in place yet, there's friction because the projects have to wait for EA to come up with its boundaries, rules and general solutions. (J. Jedema, interview, March 23, 2016)

4.4 Case 4: SNS Bank

At SNS Bank one interview was conducted, with the former Enterprise Architect Manager, who has decided to step down and take place in projects again, as an architect. He is however knowledgeable about SNS Bank as he has been with SNS Bank since 2002.

4.4.1 Organizational Structure

EA is very dependent on the management in place: if you have a management that is very architecture minded, you get a lot more mandate and a lot more responsibilities and vice versa. EA is in basis an IT effort, which also incorporates business. Management is starting to realize banks are a lot more like IT companies at the moment, rather than traditional banks. (J. Hendriks, interview, March 23, 2016)

Within SNS Bank the architects are positioned under the MT of the Information and Change Business Line, but they are responsible for the entire EA of the SNS Bank. The architects are pooled together in one department who are taking care of the entire EA discipline. In order to have an effective EA within the organization this is a prerequisite. If you have the architects positioned under different hierarchical managers you can't make one centralized architecture, every management line is going to have its own architecture. You can still have architecture delegations within the domains, because they are very knowledgeable about the domains and products. Those architects within the domains are being delegated work: they are very knowledgeable and informed and therefore able to make a Project Start Architecture for example. (J. Hendriks, interview, March 23, 2016)

4.4.2 Enterprise Architecture

The EA starts with the goals and the strategy of the business for the coming years. Based on these plans, the architects create Reference Architectures (RAs) and Roadmaps. Roadmaps are an impact analysis in order to be able to determine what has to be done to achieve the business goals. Those Roadmaps are made for all levels of the organization and allow the business to restructure and reformulate their project portfolio. Those RAs are created and updated in a pragmatic way: if you know there won't be any changes for the coming years, you don't have to update the RA. (J. Hendriks, interview, March 23, 2016)

Besides those RAs and Roadmaps, the architects feel the need to have their own artefacts: you need to have a baseline and models about the organization so that you can support the change processes and initiatives from the organization in a timely matter. You don't want to have to create and model everything after they have already asked for it. Keeping those artefacts up to date is the responsibility of the architects themselves. Whenever you start working with change processes and projects, you need to tell the other architects what you're working on, what the plans are, so you need to align and communicate a lot. (J. Hendriks, interview, March 23, 2016)

Some programs are still asking for program blueprints. The architects then create those and help give those programs focus on how to create a desired architecture within the program. They also help the program manager with defining projects and realizing them. Architects also help put together PSA for those teams. (J. Hendriks, interview, March 23, 2016)

The above sections describe how the SNS Bank EA works in a change cycle for the organization and programs, but also about their opinion on maintaining their own products. Lately they changed from the DYA EA method to TOGAF, because "just in time architecture" proved to be too late for agile development teams. (J. Hendriks, interview, March 23, 2016)

4.4.3 Software Development

Software development within SNS Bank seems to work mainly according to changes needed in the RAs. The business uses the Roadmaps created based on those RAs to start projects, so at SNS Bank the architectural context is already modelled and in place to provide to the projects. Those projects start developing the changes (or new applications) needed to fill in the gap (between the Baseline and the RA) and at the end of the project the architecture landscape is updated (in accordance

with and mainly by the architects) and fed to the Baseline. It is important to make sure that this last step (of the TOGAF method) is carried out, because architects have the tendency to start with the next project as soon as possible. The development process prescribes certain phases and requirements for projects, such as “there needs to be an architect involved” and “there needs to be a change board”. This is done to be able to keep checking if the project or program isn’t deviating from the RA and staying within the boundaries. (J. Hendriks, interview, March 23, 2016)

SNS Bank is mainly working in an agile way, with some exceptions: transaction systems aren’t developed in an agile way on. Those transaction systems are very robust, well developed and don’t need many changes and developments anymore. This system is critical to the organization, it already does what it needs to do and all new products devised fit into the system already. (J. Hendriks, interview, March 23, 2016)

Architecture within SNS Bank focusses on the big picture. It is important that project teams are adhering to the general rules, guidelines and requirements. The precise way in which they are doing this doesn’t matter. (J. Hendriks, interview, March 23, 2016)

In the past EA made sure services were constructed, so agile software development has room to develop using the legacy backend systems. These services also enable two-speed IT, inspired by Gartner: using the systems of record (legacy systems and mainframes), systems of differentiation (services and middleware) and systems of innovation (the agile initiatives, apps, etcetera). Using two-speed you can develop plan-driven on the backend, while developing fast and agile on the frontend. (J. Hendriks, interview, March 23, 2016)

4.4.4 Friction between Enterprise Architecture and (Agile) Software Development

SNS Bank also experiences some friction between EA and agile software development. This isn’t caused by one of the two processes but mainly by the people in those processes. It depends on the level of cooperation people want to have. In projects you have to have mutual respect for each other: you have to realize that you can’t do what they are doing and vice versa. People should be willing to see that if you cooperate, that you can accomplish a lot more. Of course there are some cases where the rules and process are the basis of friction: a Project Leader is responsible for the PSA, for involving an architect, even if it’s business as usual. This can be seen as a useless rule in that case and frustrates the Project Leader. (J. Hendriks, interview, March 23, 2016)

5 Analysis

The aim of this chapter is to analyze the Case Study descriptions and the Theoretical Background in order to be able to answer the research questions in Chapter 6 Limitations, Conclusions and Future Work.

5.1 Implementation of Enterprise Architecture and IT Development

This section will start by analyzing the results from the four case studies (where at the end of each case study description information towards answering the first two sub research questions was given. The two sub research questions devised were the following:

SRQ1: How is the enterprise architecture discipline implemented in agile working organizations?

SRQ2: How is IT responding to changes in agile working organizations?

The information from these sections will be aggregated per case study in order to provide a description of how the two disciplines are implemented within the organization (which are going to be the input for the conclusions of sub research questions one and two).

