October 2015

MASTER THESIS

Applying Agile in Enterprise Architecture

Martijn Hensema

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)

Exam committee:
Maria Iacob (University of Twente)
Marten van Sinderen (University of Twente)
Floris Jansen (Deloitte Consulting)

Deloitte. UNIVERSITY OF TWENTE.

Management Summary

Enterprise Architecture (EA) is a relatively young discipline that has evolved over the years. Challenges such as the ivory tower syndrome, keeping up with the changing environment or delivering results on time are commonly experienced. Agile principles, originating from software engineering, might offer possible improvements.

Agile practices can address EA Challenges. More specifically the following agile practices are of importance: the ability to deal with changing requirements, reflecting on the process and a focus on the essential. An emphasize on these agile practices is very likely to result in less EA challenges perceived in an organization. Interesting is that certain key agile practices such as incremental and iterative development are not significant.

The conclusion is based on a literature review, qualitative study and expert review. The literature review provided three pieces of information: (1) a decomposition of agile principles in eleven agile practices, (2) fourteen challenges related to EA development and (3) partial evidence that current EA development adheres to waterfall principles and, like software development, might benefit from agile principles. The literature review also revealed that agile practices are already being applied to EA. In these cases the assumption is made that this improves EA development. Validating this assumption is not within the scope of previous research. Based upon the literature a conceptual model was defined reflecting a positive relationship between agile practices applied to EA and the EA challenges perceived.

The conceptual model is validated with a qualitative study. Data was gathered with the use of a survey targeted at enterprise architects. The data gathered gives information on the agile practices applied to EA and the EA challenges perceived in the organization. With the use of statistical methods the data was analyzed. This resulted in the following practices: the ability to deal with changing requirements, reflecting on the process and a focus on the essential. The practices are significant over all the cases in predicting the EA challenges perceived in an organization.

Acknowledgements

This thesis concludes my master Business Information Technology and my life as a student at the University of Twente. The last couple of months have been an interesting journey. I have learned a lot and have been able to develop myself.

I conducted my research as a graduate intern at Deloitte Consulting within the service line Enterprise Architecture. Being able to conduct my research there has been a great experience. The opportunity at Deloitte Consulting has made this thesis possible.

Firstly I would like to thank my supervisors Maria Iacob and Marten van Sinderen from the University of Twente for their support. Their guidance and feedback have helped in improving and structuring my research.

Secondly I would like to thank Floris Jansen. His supervision has helped me in conducting my research. Discussions and critical feedback from him have helped me enormously in improving my thesis. I would also like to thank Eric Onderdelinden for his valuable insights.

A special thanks to all the people that have contributed to my research by providing data on their organization. Without their help my research would not have been possible.

Finally I would like to thank my family who has supported me during my years of study.

Contents

List of Figures

List of Tables

Abbreviations

ADM	**A**rchitecture **D**evelopment **M**ethod
ASD	**A**gile **S**oftware **D**evelopment
DSRM	**D**esign **S**cience **R**esearch **M**ethodology
EA	**E**nterprise **A**rchitecture
IS	**I**nformation **S**ystem
TAFIM	**T**echnical **A**rchitecture **F**ramework for **I**nformation **M**anagement
TOGAF	**T**he **O**pen **G**roup **A**rchitecture **F**ramework
VIF	**V**ariance **I**nflation **F**actor

Chapter 1

Introduction

1.1 Introduction

This thesis answers the question as to how agile principles originating from software development can be incorporated in the development of an enterprise architecture (EA[1]). Before diving deeper into the subject of this thesis, we will take a small step back to look at the bigger picture of EA. The bigger picture allows us to position this thesis.

Pinpointing when the EA discipline was exactly introduced is difficult. In the late 80s the information systems architecture [1] was published. Although the term EA was never used in the article, it is considered to be the starting point of the EA discipline. The information system architecture evolved into the Zachman enterprise architecture framework. Even though the framework was refined over time it never extended beyond a set of concepts that could be placed within a matrix [2]. EA started off as what could be considered an ontology[2].

At the time how to visualize an architecture or transition from an as-is to a to-be architecture were not a matter of discussion. As EA grew in popularity new frameworks were introduced which reached beyond an ontology. Early examples include the planning process of Technical Architecture Framework for Information Management (TAFIM) [4],

[1] Definition of EA used is available in Chapter 1.2.1

[2] Ontology – "in the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application." [3]

or more widely used nowadays the Architecture Development Method (ADM) part of TOGAF [5][2]. Both frameworks present a systematic method and a set of principles to guide EA. These frameworks can be considered methodologies[3].

Conceptually EA has changed, but its focus has also shifted. EA originated from the information systems (IS) field [1] which is one of the reasons for the strong focus on IT, but this has been changing over the years [2]. The strong focus on IT is also fueled by the IT background of many architects [7]. Nowadays the focus is on balancing between the business and the IT [8]. This evolvement has conceptually changed EA. It has become a discipline which links these two fields together. EA has become broader and concerns more concepts. Complexity increases and what is demanded from the architects changes.

The linking of business goals to IT is commonly referred to as business IT alignment. Maintaining this alignment has become an inherent part of EA: it is a tool to realize strategy by transforming business and IT [2]. EA frameworks maintain the alignment between business and IT [9].

This thesis shows that the methods underlying EA frameworks resemble methods from software development. This is consistent with the field EA originated from. Software development is moving from waterfall principles to agile principles. EA developments still adheres waterfall principles. EA development has not made the transition to agile principles. This transition could very well be the next step in EA.

1.2 Research Background

The combination of two concepts, enterprise architecture and agile software development (ASD), are of importance to this thesis. Both concepts are used frequently throughout the thesis. To create a common understanding each concept is defined. Other concepts are defined when used.

1.2.1 Enterprise Architecture

Before selecting a definition for EA, a number of different definitions are presented.

[3]Methodology – "body of methods, rules, and postulates employed by a discipline: a particular procedure or set of procedures." [6]

The following definition is taken from IEEE 42010:2011 [10]. The concept defined is architecture. It has a focus on the formal description of a system. A part of definition (principles of its design and evolution) endorses that a system is subject to change.

> "Fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution."

The following definition is taken from TOGAF [5] and is based upon that from IEEE 42010:2007. The difference is in the first part (1) of the definition which emphasizes that architecture should focus on moving from a current state to a future state.

> "Architecture has two meanings depending upon the context (1) A formal description of a system, or a detailed plan of the system at component level to guide its implementation (2) The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time."

The following definition for enterprise is used by TOGAF [5].

> "TOGAF defines "enterprise" as any collection of organizations that has a common set of goals. For example, an enterprise could be a government agency, a whole corporation, a division of a corporation, a single department, or a chain of geographically distant organizations linked together by common ownership."

The last definition is from Gartner [11].

> "Enterprise architecture (EA) is a discipline for proactively and holistically leading enterprise responses to disruptive forces by identifying and analyzing the execution of change toward desired business vision and outcomes."

Formulating a new definition by tailoring existing definitions seems pointless. The formulated definition will be just another definition in line with many others. The goal of

this thesis is not formulate a new definition of EA. Considering these points an existing definition is followed.

The definition from Gartner is more an explanation of a possible use of EA than it is a definition. The definition explains how business and IT alignment is realized with the use of EA. Although this is the only definition that incorporates enterprise, the enterprise itself is not defined in the definition.

The definition from IEEE is too limited, it focuses merely on the definition of a system and does not emphasize change. TOGAF does do this. Considering that the Gartner definition is focused more on the use of EA, the TOGAF definition is followed.

1.2.2 Agile Software Development

Again a numb er of definitions from literature are presented. After which one of the definitions is selected.

The following definition [12] describes agile in terms of flexibility, speed, leanness, learning and responsiveness.