5.1.1 Rabobank Wholesale, Rural & Retail

Enterprise Architecture

EA within Rabobank WRR is organizationally placed in such a way that Business Architecture is on the level of both IT and Operations (ITOPS). This was done by the MT of ITOPS, who wanted to have a link with the business. The other architecture disciplines are placed lower in the hierarchical lines. Most of the EA components are present at first glance:

- Architectural disciplines are present: Business Architecture, IT/Application Architecture, Infrastructure Architecture and Domain Architecture.
- Governing components are also described: Roles and responsibilities of architects, Artefacts (Principles and Technology Bricks/standards), Processes and Architecture Meetings.
- Deliverables to be produced are also described: Business Architecture, Target Operating Models, Information Architecture, Application Architecture, Infrastructure Architecture, Project Start Architecture, Project End Architecture and Domain Reference Architecture.

The Business Architects described an overall process of how all the architecture disciplines relate to each other, what the deliverables for Business Architecture are, how they should link to the Information Architecture, how that should define the Application Architecture, which finally leads to the Infrastructure Architecture (after which programmes and projects could be initiated).

The MT of ITOPS is still an IT department within Rabobank WRR. Its clients are the regions of the world, who are at the same hierarchical level as ITOPS. This results in the fact that the EA discipline within Rabobank WRR is still completely an IT department, even though they wish to create a link with the business. The Business Architects simply do not have a mandate within Rabobank WRR, because of this situation, the business does not see EA as an equal partner: "the one who pays gets what he wants" and it's the business with the budget to initiate projects.

This resulted in Business Architects getting goals that are not EA minded, but rather IT minded: e.g. "create a business-led architecture", "give insight in current global WRR landscape". These are still IT minded, rather than involving the business into the EA discipline, so that the business strategy and goals lead to a Target Business Architecture, which is then used to create a Target Information Architecture, and so on. This would lead to a gap analysis between the Current Architecture Landscapes and the Target Architectural Landscapes, which would result in a roadmap and projects initiated based on that roadmap.

Because the lack of mandate of Rabobank WRR MT and the lack of recognition that EA is important by the business, the EA discipline within Rabobank WRR is inefficient. This is worsened by the fact

that while the Deliverables are defined, a large amount of them are still missing, resulting in a fragmented landscape. Because the domains within Rabobank WRR are responsible for their own Domain Architectures, there is little alignment between the domains, as the architecture is not modeled from the top down, but from the bottom up.

Artefacts guide the projects regarding the choices they make in the project: there are guidelines that steer them in the right direction and there are standards that enforce hard boundaries to which they need to adhere.

Finally, the various processes and deliverables defined are not all implemented and enforced within the organization: the most important process that is being executed is the Project Start Architecture, with a Project Start Architecture (document describing the project to be started) as output. This has to be approved by the IT/Application Architects, but can be escalated through the hierarchical line managers if really needed. When the PSA is approved, there is no controlling process to check if the solution that is (being) developed adheres to the one described in the PSA.

(Agile) IT Development

Rabobank WRR is usually working in an agile way when developing its IT. In specific cases plan-driven development is chosen: if there are very big programs, or if it involves the back-end (transaction) systems, which require careful development in order to be sure not to break them, as they are critical to the organization.

The IT projects are initiated through business requests, after which a PSA has to be created and approved. When this PSA is approved they can start developing the IT. The agile teams also miss the business involvement in their projects.

5.1.2 Rabobank IT Netherlands

Enterprise Architecture

Because of the way EA is implemented within ITN, it can be concluded that EA is used as a strategic discipline, which aims to help the business to execute its strategy, support its vision and mission and reach its goals.

The importance of a strong architecture discipline is recognized, which is why the architects are placed together in one staff department directly under the MT. The architects are sourced permanently to the domains in order to stay grounded in the domains. The architects have a real voice within the organization and are placed in triangle constructions, where they are of equal importance to their IT and Business equivalents.

The Business Architects create models of current and target states, preferably in such a way that requires minimum effort because the modeled landscapes are of such abstraction that they are not that volatile. The lower you get in the organization (Enterprise Architecture, Domain Architecture, Program Architecture, Project Architecture), the more detail these architectures and models have. The architectures and models are created sensibly though: only if needed are they created. The difference between current and target states form the basis for roadmaps. These roadmaps are then used to determine which programs/projects should be initiated, in order to evolve the current state into the target state and PSAs are created by Domain Architects and Business Analysts.

Principles and standards set boundaries and rules for the programs and projects, so they know what is desired (principles) and what is (dis)allowed/obligatory (standards). As long as they adhere to these principles and standards they can develop in any way they want as long as they follow the PSA. Architects stay in touch with projects in order to check if the project is still developing under architecture, instead of a check at the end; a Project End Architecture is created to have a final check of the architectural status.

(Agile) IT Development

ITN is developing IT in various degrees of agile. Some domains are still working according to plan-driven development, while other domains are working (fully) agile. There is even a domain that has

adopted DevOps along with agile software development. The domains do not see the current EA discipline within ITN as a limiting factor for agile software development.

The process of creating the PSA is a preliminary phase of the project, so they don't have to wait for it. The contents of PSAs are such high-level that they don't specify detailed solutions, but leave enough room for the development team to maneuver around and develop in an agile way. The principles and standards give you boundaries within which you're allowed to develop in an agile way, while still developing under architecture.

5.1.3 ABN Amro Bank

Enterprise Architecture

At ABN Amro Bank, the Lead Architects are placed together in one department, directly under the board of directors. The other architects are placed lower in the hierarchy under the domain managers, resulting in a functional reporting line to the Principal Architect. The board of directors recognizes the importance of architecture however, which is why they placed the Principal Architects as a staff department directly under their control.

ABN Amro Bank is restructuring the way the EA discipline is implemented and used. EA identifies gaps in the architecture landscape by communicating with the business about the strategies and goals and how to get there. They then create a target state in which the goals can be achieved. Based on the gaps between the current and target state, projects will be initiated to close those gaps.

For the projects, boundaries will be defined, along with architectural rules. Because of the project plans based on the gaps, along with the boundaries and architectural rules, it won't matter if the projects will be executed in an agile or plan-driven way.

The Chief Architecture Data Management cares about the boundaries and rules, but doesn't want to know every project that is about to start. If the rules and boundaries are organized correctly, the projects are started based on the gaps in architecture and the development teams adhere to the boundaries and rules, it should be all right as the project is executed under architecture.

The Lead Architects are accountable that the development teams working under architecture, so a process is being devised to ensure checkups on those development teams.