> "Agility is a persistent behaviour or ability of a sensitive entity that exhibits flexibility to accommodate expected or unexpected changes rapidly, follows the shortest time span, uses economical, simple and quality instruments in a dynamic environment and applies updated prior knowledge and experience to learn from the internal and external environment."

A different definition, from the same authors but a different paper [13], is an extension of the above definition. Both definitions attempt to define ASD by explaining the individual components.

> "A software development method is said to be an agile software development method when a method is people focused, communications-oriented, flexible (ready to adapt to expected or unexpected change at any time), speedy (encourages rapid and iterative development of the product in small releases), lean (focuses on shortening timeframe and cost and on improved quality),

> responsive (reacts appropriately to expected and unexpected changes), and learning (focuses on improvement during and after product development)."

The following definition is taken from a book by Ian Sommerville: Software Engineering [14]. This definition follows the pattern of the two previous definitions. The concept is again defined by its components.

> "Methods of software development that are geared to rapid software delivery. The software is developed and delivered in increments, and process documentation and bureaucracy are minimized. The focus of development is on the code itself, rather than supporting documents."

The definition used throughout the thesis needs be abstract, i.e. not specific to an implementation of agile such as SCRUM or XP. A too specific definition is likely to focus on software development specific elements which are not applicable to EA.

The second definition is used. It is slightly more comprehensive. All three definitions emphasize the importance of addressing uncertainty and change, but the second definition also incorporates the softer side of agile development: people focused, communication oriented and the learning aspect. Both are important principles according to the agile manifesto [15].

1.3 Problem Statement

Figure 1.1 visualizes the problem context. EA development[4] is under pressure from external and internal forces to change. The problem statement elaborates upon the context.

Transformations that should be the result of EA development take too long or the transformations have already started without EA artifacts[5]. The EA artifacts are too late, recent research [16] indicates that 38% of the organizations are struggling with outdated EA results. When looking at TOGAF, a well-known and wide-used EA framework [2], it is described as too heavy, slow and documentation driven [17].

[4] EA development - all activities and projects that are executed to realize EA artifacts.
[5] EA artifact - deliverable or product from the architect(s).

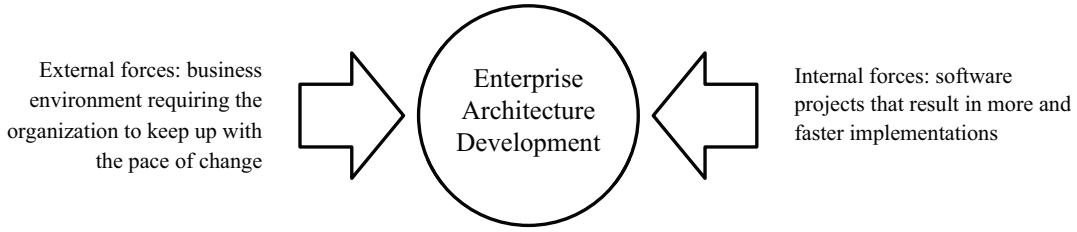

FIGURE 1.1: Drivers behind applying agile in EA

Changes might require re-alignment in the architecture, which in turn results in changes in the IT landscape or to the business. Components, their relationships and principles might need to be reconsidered. Having an architecture gives an organization an overview of their IT and their business.

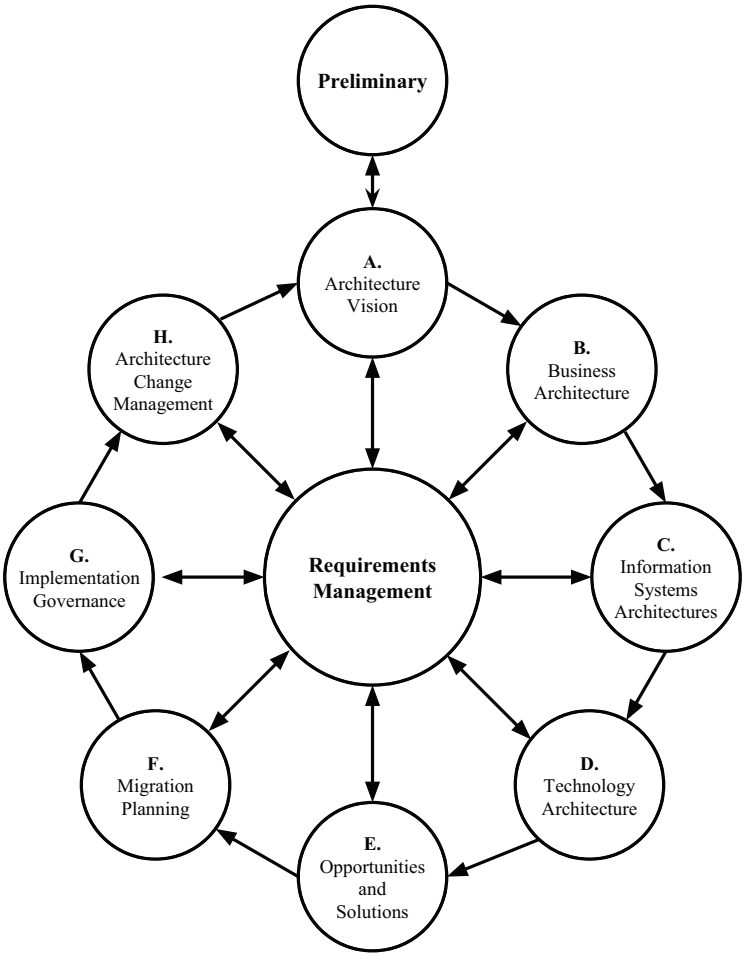

FIGURE 1.2: ADM cycle from The Open Group Architecture Framework (TOGAF) [5]

EA is supported by frameworks such as Zachman Framework or The Open Group Architecture Framework (TOGAF). A process is defined by TOGAF for EA development.

This process is called the ADM shown in figure 1.2, it has an iterative cycle that starts with an architecture vision and ends with change management. In the process several phases are completed such as developing the three layers of the architecture: business, information systems (application & data) and technology layer. Continuing to the next phase requires you to finish, review and validate the current phase.

The ADM adheres to waterfall principles from software development: it is documentation driven [17] and the scope is defined beforehand. When strictly following the process of TOGAF, changes in the business require an organization to cycle through most of the phases. Over time the architecture changes: this can result in changes in the business or to the IT landscape. Organizations are subject to change from the external environment, e.g. customers develop new preferences or competitors introduce new products. Organizations adapt to these changes by creating a new strategy. The architecture therefore changes to ensure continued alignment between the business and IT.

There is a misfit between EA development and the context in which it is used [18]. TOGAF, and all other widely used EA frameworks adhere to waterfall principles in developing an architecture. That is if they even define a process, for example the Zachman framework does not prescribe any process or development approach. Waterfall development is suitable when all requirements are known upfront but is not suited to deal with uncertainty, for example requirements that change once they are signed-off.

Software development is demanding faster results from EA development. IT projects, more specifically the development of software, has moved or is moving towards agile software development. In a survey 14 % of the companies indicated that they are employing agile methods [19]. The result is that new applications are implemented more frequently. Which in turn demands faster results from EA development. Organizations from the IT products and services industries, which are faced with rapidly changing environments, deal more frequently with outdated EA development results [20]. If EA development cannot keep up with the software development, is EA then still able to realize its full potential?

In software development people are relying less on the plan-based or traditional methods [19]. Instead agile methods are employed because these are better at dealing with change [19]. Depending less on plan-based methods might be a solution for EA development. So how can agile approaches be incorporated in EA development.

Chapter 2

Research Design

The research methodology is explained first. Based on the methodology research questions and objectives are formulated.

2.1 Research Methodology

The Design Science Research Methodology (DSRM) is followed. DSRM is a methodology that offers an approach for design science research in information systems (IS) [21]. The methodology enables a researcher to understand and improve an artifact in the context of IS.