(Agile) IT Development

ABN Amro Bank is mainly developing IT through plan-driven software development. Some pilots are taking place. They are wary of agile development as it might lead to taking shortcuts when focusing on fast and cheap delivery. They don't expect friction between EA and agile software development though, as EA should have boundaries, rules and general solutions in place before the project starts, so it's not interfering with the timeline of projects.

5.1.4 SNS Bank

Enterprise Architecture

At SNS Bank placed all the architects in one department as a staff function under the MT of the Information and Change Business Line, in order to be able to create a centralized EA discipline. Within the domains there are still some architects, who get delegated work from the central EAs.

EA starts with the business strategy and goals for the coming years. Based on the strategy and goals, Reference Architectures are created/updated and compared to current states. This leads to Roadmaps, which are created for all levels of the organization and form the basis for projects. Business initiates these projects based on the created Roadmaps. At the end of a project, the architecture landscape (current state) is updated.

(Agile) IT Development

SNS Bank is working in an agile way, except for their transaction systems. Projects need to adhere to architectural rules, guidelines and requirements to ensure they are working under architecture.

Because the gaps in architecture are already identified, they have a dot on the horizon as a goal. How they get there doesn't matter, as long as they work under architecture.

5.2 Relationship between Enterprise Architecture and IT Development

The third sub research question aims to shed light on areas of friction or areas that work well together, by comparing the results from sub research question one and two:

SRQ3: What are the differences between how IT aims to respond to (disruptive) innovations and how the enterprise architecture discipline is implemented?

Following the description regarding implementation of EA and (agile) IT development it was intended to describe what the differences and similarities were regarding Enterprise Architecture and (agile) IT development. As it turned out, however, the two disciplines are not competing with each other, instead they complement each other, if implemented correctly.

This resulted in the outcome of the analysis to be different from what is expected when reading SRQ3: it does describe if there is friction between the two disciplines, but the frictions is caused by the way EA is implemented, not because of similarities or differences.

Therefore, the description will be how they are impacting and complementing each other. This will be discussed per case, in order to be able to compare the four cases with each other and determine if there is any generalization possible.

5.2.1 Rabobank Wholesale, Rural & Retail

EA and IT development are impacting each other to a large degree within Rabobank WRR. They experience a lot of friction between them, because EA is seen as a controlling and policing discipline, rather than an enabling one. EA is mainly reactive when it comes to starting projects, rather than proactive: the projects are not initiated based on architecture, but because the business asks for a solution/product.

EA is not seen as a strategic tool with which you can structurally evolve your organization based on business needs. IT treats EA as a tool to get input from the business, rather than working together with it. There is no traceability from the top down towards projects, or vice versa.

While guidelines and standards are in place, this does not guide projects enough to develop without first having extensive interaction with EA.

5.2.2 Rabobank IT Netherlands

The EA and IT development are impacting each other in a minimal way. EA is used to help the organization to evolve in a structured way to fulfill its strategy and achieve its goals, by creating models and roadmaps to initiate programs and projects. The IT development starts because of the EA efforts, not by direct assignments of the business and are guided by predefined architectural principles and standards.

5.2.3 ABN Amro Bank

ABN Amro is in the process of restructuring its EA discipline and making it of strategic importance. It expects little to no friction between EA and agile software development, as they are to separate sequential activities. EA will help the organization to evolve in a structured way to fulfill its strategy and achieve its goals, by identifying architectural gaps current and future states. Projects will be initiated based on these gaps identified by EA and guided by predefined architectural boundaries and rules.

5.2.4 SNS Bank

SNS Bank uses EA as a strategic important discipline. It helps the organization evolve in a structured way, so it can execute its business strategy and achieve its business goals. Projects are initiated by the business after gaps are identified between current (Baseline) and target (Reference

Architecture) states. Projects are guided by predefined architectural rules, guidelines and requirements.

5.3 Case Study and Systematic Literature Review

The main research question (MRQ) aims to determine how EA should be implemented in agile working organizations, in order to improve the responsiveness of IT:

How can the enterprise architecture discipline (including its governance, models and artifacts) be (re)designed, with the aim to improve the responsiveness of IT in agile working organizations operating in the financial domain?

With the insight and analysis results of SRQ3 in Section 5.2, the MRQ is can partially already answered. However, the comparison of the SLR and information extracted from the case study might still add new insights as well.

5.3.1 Recap of the Scientific Literature Review

The amount of literature found was scarce as not many scientific research has been done regarding the interaction between EA and (agile) IT development. However, lately it sparked the interest of the scientific field and two master theses were devoted to making EA more agile. The proposals of these two researches will be discussed in this chapter, by relating the findings in the four cases of the case study to them.

The first thesis to address EA and Agile SDM was Hensema [24], who aims to provide key points, inspired by agile practices that could be beneficial to EA. These key points should be addressed in order to make EA more agile. He starts with a survey to determine which agile practices would be beneficial, after which three remain.

- A process such that during development new requirements can be incorporated. Development could be done incrementally and iteratively. This requires formulating high-level boundaries and principles which can be elaborated upon level-by-level.
- A method of reflecting on the work that has been done. The method should incorporate EA challenges perceived in the organization.
- A system that enforces that only artifacts are produced which are essential. Architects must realize that the process in creating the artifact is what delivers value. Creating a definition of done can help in determining what is useful and create a sense of time pressure for architects.

These three agile practices are then used in a round table discussion, which eventually leads to four key points, which should be incorporated into EA in order to make it more agile.

1. EA artifacts are not what delivers value. The process of creating the artifact is what delivers value to stakeholders. EA artifacts are a tool for documenting the process.
2. Communication of artifacts is not solved by a development approach. It is a skill of the architect. It is influenced by the amount of creativity given to the architect and formalization of artifacts.
3. Incremental and iterative development are possible but require rough borders: high-level principles and boundaries.
4. Create a definition of done for EA artifacts. This can define what is considered useful by stakeholders, it creates a sense of time pressure for architects and enables EA development progress tracking.