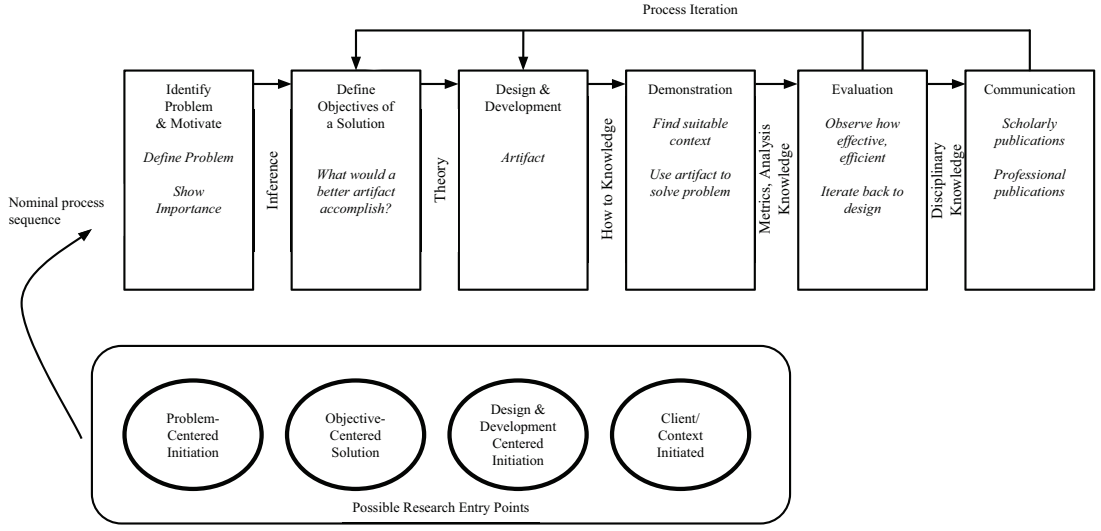

FIGURE 2.1: Design Science Research Methodology (DSRM) [21]

2.2 Research Questions

The research questions are based upon the phases from DSRM. The research entry point is a problem-centered initiation. The process starts at the first phase: identify problem & motivate. Both a mapping of the research questions to the chapters of thesis and a mapping to DSRM are given.

This thesis answers the following main research question. Which is supported by five sub-questions.

How can agile approaches[1] originating from software development be incorporated in the development of an enterprise architecture?

1. What are the current challenges with EA Development?

Existing approaches to EA development exist. Before creating an adaption of the current approach it is of importance to be familiar with the current situation and its problems. This sub-question results in the challenges with current approaches. This maps to the identify problem & motivate phase.

2. What characterizes agile software development?

When applying agile principles to EA it is necessary to know what the principles consist of. This sub-question results in the characteristics of agile software development. The characteristics define the objective of the solution.

3. Does EA benefit from agile approaches?

Before adopting agile principles it needs to be shown that it will result in benefits. The first sub-question results in the current situation which reveals areas of improvement. This combined with the agile characteristics obtained at the second sub-question results in areas that can be improved upon with the use of agile principles. This maps to the design & development phase.

4. What are the requirements for an agile EA approach?

The current situation of EA has its advantages and disadvantages. From this situation requirements can be formulated for an adapted approach to EA development. The

[1] Agile approach – following principles from agile software development.

requirements will be based upon the description of the current situation derived from the literature review and the results of the third sub-question. The requirements describe the solution which maps to the design & development phase.

5. Do the requirements result in an applicable agile EA approach?

The previous sub-question delivers requirements for an agile EA approach. The requirements will validated in order to ensure that these are achievable and applicable in practice. Evaluating whether the requirements for the design are suitable maps to the demonstration phase.

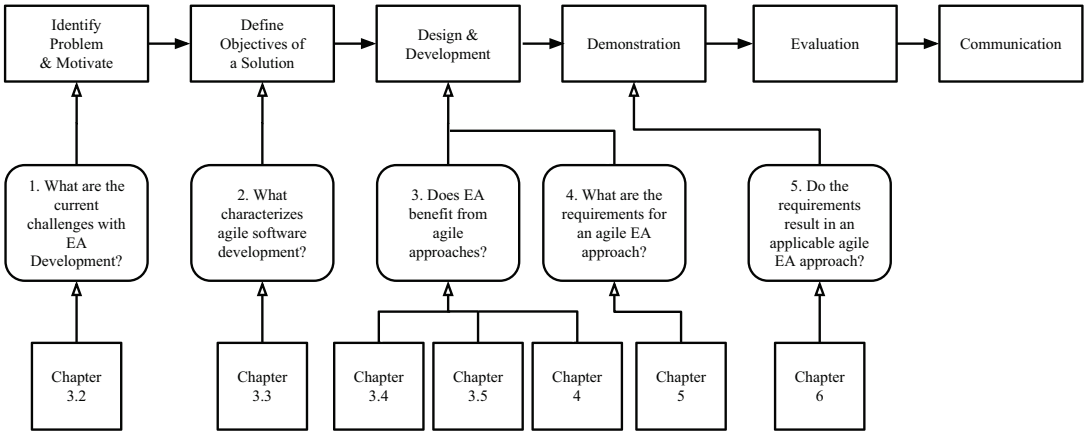

FIGURE 2.2: Overview of structure and research questions

In figure 2.2 a mapping of the research questions to the phases from DSRM and the chapters are given. The last two phases of the cycle are not completed. Due to time constraints these are outside the scope of this thesis. The development of the artifact is limited to the formulation and validation of the requirements.

2.3 Research Objectives

This section elaborates upon the research objectives and the methods and tools that are used to answer the research questions.

This master thesis delivers requirements for an agile approach to EA development. In achieving this objective four steps are taken, each with a separate deliverable. The steps are given figure 2.3. Each step has its' own objective, per phase a short description of the deliverable is given.

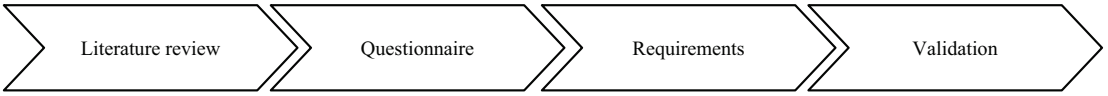

FIGURE 2.3: Steps taken in achieving research objective

The steps in figure 2.3 are sequential. The deliverable of the previous phase serves as input for the next phase. The literature review uncovers gaps in the literature and forms the basis for the questionnaire. The questionnaire together with the literature review is used to define requirements which in turn are validated in the last phase. Although the phases are sequential each phase offers new insights affecting the previous phases.

2.3.1 Literature Review

The literature review establishes the current state of research and uncovers gaps in the field. A structured literature review is done in four distinct areas.

1. List challenges EA development based upon literature. This elaborates on the problem context as described in the problem statement and other areas that can be improved upon.

2. Show similarities between EA development and the waterfall principles from software development. This is a prerequisite in applying agile principles from software development, if no similarities exist it is not a logical step to apply agile principles.

3. List the characteristics of agile software development. In applying agile principles to EA it is important to know what the different aspects of agile software development are.

4. Current state of research on the application of agile within EA. Applying agile principles to EA is not new, existing research on the subject exists.

2.3.2 Questionnaire

The questionnaire is used to gather data from organizations. The challenges perceived regarding EA development and agile practices applied to EA development are measured. The data is used to uncover the relationship between these two concepts.

Statistical analysis

Statistical analysis is applied to the data of the questionnaire. The analysis results in statistically significant relationships within the data. The results are used to formulate requirements.

2.3.3 Requirements

The questionnaire and the literature review serve as input for this phase. The analysis of the data uncovers agile practices that have a significant impact on EA challenges. This results in a list of practices which should be focused on when applying agile to EA. These areas of focus can be used to formulate requirements for an agile approach to EA.

2.3.4 Validation

The requirements are based upon statistical analysis done on the questionnaire, which again was based upon practices and problems from literature. The requirements are the result of statistics, and correlation does not imply causation, validation of these requirements is needed.

Validation is done by an expert review. Outcomes of the questionnaire are discussed with practitioners in order to validate the interpretation given to the data.

Chapter 3

Literature Review

The literature review addresses three phases from DSRM. The literature offers an answer to the first, second and partially the third sub-question. Below an overview of the results from the literature review for each phase are given.

1. **Identify problem & motivate**

   Chapter 3.1 identifies fourteen challenges with EA development: (1) stakeholder commitment, (2) EA governance, (3) stakeholder coordination, (4) stakeholder communication, (5) understanding requirements, (6) shared understanding, (7) architect experience, (8) EA frameworks, (9) knowledge documentation & presentation, (10) tool support, (11) architecture scale, (12) architecture scope, (13) rapidly changing conditions and (14) outdated results. This provides the answer to the following sub-question.

   What are the current challenges with EA Development?

2. **Define objectives of a solution**

   Chapter 3.2 gives a break down of agile principles in eleven agile practices: (1) dealing with changing requirements, (2) frequent delivery of working software, (3) collaboration between business and developers, (4) create trust and motivated individuals, (5) rely on face-to-face communication, (6) working software as measure of progress, (7) maintain a constant pace, (8) technical excellence and good design, (9) focus only on the essential, (10) self-organizing team and (11) reflection on the process. This provides the answer to the following sub-question.

   What characterizes agile software development?

3. **Design & development**

   Chapter 3.3 & 3.4 shows the similarities between EA development and waterfall principles from software development. It also shows that the previous research on applying agile to EA development exists but it is unknown whether agile has an impact on EA development. Together these sections provide a partial answer to the following sub-question.

   Does EA benefit from agile approaches?

Each section first describes the approach taken for the literature review. After which the results of the literature review are described.

3.1 Challenges with EA

This section describes the EA challenges that were formulated based upon literature. First the approach to the literature search is given after which an overview of the articles is given and finally each of the separate EA challenges are explained.

3.1.1 Approach

The literature search is structured in three stages: define, search and select. These are given in figure 3.1. The search is done with the use of two academic search engines: Google Scholar and Scopus. An unstructured approach has also been taken with the use of two other search engines: Web of Science and Microsoft Academic Search. The unstructured search limited itself to the article titles and the first page of results. Together with checking the references of literature found, should ensure that no literature is left uncovered.

The goal, keywords and criteria used for the search are given below.

Goal: to identify papers that describe problems or challenges with enterprise architecture and aggregating these.

Keywords: the following search string was used for uncovering relevant literature: enterprise architecture AND (problems OR issues OR challenges).

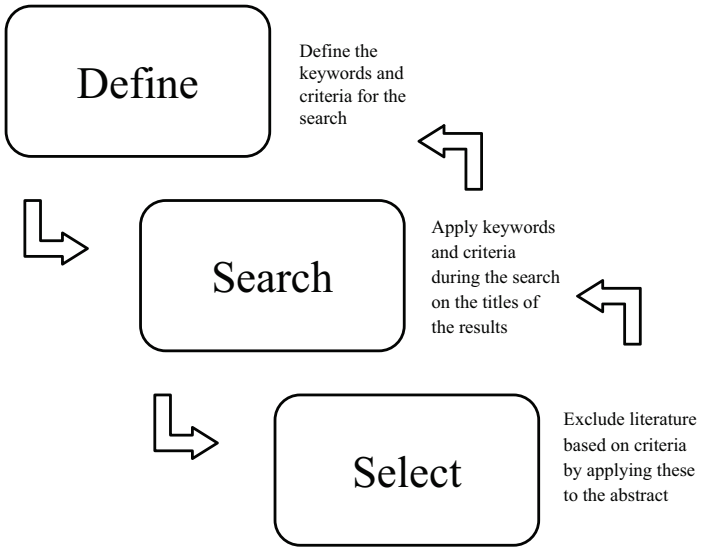

FIGURE 3.1: Approach taken in the literature search

Criteria: using the above search string on Scopus limited to just the title and abstract resulted in over 800 papers, of which many were not relevant. In limiting the amount of results a number of criteria were applied:

- Limited literature to that from the computer science and business management field.
- Excluded the abstract from the search, and only focused on the title of the article. This resulted in a little less than 100 papers. This does severely limit the amount of literature. It is a drastic step to make the search feasible. By backtracking citations of the literature that was found, it was ensured that no literature was left untouched.
- Preference for literature that accompanies a literature study or body of knowledge.

The articles found show overlap in challenges presented which is not unexpected since the articles cite each other. An overview of the citations among the articles is given in figure 3.2. An interesting observation is that almost all the papers reference two articles from Armour et al. [22] [23] (not shown in figure 3.2 because these are not used for the EA challenges). Numerous challenges listed are first mentioned in these two articles, but both articles never had the intention of listing EA challenges. Instead they focused on how to realize EA and in the process indirectly mentioning possible pitfalls.

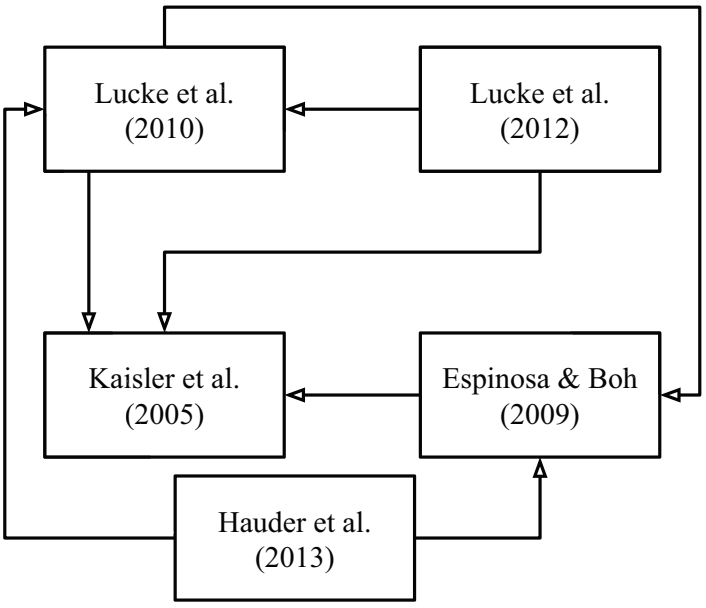

FIGURE 3.2: Citations among the articles

3.1.2 Results

The articles shown in figure 3.2 are used in defining the challenges with EA. The article from Lucke et al. [24] with the refined EA challenges categorization scheme was used as starting point. It is partially an accumulation of the other articles shown in figure 3.2. The empirical evidence of Hauder et al. [20] is used to elaborate upon the categories. The following sections each describe an EA challenge.

Stakeholder Commitment

Stakeholders are of importance to EA. According to TOGAF winning the support of others can make the difference between a successful project and an unsuccessful project [4]. Stakeholder management is part of the ADM cycle. EA is heavily dependent on others [25].

In practice stakeholder commitment is a challenge. 51% of the respondents agree with the statement that they have unavailable stakeholders and 64% have to deal with reluctant information providers [20].

EA Governance

EA governance concerns how EA is controlled and managed within the enterprise [5]. Management and control of EA is frequently absent in organizations [9].

In a study focused on the adoption of EA numerous challenges concerning governance are identified: appointing leadership or ownership, delegation of decision-making rights and responsibilities, insufficient mandate and integration in existing governance [26]. The importance of governance is also emphasized by Esponisa and Boh [25]: 'Even the best target architecture is useless when there is no compliance with the architecture'.