The second thesis, by Lumor [33], proposed an Agile Enterprise Architecture Management Method (Agile EAMM). This Agile EAMM is based on eight "essential elements" (EEs), which are derived from "factors that influence the agility of the enterprise, namely; agility drivers, agility providers and agility capabilities". The eight EEs are:

1. "An Agile EAM Method should provide a mechanism to develop and continuously refine EA vision, goals, purpose, principles, and scope to accommodate changes in enterprise competitive bases, and turbulence in the external and internal environment."
2. "An Agile EAM Method should maintain appropriate balance between long-term and short-term orientation and satisfy EA stakeholders through early, continuous and incremental delivery of effective and efficient EA products towards achieving a (shifting) long term goal/strategies."
3. "An Agile EAM Method should provide an appropriate decision making and governance mechanisms (e.g. collaborative decision making and consensus building) that result in participation and commitment of all stakeholders, and decisions that benefits the enterprise as a whole rather than individual domains."
4. "An Agile EAM Method should promote the development of skilled teams and adaption of methods and tools, and provide mechanisms for defining EA requirements and criteria for project goal fulfillment."
5. "An Agile EAM Method should support appropriate representation of architectural artifacts and communicate architectural documentations across the enterprise in a common language."
6. "An Agile EAM Method should provide mechanism to develop and maintain an integrated architecture repository for knowledge management and enterprise awareness initiatives."
7. "An Agile EAM Method should act as a dynamic capability (e.g. as sensing, learning, integration, and coordination capabilities), and support enhancement of agility capabilities and development of agility providers of the enterprise."
8. "An Agile EAM Method should promote the anchoring/integration of its mechanisms and tools into the culture of the organization."

5.3.2 Comparison of Theoretical Background and Case Studies

Looking at the four key points put forward by Hensema [24] and the eight Essential Elements (EEs) that are part of the Agile EAMM proposed by Lumor [33], it is possible to map the four key points of Hensema onto the eight EEs defined by Lumor, as depicted in Table 1 Key Points of Hensema [24] mapped to EEs of Lumor [33]. Therefore, only the eight EEs of Hensema [24] will be discussed as part of this analysis.

Key Point proposed by Hensema [24]	Maps to EE(s) of Lumor [33]
EA artifacts are not what delivers value. The process of creating the artifact is what delivers value to stakeholders. EA artifacts are a tool for documenting the process.	2 and 3
Communication of artifacts is not solved by a development approach. It is a skill of the architect.	4, 5 and 6,
Incremental and iterative development are possible but require rough borders: high-level principles and boundaries.	1 and 2
Create a definition of done for EA artifacts.	4

Table 1 Key Points of Hensema [24] mapped to EEs of Lumor [33]

In order to create an overview of the degree to which the eight EEs proposed by Lumor [33] are used within the four case companies, Table 2 EA of Case Study companies mapped to Essential Elements of Lumor [33] was created. For each of the eight EEs an indication is given whether or not the case company has incorporated the EE (+); has partially incorporated the EE (+/-); has not incorporated the EE at all (-); or it is unknown (Unknown).

Essential Element of Lumor [33]	Rabobank WRR	Rabobank ITN	ABN Amro Bank	SNS Bank
An Agile EAM Method should provide a mechanism to develop and continuously refine EA vision, goals, purpose, principles, and scope to accommodate changes in enterprise competitive bases, and turbulence in the external and internal environment.	+/-	+	+/-	+/-
An Agile EAM Method should maintain appropriate balance between long-term and short-term orientation and satisfy EA stakeholders through early, continuous and incremental delivery of effective and efficient EA products towards achieving a (shifting) long term goal/strategies.	-	+/-	+/-	+/-
An Agile EAM Method should provide an appropriate decision making and governance mechanisms (e.g. collaborative decision making and consensus building) that result in participation and commitment of all stakeholders, and decisions that benefits the enterprise as a whole rather than individual domains.	-	+	+	+/-
An Agile EAM Method should promote the development of skilled teams and adaption of methods and tools, and provide mechanisms for defining EA requirements and criteria for project goal fulfillment.	+/-	+/-	+/-	+/-
An Agile EAM Method should support appropriate representation of architectural artifacts and communicate architectural documentations across the enterprise in a common language.	+/-	+	+/-	+/-
An Agile EAM Method should provide mechanism to develop and maintain an integrated architecture repository for knowledge management and enterprise awareness initiatives.	+/-	+	Unknown	Unknown
An Agile EAM Method should act as a dynamic capability (e.g. as sensing, learning, integration, and coordination capabilities), and support enhancement of agility capabilities and development of agility providers of the enterprise.	-	+	+	+
An Agile EAM Method should promote the anchoring/integration of its mechanisms and tools into the culture of the organization.	+/-	+	+	+

Table 2 EA of Case Study companies mapped to Essential Elements of Lumor [33]

The three case companies that incorporated (most of) the eight EEs into their EA disciplines report to experience little to no friction between EA and (agile) IT development, while still enabling them to evolve the organization in a structured way based on the strategy, vision and goals of the business.

Whereas the one case company that did not incorporate (most of) the eight EEs into its EA discipline reports friction between EA and (agile) IT development. The EEs that they did not incorporate are:

An Agile EAM Method should maintain appropriate balance between long-term and short-term orientation and satisfy EA stakeholders through early, continuous and incremental delivery of effective and efficient EA products towards achieving a (shifting) long term goal/strategies.

An Agile EAM Method should provide an appropriate decision making and governance mechanisms (e.g. collaborative decision making and consensus building) that result in participation and commitment of all stakeholders, and decisions that benefits the enterprise as a whole rather than individual domains.

An Agile EAM Method should act as a dynamic capability (e.g. as sensing, learning, integration, and coordination capabilities), and support enhancement of agility capabilities and development of agility providers of the enterprise.

If you look at the main differences between the three organizations without friction and the organization with friction, two stand out. The three organizations without friction...

... viewed their EA discipline as a strategic ability to transform the organization in a structured way and realized that EA needs to address the entire organization, not just the IT side, so they made sure both the business and IT were involved, or at least integrated into the EA.

... recognized the need for an EA discipline capable of steering the organization, so they placed the (Lead) Architects high up in the organization, so they have mandate and authority through the hierarchical lines. Furthermore, they made the architecture community into one team, that reported to other architects up the chain (even if they were not one organizational unit).

When looking at the three missing EEs and the two main differences in EA implementation and organization, it seems that these are related to each other:

Stating that the EA discipline is of strategic ability and making sure that both IT and business are involved, as well as giving the EA discipline the organizational structure and mandate to architect the organization enables them to ensure that all three missing EEs could be fulfilled and executed in the organization.

6 Limitations, Conclusions and Future Work

6.1 Limitations

Several limitations can be seen regarding this research. Perhaps the most pressing one is sub research question three: when setting up the research, extensive talks and meetings were already held and attended within the main case company (Rabobank WRR). These talks and meetings led to the research questions used in this research, as the organization was experiencing friction between (agile) IT development and their EA effort. This led to the proposition that they were related and had to be researched. The information gathering focused heavily on this proposition and therefore gathered information on EA and agile IT development, also at the intersection of the two, which hardly yielded results. In the case study it was attempted to delve deeper into the friction, but statements and results seemed to not corroborate the statement. It was only when interviews at other financial organizations started to take place that it became clear that EA and agile IT development weren't experiencing friction because of their different natures (long-term vs short-term), but that it was related to the way EA was organized at Rabobank WRR. As such, the third sub question proved to be irrelevant and easy to answer. When (during the analysis phase) revised to the alternative version, it made much more sense and almost immediately answered the main research question. All in all, this should not have too big of an impact on the research, as an important conclusion is that a correct way of organizing and implementing EA will improve the responsiveness of (agile) IT development nonetheless.

Furthermore, during the case study a lot of evidence and statements were gathered. Not all of it was useful for this research, so during the case study description the information was somewhat filtered in order to be able to write a coherent case, instead of a book-like story that did not provide much useful information. In the end the most important information was used, it is not likely that by describing all information collected in the case study, the conclusions would have been different from what they are now.

The case study research addressed one organization very elaborate. The other cases had one to three interviews per case, so they yielded very general information regarding those cases. Interviews took place with architects in high positions, so their information might have had some sort of bias or might have been distorted because of their position in the hierarchy. Nonetheless, they described their EA discipline quite elaborately, it did not seem that they were giving utopian information to "look better". They also had nothing to gain by giving distorted information and were very forthcoming and open regarding their processes, departments, motivations etcetera.

One big Dutch bank was not interviewed, which is also the bank that is working in a very agile way. They are working fully according to DevOps, have an elaborate EA discipline, but none of the attempts to contact them was answered. Even through the global manager of Architectural Services within Rabobank WRR no contact was made.

Finally, there is no validation done in this research. There are two reasons for this: the case study was very elaborate and therefore took a lot of time, resulting in the fact that no validation could be done within the timeframe of conducting the research and thesis. Furthermore, just after the information gathering ceased within Rabobank WRR (around may 2016), the organization had a major reorganization, resulting in the merger of Rabobank WRR and Rabobank IT Netherlands into one entity: one bank. This resulted in the fact that the organizational department which facilitated this research ceased to exist, all architects were placed elsewhere in the organization, some even left. This had such a big impact on the organization and its employees that validation was rendered impossible at the time. It would have been most interesting to present these results at Rabobank WRR and see if any changes could be made in order to improve their EA discipline. The other banks already reported to be working this way, so the validation would not mean much if done at those organizations.

6.2 Conclusions

This section will first discuss three the sub research questions and answer them, before an answer to the main research question is devised. The answers to these research questions are based on the analysis (Chapter 5 Analysis) of information found from real world settings (Chapter 4 Case Study) and scientific sources (Chapter 3 Theoretical Background) as described and planned (Chapter 2 Research Approach) based on the research questions put forward to solve the problem statement (Chapter 1 Introduction).

Sub Research Question 1

“How is the Enterprise Architecture discipline implemented in agile working organizations?”

While organizations have different ways of implementing EA within their organization, generally they recognize Enterprise Architecture to be comprised of more or less the same components.

There are models that describe the organizations. An organization starts with defining the **Vision** and **Mission** of the organization, which lead to a **Business Strategy** and **Business Goals**. The Business Strategy and Business Goals require the organization to be **structured** and **function** in a certain way (**Business Architecture**). In order to be able to achieve your goals and follow your strategy there need to be **processes** in your organization: they are depicted through **(Target) Operating Models**. These (Target) need **information** as input, which is defined in the **(Target) Information Architecture**. The (Target) Information Architecture will need to store its information in **systems** and **applications**: the **(Target) Application Architecture**. This (Target) Application Architecture needs **infrastructure** and **hardware** to be put onto: the **(Target) Infrastructure Architecture**.

There are different kinds of architectures: the **Current State Architectures**, which describe the different architecture landscapes as they currently are. Then there are the **Target State Architectures**, which describe the way the architecture landscapes should look like to enable the organization to follow its strategy and achieve its goals. Usually there is a difference between Current State and Target State Architectures: **gaps**. These gaps would need to be addressed, because the organization otherwise would not be able to execute its processes in the most efficient way (as the Target State Architectures are modeled based on the current strategy and vision). Therefore **Roadmaps** are created to provide the business with a tool to address the organization's architecture and close the gaps and enable the organization to evolve in a structured way, so that it can achieve its goals and strategy as efficiently as possible.

Other EA components are guidelines and standards to guide the transformation in such a way that the architectural landscapes are coherent and of high quality, ensuring the ability to change it on the next change of strategy and/or goals.

The EA discipline needs to be placed in the organization, this can be centralized or decentralized, with a lot of authority, or not so much authority. This placement depends on how the MT or Board of Directors views the EA discipline: is EA important and of strategic importance, or is EA a tool in order to govern the IT departments. If EA is thought to be of strategic importance, EA will most likely be positioned high in the hierarchical line, perhaps even directly under the MT or Board of Directors, if not of strategic importance, it will be positioned lower. The higher EA is placed in the organization, the better they can enable the organization to evolve in a structured way.

The conclusion to this sub question is therefore:

Organizations implement EA in different ways, usually recognizing the same components, but at different places in the hierarchical line. The higher in the hierarchy, the more important EA becomes and the more it can enable the organization to evolve in a structured way.

Sub Research Question 2

“How is IT responding to changes in agile working organizations?”

The way IT is responding in organizations is just like EA very different. Organizations still haven't completely cast aside plan-driven development methods, but are (usually) eager to embrace agile

IT development methods. IT is the backbone of most (financial) organizations, and as such organizations will want to have an IT discipline that can deliver solutions according to the needs of the business.

A way to improve the chances that IT is delivering solutions according to the business' needs, is to do so through agile IT development. Agile IT development has "customer collaboration" in high regard, together with "working software" and "responding to changes" [19]. If implemented correctly, these values help IT development to respond to changing requirements of customers during projects. Agile also provides a way to deliver solutions with a minimum of functionality and to keep expanding those functionalities. This way IT is able to provide the customers with working solutions in an early stage of the project.