Stakeholder Coordination

Coordination encompasses coordinating the different parties involved in EA. These different parties could consist out of architects each responsible for a domain of the architecture and other individuals who require results from the architects. Because the domains need to be consistent it is important to have coordination among the stakeholders.

Coordination is hindered by two things in particular: (1) geographical distance and separation and (2) time separation [9]. These challenges are apparent with all sorts of projects with scattered team members, but since EA requires alignment between the views these challenges are more problematic. In addition to this there are conflicting interest among the stakeholders, 74% of the participants in a survey agree with the statement on this [20].

Stakeholder Communication

Communication of EA artifacts is complicated. First and far most because of diversity in backgrounds and the needs of the different stakeholders. Architects usually have a technical background and therefore are likely to lack the ability to speak the business language [25].

In a seminar workshop on top EA challenges [27] the communication of artifacts is considered to be a challenge experienced by practitioners. Imagining the suitable view, or artifact, for each of the stakeholders is a challenge.

Understanding Requirements

Unclear business goals is the second most agreed upon challenge in a survey on EA challenges [20]. As explained at stakeholder communication, people have different types of backgrounds which complicates the development of a mutual understanding. This challenge is not only the case between the business and architects, but also between the architects and developers [9].

Architects must comprehend two types of people and requirements. On one hand you have the business and on the other hand the technical infrastructure. Understanding both of these different areas proves to be difficult. The focus is usually too much on IT side, in a survey 67% agreed with the statement that EA team focuses primarily on IT [20].

Shared Understanding

Achieving a shared understanding of EA proves to be difficult for organizations. Driven by departments living in silos and resistance to change makes it difficult to find shared goals for EA [26]. In a survey under practitioners 84 % indicate that there is a unclear understanding of business goals and 75% indicate that the demands for the EA team are unclear [20].

Architect Experience

Another challenge mentioned in the literature, is the experience of enterprise architects, or more the lack of experience [9]. 87% of practitioners that participated in a survey agreed on the statement that it is hard to find experienced enterprise architects [20].

EA Frameworks

A broad selection of EA frameworks is available, e.g. Zachman framework for enterprise architectures, TOGAF and Federal Enterprise Architecture (FEA). The shortcomings of EA frameworks are considered to be a challenge.

The criticism on EA frameworks is on the complexity, insufficiently prescribing the process for generating EA artifacts and political, project and organizational challenges are not supported by the frameworks [9].

Knowledge Documentation & Presentation

Knowledge extends beyond the documentation of the models of the architecture. It includes documenting which choices were made in the process and why [22]. In ensuring good knowledge management, repository management is of importance [28]. Knowledge management is considered to be a challenge [9]. The reasoning behind decisions is not always documented [23].

The lack of documentation is not the only challenge within knowledge management. The opposite, overproduction of documentation, is also the case. The EA artifacts are often over-sized and too difficult [20]. Interesting to note is that there is a significant

difference between the US and Europe, only 6% of the organizations in the US experience this challenge whereas this is 35% in Europe [20].

Knowledge management can use improvement. For stakeholders it is important that they know what decisions were made, why the given decision has been made and what the impact of the decision is [29].

Tool Support

In the article by Lucke et al. [9] the lack of proper tool support is mentioned. The challenge is based upon an article by Kaisler et al. [29]. According to the article EA tooling is lacking the ability to create alignment between business operations and processes and is unable to support the diverse number of entities in an EA.

Both the paper from Lucke et al. [9] and Kaisler et al. [29] date back to more than five years ago. Since then tooling has progressed, and possibly addresses the problems described. It is unclear whether this is currently still a challenge. In the survey [20] that supports many of the problems already mentioned in this sub-chapter, this challenge is not mentioned by practitioners.

Architectural Scale

Architectural scale consists out of two different dimensions [9]: the scale of the size of the enterprise (e.g. an organization in the financial sector with hundreds of legacy systems) and the modeling of the various different views of the architecture for different stakeholders. Both dimensions reinforce each other, more systems results in more stakeholders which in turn results in more views.

Architectural Scope

When modeling an enterprise the scope must be considered. Modeling every aspect to the most detailed level is unlikely to serve any goal. Incorrect or insufficient scoping of the project can contribute to 'a never-ending series of analyses, analysis paralysis, and end up with nothing' [22].

Scoping relies upon knowing what the goal is beforehand. Making an informed decision requires you to have the necessary information at hand. This makes scoping complex, after the scope has been set, new information discovered later might change the situation.

In literature based on empirical data, this challenge is less apparent. Only 27% of the respondents agree with the statement that the right level of abstraction is not met.

Rapidly Changing Conditions

In Organizational Factors Influencing Enterprise Architecture Management Challenges [20] this challenge is most agreed upon, 71% of the respondents agree with the statement that the enterprise environment changes too quickly.

Software lifecycles are getting shorter [9]. This impacts EA development because results are expected earlier. A 'big bang' approach to EA considered to be impossible [22]. Dealing with rapidly changing conditions is considered to be a challenge.

Outdated Results

EA attempts to create alignment between the business goals and IT. In achieving alignment predefined processes exist, such as ADM, the output of these processes should result in IT projects or changes to the business.

Outdated results are present when transformations are initiated before the results of EA are available. Delivering results on time is considered to be a challenge in practice [30]. A survey on the use of agile in EA development also shows that this is a challenge [20], 38% of the organizations are struggling with outdated EA results.

3.2 Characteristics of Agile Development

This section describes characteristics of agile development first an explanation of the approach is given. After which the results are presented.

3.2.1 Approach

The same approach as prescribed in figure 3.1 is used. In this case with the following goals, criteria and keywords:

Goal: to identify relevant papers on agile methods in order to construct a list of characteristics of agile.

Criteria: considering the amount of results a number of strict criteria were applied (a search on the term *agile* and *characteristics* on Google Scholar results in a little over 160,000 results). The following criteria were followed in selecting relevant literature:

- Literature should originate from the computer science field. Although agile methods were first applied in computer science, agile methods have also been adopted in other fields, e.g. supply chain management. Literature from other fields is omitted. This literature is merely an application of that from the computer science field. This increases the feasibility by significantly limiting the amount of literature.
- Empirical studies are excluded. The goal is to deduce a list of characteristics. The application of agile in practice is not the focus, although such literature might give interesting insights it does not directly contribute to gathering characteristics of agile approaches.
- Literature preferably describes agile and not an implementation of agile, i.e. the unified process or SCRUM.
- Literature describing the importance and workings of agile can be used to base characteristics upon.

Keywords: the following search string was used for uncovering relevant literature: TITLE(agile) AND (characteristic OR characteristics OR practice OR practices).

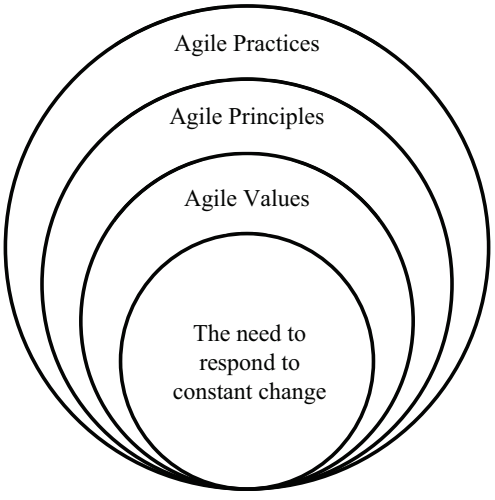

FIGURE 3.3: Break down from the practices, principles and values to handling change adapted from Becoming Agile in an Imperfect World [31]

In figure 3.3 a break down of concepts is shown, from the need to respond to constant change to the practices needed to achieve this. The figure moves from abstract concepts to more concrete concepts. A more concrete concept is useful in the case of this thesis. Whereas the value 'working software over comprehensive documentation' [15] is difficult

to measure, a characteristic such as focus on face-to-face communication can be perceived by individuals in an organization.