At the same time financial organizations have a lot of legacy systems in place, which are critical to the business. These systems don't require a lot of development anymore, however, as they provide the core functionalities, such as transactions. Those legacy systems therefore do not require teams that are able to develop solutions and/or change/add functionalities on top of those legacy systems. This can be done in a slow-paced, thoroughly designed, checked and tested project, in order to ensure business continuity.

The conclusion to this sub question is therefore:

IT respond to changes in various ways: if possible they will most likely choose to do so in an agile way, in order to ensure fast incremental deliveries to customers in order to receive early feedback on those increments, which enable them to change the next iteration to deliver the most important functionalities first. If a project needs to be fully in control of everything that happens regarding a (legacy) system, they will most likely choose to do so in a plan-driven way, to ensure thoroughly designed, developed and tested solutions/changes, with extensive functional/technical documentation to ensure future work knows how the system is designed and built.

Sub Research Question 3

What are the differences between how IT aims to respond to (disruptive) innovations and how the enterprise architecture discipline is implemented?

When looking at the conclusions from sub research question one and two, the third sub research question (SRQ3) should have been formulated differently. EA and (agile) IT development aren't competing with each other, instead they complement each other. This means that there are no differences or similarities to be identified, but rather expectations regarding input and support.

So instead of

"What are the differences between how IT aims to respond to (disruptive) innovations and how the enterprise architecture discipline is implemented?",

a better research question would have been:

"What do IT development and enterprise architecture expect of each other in terms of input and support?".

This would have (most likely) led to the same results as above, but those would have led to a direct answer of the sub research question.

However, in this case, the answer to the research question has to be:

There are no differences (or similarities for that matter) between how IT aims to respond to (disruptive) innovations and how the enterprise architecture discipline is implemented. Instead of comparing the two, it should be described how they can complement each other, by meeting each other's expectations regarding input and support:

Enterprise Architecture supports (agile) IT development by defining gaps in the IT architecture and providing them with a general (high-level) solution on how to (partially) close that gap, while guiding them through principles and standards.

Enterprise Architecture expects (agile) IT development to adhere to these principles and standards, while working towards the general solution so that the input of (agile) IT development towards Enterprise Architecture will be a solution, constructed under architecture, which (partially) closes a gap in the IT architecture.

In turn, (agile) IT development supports Enterprise Architecture by delivering the requested solution used to (partially) close the gap, while working under architecture.

(Agile) IT development expects from Enterprise Architecture an architecturally sound (high-level) design, so that it can create the solution as it sees fit, while adhering to the principles and standards.

When plan-driven IT needs to respond to (disruptive) changes, there is no need to act as fast as possible, but not much changes with regard to the above statements: there are still principles and standards to adhere to and gaps to be closed.

Main Research Question

“How can the enterprise architecture discipline (including its governance, models and artifacts) be (re)designed, with the aim to improve the responsiveness of IT in agile working organizations operating in the financial domain?”

With the insight of SRQ3 the answer to the Main Research Question is rather simple:

The Enterprise Architecture Discipline, implemented in its traditional form is not necessarily an impediment for IT development in agile working organizations. Therefore, the EA discipline does not need to be (re)designed to improve the responsiveness of IT.

The way EA can improve the responsiveness of IT if the leaders (i.e. MT, Board of Directors) of the organization recognize the value of architecting the organization from top (vision and mission) to bottom (Infrastructure Architecture Landscape) and implements EA accordingly by placing it high enough in the hierarchical organization so that it has a mandate and is an equal partner to the business. This way EA can allow the organization to transform in a structured way so that it can fulfill its strategy and goals, by having an effective and efficient organization.

When compared to an organization without such an EA discipline, the IT responsiveness would be improved because the Architectural Landscapes are already known (without EA you would need to construct them at the beginning of the project), when anything in the landscape changes, you already know the gap. IT immediately knows that gap and what needs to be done in order to close it (a dot on the horizon). And by having devised the guidelines and standards, teams know the boundaries within which they are allowed to maneuver and the rules to which they need to adhere.

If the eight Essential Elements proposed by Lumor [33], are taken into account when implementing the EA discipline within organizations, it is likely that the EA will improve the responsiveness of IT.

6.3 Future Work

Future work should focus on validation of this research. First, it would be interesting to know if EA really doesn't impact EA if implemented taking the eight Essential Elements of Lumor [33] into account. Special attention should be given to the two causes of EA failure found in Rabobank WRR: will they always cause failure, or can a decentralized EA discipline with mandate still be successful for example, even if the business isn't involved? Or vice versa.

Further validation work could look at other organizations in the financial domain in order to validate that EA and agile IT development can complement each other if organized and implemented correctly.

Finally, this research might be much more generalizable, beyond the financial domain. Any IT dependent organization that's developing in an agile way might benefit from the conclusions in

this research and implementing a good EA discipline. Future research could aim to generalize it beyond the financial domain.

7 References

1. Abraham R, Aier S, Winter R (2012) Two Speeds of EAM—A Dynamic Capabilities Perspective. In: Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation. Springer, pp 111-128
2. Abrahamsson P, Salo O, Ronkainen J, Warsta J (2002) Agile software development methods: Review and analysis. VTT Finland,
3. Arner D, Barberis J, Buckley R (2015) The Evolution of FinTech: A New Post-Crisis Paradigm?
4. Beck K (1999) Embracing change with extreme programming. *Computer* 32 (10):70-77
5. Beck K (2000) Extreme programming explained: embrace change. addison-wesley professional,
6. Brynjolfsson E, McAfee A (2014) The second machine age: work, progress, and prosperity in a time of brilliant technologies. WW Norton & Company, New York
7. Buckl S, Matthes F, Monahov I, Roth S, Schulz C, Schweda CM Towards an agile design of the enterprise architecture management function. In: 2011 IEEE 15th International Enterprise Distributed Object Computing Conference Workshops, 2011. IEEE, pp 322-329
8. Bygstad B, Ghinea G, Brevik E (2008) Software development methods and usability: Perspectives from a survey in the software industry in Norway. *Interacting with computers* 20 (3):375-385
9. Chen X, Fang Y (2013) Enterprise systems in financial sector—an application in precious metal trading forecasting. *Enterprise Information Systems* 7 (4):558-568
10. Christensen C (2013) The innovator's dilemma: when new technologies cause great firms to fail. Harvard Business Review Press, Boston, Massachusetts
11. Committee AWGotSE (2000) Recommended Practice for Architectural Description of Software Intensive Systems. IEEE Standards Department.
12. Conway ME (1968) How do committees invent. *Datamation* 14 (4):28-31
13. Denscombe M (2014) The good research guide: for small-scale social research projects. McGraw-Hill Education, United Kingdom
14. Dyba T (2000) Improvisation in small software organizations. *IEEE Software* 17 (5):82
15. Dybå T, Dingsøyr T (2008) Empirical studies of agile software development: A systematic review. *Information and software technology* 50 (9):833-859
16. Edwards C (2006) Agile Enterprise Architecture, Part 1. USA: ProcessWave
17. Erol O, Sauser B, Boardman JT (2014) Creating Enterprise Flexibility Through Service-Oriented Architecture. In: The Flexible Enterprise. Springer, pp 27-36
18. Fehskens L Re-thinking architecture. In: 20th Enterprise Architecture Practitioners Conference, The Open Group, 2008. pp 75-106
19. Fowler M, Highsmith J (2001) The agile manifesto. *Software Development* 9 (8):28-35
20. Graves T (2008) Real Enterprise Architecture. Tetradian,
21. Greefhorst D Service Oriented Enterprise Architecture. In: Proceedings of the 2nd Workshop on Landelijk Architectuur Congres, 2006.
22. Hauder M, Roth S, Schulz C, Matthes F Agile enterprise architecture management: an analysis on the application of agile principles. In: International Symposium on Business Modeling and Software Design BMSD, 2014.
23. Henderson JC, Venkatraman H (1993) Strategic alignment: Leveraging information technology for transforming organizations. *IBM systems journal* 32 (1):472-484
24. Hensema M (2015) Applying Agile in Enterprise Architecture.

25. Higgins JM (1985) Strategy Formulation, Implementation and Control. Dryden Press, New York
26. Highsmith J, Cockburn A (2001) Agile software development: The business of innovation. *Computer* 34 (9):120-127
27. Hoogervorst J (2004) Enterprise architecture: Enabling integration, agility and change. *International Journal of Cooperative Information Systems* 13 (03):213-233
28. Kelly J (2015) Blockchain technology could reduce role of banks, says BIS. <http://www.reuters.com/article/global-banking-blockchain-idUSL8N13I3VG20151123>. Accessed November 2015
29. Lankhorst M, Proper H (2012) Agile architecture. In: *Agile Service Development*. Springer, pp 41-57
30. Lankhorst MM, van Gils B (2015) Building Agile Enterprises: A Model-Based Approach to Rapid Realization of Business Value. In: *Business Architecture Management*. Springer, pp 205-225
31. Lapalme J, Gerber A, Van der Merwe A, Zachman J, De Vries M, Hinkelmann K (2015) Exploring the future of enterprise architecture: A Zachman perspective. *Computers in Industry*
32. Louise Barriball K, While A (1994) Collecting Data using a semi-structured interview: a discussion paper. *Journal of advanced nursing* 19 (2):328-335
33. Lumor T (2016) Towards the design of an agile enterprise architecture management method.
34. McCauley R (2001) Agile development methods poised to upset status quo. *ACM SIGCSE Bulletin* 33 (4):14-15
35. Miller GG The characteristics of agile software processes. In: *tools, 2001*. IEEE, p 0385
36. Moore GE (1975) Progress in digital integrated electronics. *IEDM Tech Digest* 11
37. Morabito J, Sack I, Stohr E, Bhate A (2014) Architecting Flexible Organizations. In: *The Flexible Enterprise*. Springer, pp 113-135
38. Moreno J, Gera P, Colangelo M, Proverbio M, Skan J, Busch W, Caroll G (2015) *The Everyday Bank - A New Vision for the Digital Age*.
39. Nerur S, Balijepally V (2007) Theoretical reflections on agile development methodologies. *Communications of the ACM* 50 (3):79-83
40. Nielsen J (1998) Nielsen's Law of Internet Bandwidth. <https://www.nngroup.com/articles/law-of-bandwidth/>. Accessed November 2015
41. Okoli C, Schabram K (2010) A guide to conducting a systematic literature review of information systems research. Available at SSRN 1954824
42. Op't Land M, Proper E, Waage M, Cloo J, Steghuis C (2008) *Enterprise architecture: creating value by informed governance*. Springer Science & Business Media,
43. Open, Group, The (2011) *TOGAF, an Open Group Standard*. <http://www.opengroup.org/subjectareas/enterprise/togaf>. Accessed November 2015
44. Porter ME (1979) How competitive forces shape strategy.
45. Proper EA, Lankhorst MM (2014) Enterprise Architecture: Towards essential sensemaking. *Enterprise Modelling and Information Systems Architectures* 9 (1):5-21
46. Rabobank (2016) Changes to organisational design at Rabobank Nederland. https://www.rabobank.com/en/press/search/2016/20160419_organisational_design.html. Accessed December 2016
47. Rabobank (2016) History. <https://www.rabobank.com/en/about-rabobank/profile/history/index.html>. Accessed December 2016
48. Rabobank (2016) Who we are. <https://www.rabobank.com/en/about-rabobank/profile/who-we-are/index.html>. Accessed December 2016