The literature revealed a number of papers that are used in determining the practices for achieving an agile way of working. A paper that proved most useful described a methodology for assessing agile software developments methods [32]. The dissertation [33] upon which the paper is based provided more background.

Agile development shares a common set of values, which are translated into characteristics. The values are derived from the agile manifesto [15]. The characteristics listed can be broken-down into practices. For each characteristic a number of practices are derived from the literature.

This break down is based upon a number of articles uncovered in the literature search. The types of articles found can be divided in two categories: articles describing agile maturity models and articles describing aspects of agile software processes. The following articles form the basis for the agile practices.

- A Methodology for assessing Agile Software Development Approaches [32]
- Light Maturity Models (LMM): An Agile Application. [34]
- Using factor analysis to generate clusters of agile practices (a guide for agile process improvement) [35]
- Driving process improvement via Comparative Agility assessment [36]
- The Characteristics of Agile Software Processes [37]
- Agile practices in global software engineering [38]

3.2.2 Results

Programming specific practices, e.g. pair programming and refactoring, are mentioned frequently due to the fact that agile development originates from the computer science field. These practices were disregarded since they do not support the goal of this sub-chapter. Each of the agile principles [15] are described below with the agile characteristics found in the literature.

Continuous Delivery of Valuable Software

The continuous delivery of valuable software is the first principles listed in the agile manifesto. This principle is presented as the 'highest priority'. Although the manifesto does not give the relationships between the principles, the continuous delivery of valuable software could be seen as the main principle which is supported by the other principles. E.g. the frequent delivery of working software and dealing with changing requirements help achieve the continuous delivery of valuable software. This distinction is also made in the paper of Soundararajan [33]. Continuous delivery is therefore not seen as a distinct principle, but rather as an overreaching one.

Dealing with Changing Requirements

Practice	Source
Adaptive	[37]
Convergent	[37]
Lightweight requirements	[35]
Product backlog	[32]

TABLE 3.1: Practices for dealing with changing requirements

The practices in table 3.1 enable dealing with changing requirements. On top of the given practices, incremental and iterative development also support this. Each iteration enables you to take on new requirements.

Not everything can be planned beforehand, the process should therefore be adaptive [37]. If during an iteration new requirements emerge these should be addressed by initiating new activities. Similar to this the process needs to be convergent [37]. Risks should be actively attacked.

There are also requirements to the requirements. Requirements should be lightweight [35] which is achieved by two things: (1) specification of requirements should be high-level and (2) use cases should be light. The product backlog is used to give an overview of what needs to be done [32].

Frequent Delivery of Working Software

The practices in table 3.2 ensure the frequent delivery of working software. Two practices reoccur frequently in the literature: iterative and incremental development. In

Practice	**Source**
Time-bound	[37]
Incremental and Iterative	[37] [35] [32]
Evolutionary requirements	[32]

TABLE 3.2: Practices for achieving continuous delivery of valuable software

Characteristics of Software Processes [37] these are respectively defined as follows: continually making short cycles to refine the deliverable and building small parts of the system instead of taking a holistic approach. This difference is illustrated in figure 3.4.

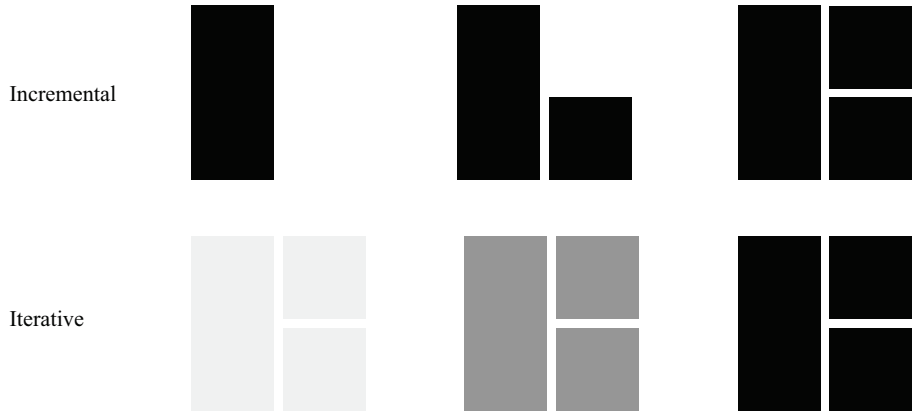

FIGURE 3.4: Difference between iterative and incremental

Evolutionary requirements are defined by four components [32]: (1) just-in-time establishment, (2) feature driven, (3) prioritization by the customer and (4) tools supporting the process. The idea behind this is that the system arrives at its final form by close customer involvement. Due to the involvement of the customer the system is more likely to be valuable to the customer. The time-bound component emphasizes two things [37]: (1) the iteration is done within a beforehand set short time period and (2) activities that cannot be completed within the iteration are dropped. This ensures continuous delivery of software, constant pace and due to the short time period enables you to deal with changing requirements.

Collaboration between Business and Developers

Practice	**Source**
Client-driven iteration	[32]
Collocated teams	[38]

TABLE 3.3: Practices for achieving collaboration between business and developers

In the agile manifesto this principle is illustrated with the example of buying a car. You specify what you want up front, discuss a price, and order the car. After a period of time you get exactly what you ordered. With software this is not the case. Specifying upfront what you want is unlikely to deliver the desired result.

Since specifying the requirements upfront is difficult, collaboration between business and developers is needed to move towards more low-level requirements. This can be done with client-driven iterations [32]. The client, or customer, decides which features will be dealt with in the next cycle. Collocated teams work at the same location, which makes communication easier.

Create Trust and Motivated Individuals

Practice	**Source**
Team composition	[36]
Appropriate distribution of expertise	[32]
Title and salary alignment	[39]
Empower the individual	[33]

Table 3.4: Practices for creating trust and motivated individuals

Agile development relies on individuals 'who make the difference'. Creating trust between team members and having motivated individuals is therefor of most importance, it enables teams to develop software incremental and in parallel [37].

By creating emphasize on the individual by taking decision-making to the lowest level [33], choosing people with the right expertise [36] [32] and equal rewards should help in creating trust and motivated individuals [39].

Rely on Face-to-Face Communication

Practice	**Source**
Regular stakeholder meetings	[33] [38]

Table 3.5: Practices for ensuring face-to-face communication

Agile development relies on face-to-face communication. This has two reasons: on one hand the goal is to limit the amount of documentation . If documentation is kept to a minimum, communication must be done via some other medium.

Face-to-face communication should not be limited to the development team. Communication with stakeholders should also be done face-to-face [38]. Practices identified generally boil down to having frequent face-to-face meetings. In a popular implementation of agile, SCRUM, this is achieved by daily stand-up meetings.

This principle has only one practice supporting it. This is because it is tightly linked with collaboration between business and developers and it is straightforward.

Working Software as Measure of Progress

Practice	**Source**
Daily progress tracking meetings	[33]
Iteration progress tracking and reporting	[33]

TABLE 3.6: Practices for achieving working software as measure of progress

In a waterfall approach implementation and delivery are done at the end of the project. This introduces a risk, only at the end of the project will be discovered whether the software meets the expectations of the customer. With an agile approach frequent delivery of working software is the goal. This overcomes the risk that late in the project unexpected problems are discovered. If a partial solution does not work properly this is uncovered much earlier on.

Two practices were identified in the literature that support this principle. Each iteration should deliver a working piece of software, tracking the progress of the whole iteration is therefore important. Tracking on a lower level is also desirable, this is achieved by daily progress tracking meetings [33].

Maintain a Constant Pace

The constant pace is about sustaining a steady production of deliverables. In software development this would be completing a constant amount of user stories. It is not the

Practice	Source
Sustainable pace	[39]
Agile project estimation	[32]
Response to stress	[39]

Table 3.7: Practices for maintaining a constant pace

purpose to make overtime to finish an iteration. If that is the case the workload is too high and the amount of work done in an iteration should be scaled back.