49. Rabobank, Group (2016) Interim Report 2016.
50. Reifer DJ (2002) How good are agile methods? *Software, IEEE* 19 (4):16-18
51. Riempp G, Gieffers-Ankel S (2007) Application portfolio management: a decision-oriented view of enterprise architecture. *Information Systems and E-Business Management* 5 (4):359-378
52. Romein J (1937) *Het onvoltooid verleden: kultuurhistorische studies*. Querido Amsterdam, Amsterdam
53. Ross JW, Weill P, Robertson D (2006) *Enterprise architecture as strategy: Creating a foundation for business execution*. Harvard Business Press,
54. Shirazi H, Rouhani B, Shirazi M (2009) A framework for agile enterprise architecture. *International Journal of Intelligent Information Technology Application*, 2 (4): 182 186
55. Spewak SH, Hill SC (1993) *Enterprise architecture planning: developing a blueprint for data, applications and technology*. QED Information Sciences, Inc.,
56. Stettina CJ, Heijstek W Necessary and neglected?: an empirical study of internal documentation in agile software development teams. In: *Proceedings of the 29th ACM international conference on Design of communication*, 2011. ACM, pp 159-166
57. Sushil (2014) *The Concept of a Flexible Enterprise*. In: *The Flexible Enterprise*. Springer, New Delhi, India, pp 3-26
58. Sushil, Stohr EA (2014) *The Flexible Enterprise*. Springer, New Delhi, India
59. van de Weerd I, Brinkkemper S (2008) Meta-modeling for situational analysis and design methods. In: Rahman M, Nessa S (eds) *Handbook of research on modern systems analysis and design technologies and applications*. Information Science Reference, Hershey, New York, pp 35-54
60. van Oosterhout M, Waarts E, van Hillegersberg J (2006) Change factors requiring agility and implications for IT. *European Journal of Information Systems* 15 (2):132-145
61. Wagter R (2009) *Sturen op samenhang op basis van GEA®*. Van Haren,
62. Walter C (2005) *Kryder's Law*. <https://www.scientificamerican.com/article/kryders-law/>. Accessed November 2015
63. West D, Grant T (2010) *Agile Development: Mainstream Adoption Has Changed Agility*. Application Development & Program Management Professional. Forrester, Cambridge, USA
64. Yin RK (2009) *Case Study Research: Design and Methods: Design and Methods*, vol 5. Sage Publications, California
65. Yu E, Deng S, Sasmal D (2012) Enterprise architecture for the adaptive enterprise—A vision paper. In: *Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation*. Springer, pp 146-161
66. Zachman J (1987) A framework for information systems architecture. *IBM systems journal* 26 (3):276-292

Appendices

[First appendix on next page]

A Interview Protocol

Interviewee

- Name, Function / Role
- Short description of the domain / department / unit

Enterprise Architecture

- Governance
 - Are there any restricting factors/impediments that arise from the current organizational structure?
 - How do you see your own role in the future? From the designer towards coach?
 - Architecture Board
 - Improvements
- Process
 - Where is the line drawn between EA, local architecture and SA? And where should it be? Project engineer architecting SA vs enterprise architect architecting own landscape.
 - Improvements
- Deliverables
 - PSA
 - Improvements

Domain Architecture

- Governance
 - How often does the Local Architecture Board meet?
 - Who is in the LAB?
- Process
- Deliverables
 - PSA

(Agile) IT Development

How does the domain respond to changes and innovations? Is it initiated from the business alone, or also from within the domain?

- Process
- Deliverables
 - Before
 - PSA
 - Are the EA IT principles and bricks used, or is it more like FFA?
 - During
 - After
 - Does a project often deviate from the PSA?

Can you give a top 3 of good and a top 3 of bad properties/activities of agile SDM?

Friction between Enterprise Architecture and (Agile) IT Development

Agile Way of Working

- What do you think a more general agile way or working means?
- Can you see agile being extended to other aspects of the organization as well?
- Is the domain planning on expanding agile to a more general way of working?
- Will this work?
- Is Rabobank WRR ready to work in an agile way?

General

- Is there anyone else I should talk to?
- Other remarks?

B Tags used for Interview Results Summary

Category	Department / function
Enterprise Architecture Deliverables	APS
Enterprise Architecture Governane	Agile Coaches
Enterprise Architecture in General	Business Architecture
Enterprise Architecture Process	Corporate Banking
Domain Architecture Deliverables	Financial Markets
Domain Architecture Governance	Internet Direct Retail Bank
Domain Architecture in General	ITI Architects
Domain Architecture Process	ITI Project Manager
Software Development in General	Rabogroup
Software Development: Agile at Rabobank	Risk & Finance
Software Development: Agile in General	
Software Development: Agile Pro's and Con's	
Software Development: Plan Driven	
Software Development: Plan Driven vs Agile	
Infrastructure Development	
Friction Between Enterprise Architecture and Agile	
Agile way of working	
Uncategorized	

Table 3 Tags Used for Interview Results Summary

C Interview Results Summary Example

Initials	Category	Department / function	Statement
EDN	Agile way of working		Edwin is not sure if the domains are extending agile to their general way of working.
EDN	Domain Architecture Deliverables	Risk & Finance	started working on a Reference Architecture for R&F
EDN	Domain Architecture Governane	Risk & Finance	Risk and Finance has a local architecture board
EDN	Domain Architecture Governane	Risk & Finance	business side is also represented in the Risk and Finances local architecture board
EDN	Enterprise Architecture Deliverables		PSAs are a deliverable, which are being created before projects
EDN	Enterprise Architecture Deliverables		The PSA proposes the general outlines, not a specific design.
EDN	Enterprise Architecture Governane		The WRR architecture board within WRR meets every week
EDN	Enterprise Architecture Process	Risk & Finance	Edwin is regularly contacted before a PSA is even submitted, to review it and provide feedback.
EDN	Enterprise Architecture Process		Sometimes the PSA will be held on hold, because further information/clarification is needed.
EDN	Friction Between Enterprise Architecture and Agile		There is the risk that everyone just does their own thing and loses sight of the context
EDN	Software Development: Agile at Rabobank	Risk & Finance	R&F mainly develops in an agile way, but whether or not all teams work at the same level of agile development?
EDN	Software Development: Agile Pro's and Con's		advantage is the focus on communication and interaction, both within the team as between the team and the business
EDN	Domain Architecture Deliverables	Risk & Finance	R&F created a document with a target state architecture, depicting gaps and how they globally aim to fill those gaps
EDN	Domain Architecture Governance	Risk & Finance	The local architecture board of Risk and Finance has the architecture discipline, a system owner representative and the business.
EDN	Domain Architecture in General	Risk & Finance	Edwin joins the weekly architecture meeting from Finance and Risk IT, with which they aim to be the bridge between the domains
EDN	Uncategorized	APS	the architects aim to align the three different domains (Ronald: Financial Markets, Pieter Jan: Corporate Banking and Edwin: Finance and Risk IT).
EDN	Friction Between Enterprise Architecture and Agile		Things are being treated too isolated, within their own boundaries, but the bigger context is not always taken into account.

Table 4 Interview Results Summary Example