Three practices were identified in the literature that support maintaining a constant pace. The first practice is sustainable pace [39]. Agile project estimation [32] incorporates the customer into the project estimation and response to stress when it is signaled [39].

Technical Excellence and Good Design

Practice	Source
Just-in-time	[32]

Table 3.8: Practices for achieving technical excellence and good design

Technical excellence is achieved by selecting the right processes, right people and right tools. In the articles found practices for achieving technical excellence and good design are mostly related to coding practices. Practices that pertain to technical excellence and good design are less applicable in the context of this thesis.

Just-in-time planning is planning as little and late as possible. Agile methodologies endorse the fact that you cannot predict the future. Planning should be done as late as possible and on a continuous basis.

Focus Only on the Essential

Practice	Source
Agile documentation	[32]
Parsimony	[37]

Table 3.9: Practices for ensuring focus on only the essential

Delivering working software is of paramount importance for agile approaches. Documentation needs close consideration. Waterfall approaches require the production of formal

documentation to finalize phases and communicate with stakeholders. Agile documentation consists out of tools such as user stories to document requirements, and tooling for developers to create documentation [32].

In a paper on agile characteristics the concept parsimony is introduced [37]. This is the practice of minimizing the risk and number of resources needed to a achieve a certain goal.

Self-Organizing Team

Practice	**Source**
Collective code ownership	[36]
Self-managing	[33]

TABLE 3.10: Practices for achieving self-organizing teams

Self-organizing teams are teams which are not governed by management, but by the individuals themselves [33]. Management still selects the team members and might exercise some influence when problems occur, but there are no formal roles or responsibilities defined.

Individuals assign themselves work, and take ownership for their work. This is referred to as collective code ownership.

Reflection on the Process

Practice	**Source**
Continuous feedback	[32]
Retrospective meeting	[32]
Team-learning	[39]

TABLE 3.11: Practices for ensuring reflection on the process

As already shown agile puts emphasize on the soft side of the process. Reflection on the process is also a principle of an agile approach. Reflection is done within the team by retrospective meetings [32] which are held after each iteration and focus on possible improvements to become more effective. Also stakeholders partake in the process by continuous feedback [32]. There is more to the process than just deliverables, evaluating the process itself is just as important.

3.3 Transition from Waterfall to Agile

This section describes the underlying reasons for transitioning from a waterfall to agile development. The approach to the literature review will be explained first after which the results are described.

3.3.1 Approach

The approach from figure 3.1 is followed again with the following goal, criteria and keywords:

Goal: find literature that describes or identifies the drivers behind the transition from waterfall methodologies to agile methodologies.

Criteria: in selecting the literature two categories of literature are of interest (in order of preference).

1. Preferably the drivers behind the transition from the waterfall methodology to agile methodologies are given.
2. Comparison between different software development approaches, from these characteristics the drivers could be deducted.

Keywords: the following search string was used for uncovering relevant literature. The keyword search was limited to the title and abstract. These keywords do not specifically focus on drivers or transitions but making the keywords to specific proved to deliver little relevant articles: agile AND (waterfall OR traditional) AND (software development)

Getting consistent results, i.e. a search that delivered results that were relatively useful, was difficult on both Scopus and Google Scholar, variations on the keywords had a very slight impact on the results. Literature tends to focus on factors that impact the acceptance of agile methodologies, e.g. perceived benefits, training and technical factors, and not on the fit or drivers for applying agile to projects.

3.3.2 Results

The literature search revealed that no straightforward answer to the third sub-question, does EA benefit from agile approaches, exists. The next sub-chapter shows that although agile practices have been applied to EA, this very fundamental question is left unanswered. Before measuring whether agile practices benefit EA, a basis for this assumption is formed. This is done by verifying the following three preconditions (1) examining the underlying drivers for a transition from waterfall to agile, (2) showing that EA follows a waterfall approach and (3) verifying that the drivers are also applicable in the context of EA.

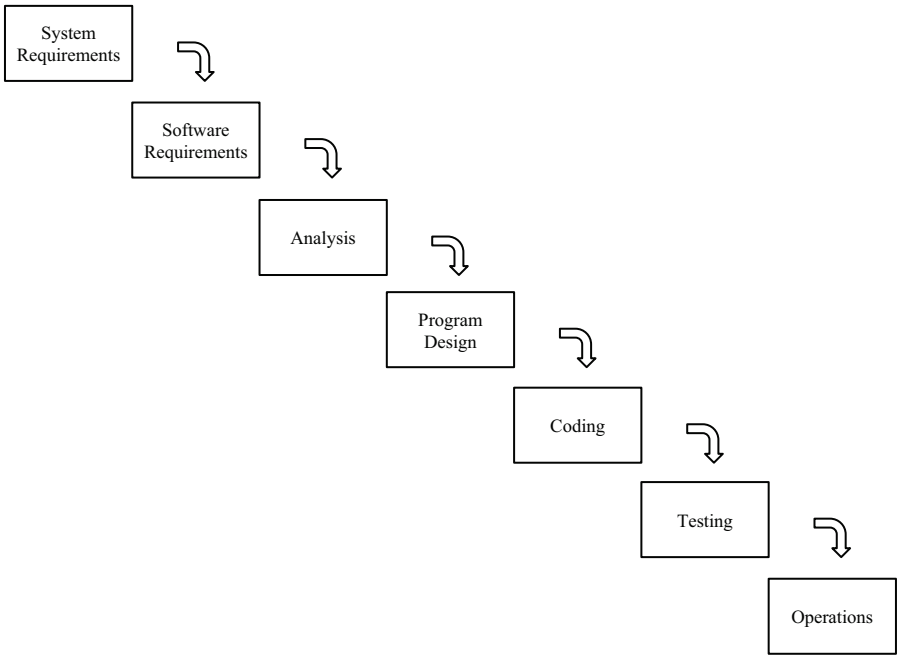

FIGURE 3.5: Implementation steps of waterfall method as envisioned by Royce adapted from Managing the Development of Large Software Systems [40]

In software development a common practice to managing a project is the waterfall method. This method was envisioned by Winston Royce in 1970 [40] which is shown in figure 3.5, it was derived from the area of systems engineering. The waterfall method is a sequential approach in which a number of phases are executed. Advancing to the next phases requires you to have fully completed and signed-off the current phase. A common set of phases that are used today in software development are: requirements gathering, design, implementation and testing.

Waterfall development has its shortcomings. These included, but are not limited to the following taken from Software engineering [14] and relate to properties of the approach itself. Other common heard disadvantages are project overruns and project failures.

1. Inflexibility due to the distinct separate phases.
2. Only suitable when requirements are well understood beforehand.
3. Possible premature freezing of requirements. Changing afterwards is costly.
4. Errors and need for extra functionality identified late in the process.

In software development different types of development approaches exist. Whereas the waterfall approach was first most used, a transition is taking place towards more agile approaches. This transition is fueled by the underlying assumptions or conditions of the waterfall approach not being suitable anymore.

The goal of this sub-chapter is to specify the conditions underlying a choice for a certain software development approach for a project. The types of software projects can then be compared to EA development to determine which underlying conditions are also applicable to EA development. This comparison forms the basis for showing that EA could benefit from agile methodologies. This lays the groundwork for the further research into applying agile in EA development.

There are different types of approaches for developing software. Different types of software projects require different types of approaches [41]. The matrix in the figure 3.6 defines the type of software projects on two scales: clarity of the goal of the project and the clarity on the solution the project should deliver. Each of the quadrants requires a suitable software development approach.

In agile project management agilism versus traditional approaches [42] software development approaches are mapped to the four quadrants.

In quadrant 1 both the goal and solution are clear beforehand. Both the scope and requirements are therefore known. This makes a linear approach best suited for this quadrant. Linear approaches include the waterfall and incremental methodologies. Main difference between incremental and waterfall is that an incremental approach delivers

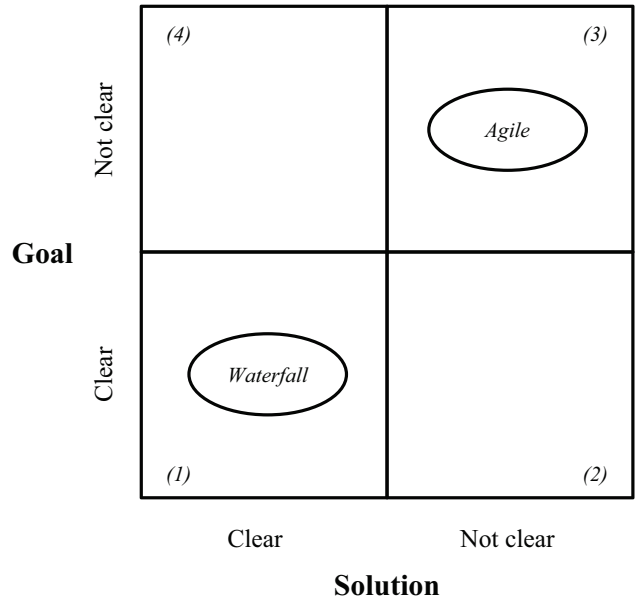

FIGURE 3.6: Different types of software projects adapted from Effective Software Project Management [41]

business value early on in the process. Both methodologies rely heavily on documentation and strictly predefined processes. Because linear methodologies presume a clear goal and solution, they cannot accommodate change in the deliverable or process.

Thus the conditions underlying linear methodologies do not make is suitable for change. They presume environments with little uncertainty.

In quadrant 2 the goal is clear but the solution is not. This means that the requirements are subject to change. It is clear where the project is headed, but it is still a discussion on which route to take. The agile methodologies are best suited for this quadrant. Agile approaches can be iterative, adaptive or both. The common denominator between these approaches is that they deliver intermediate solutions. The intermediate solutions act as navigator, the details of the solution are discovered through each iteration. This does require an active customer, which also approves of the lack of clarity of the final solution.

In quadrant 3 both the goal, and (partially) the solution are not clear. This results in a situation in which during the project the goal and solution become more detailed. As with second quadrant, agile methodologies are best suited here. Instead of an iterative approach, an adaptive approach is followed. With each iteration the solution and goal become clearer. Disadvantages of the adaptive approach are that customer involvement is required and the exact details of the final deliverable are unknown.

The transition from waterfall methodologies towards more agile methodologies has just been explained by defining the type of software project on two aspects: clarity of goal and solution.

The question remains in which quadrant EA should fit. If we take a look at the challenges that were uncovered in the previous sub-chapter, and then specifically those of Hauder et al. [20] a number of challenges are related to the goal or solution: ad hoc EAM demands, unclear business goals and the enterprise environment changing too quickly. These challenges strongly indicate that at least the goal is unclear and a changing environment might also impact the solution. This would position EA in quadrant 3 or 4. This does not answer the complete question since it is unknown in which quadrant EA is currently placed. The next section elaborates upon this question.

Relating Waterfall to EA Development

Different frameworks for EA exist. Determining whether EA follows a waterfall approach requires a closer look at these frameworks. Analyzing all existing frameworks would be too time consuming, instead one of the most used frameworks, TOGAF, will be taken as starting point. TOGAF has specified ADM as the method for EA development.

By highlighting a number of aspects from ADM. It is shown that the approach followed resembles waterfall. The following quote taken from documentation from The Open Group on TOGAF states almost the exact opposite of what is argued here.

> "The graphical representation of the TOGAF ADM .... and the description of the ADM phases discretely in order in Part II, can be read to imply a deterministic waterfall methodology. This method of presentation is provided for the purpose of quickly communicating the basics of architecture development and the architecture lifecycle. In practice, two key concepts are used to manage the complexity of developing an enterprise architecture and managing its lifecycle - iteration and levels."

The documentation also proposes a number of phases which can be iterated over. These predefined iterations are given in figure 3.7. Based on four distinct aspects it is shown that TOGAF ADM adheres an approach which shows close resemblance to waterfall.

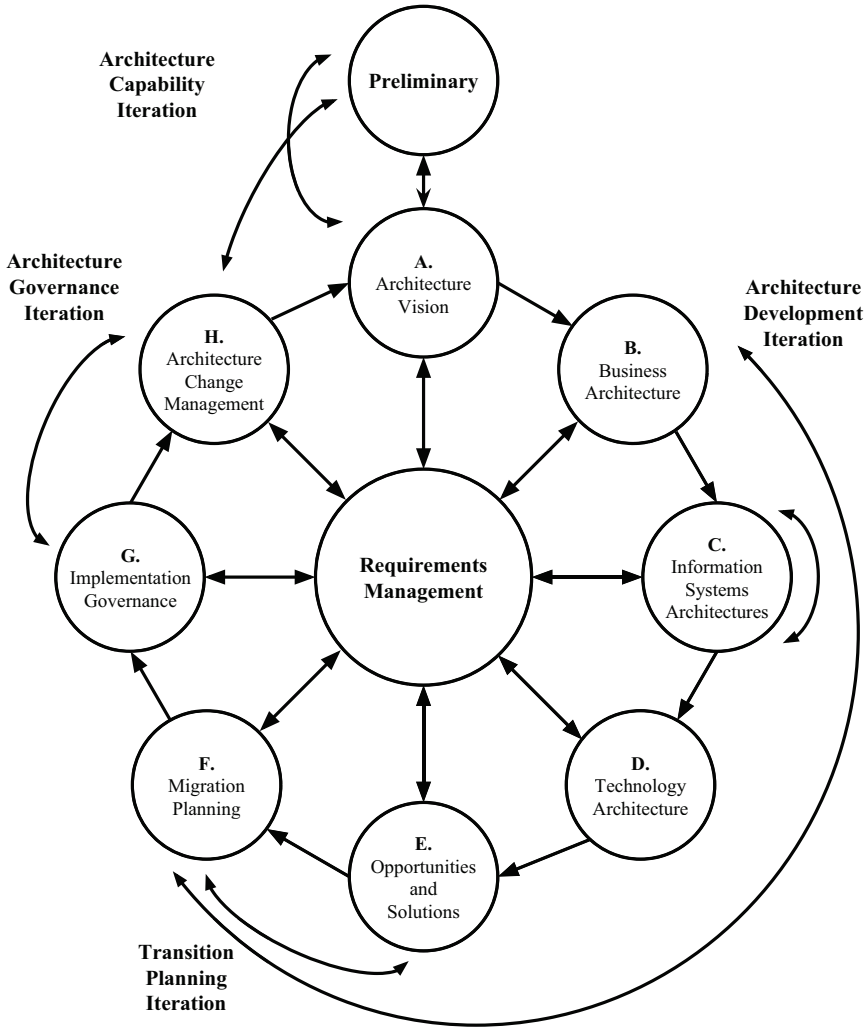

FIGURE 3.7: Iterations in TOGAF ADM

Goals not clear

In the preliminary phase requirements for the architecture work are set. In next phase, Phase A: Architecture Vision, the scope for the baseline architecture is determined. The scope is set early on in the process, and setting the scope requires a clear goal. Requirements management is not specified in detail, but during the ADM cycle the phases are validated against the requirements. Defining requirements upfront presumes some clarity on the solution.

In figure 3.6 the ADM cycle is positioned more towards the first than the third quadrant. By defining the scope and setting requirements at the beginning of the process it is assumed that there is already certainty on what is to be delivered. This fits with a waterfall approach which also starts by an analysis to define what will be delivered.

Iterative does not equal agile

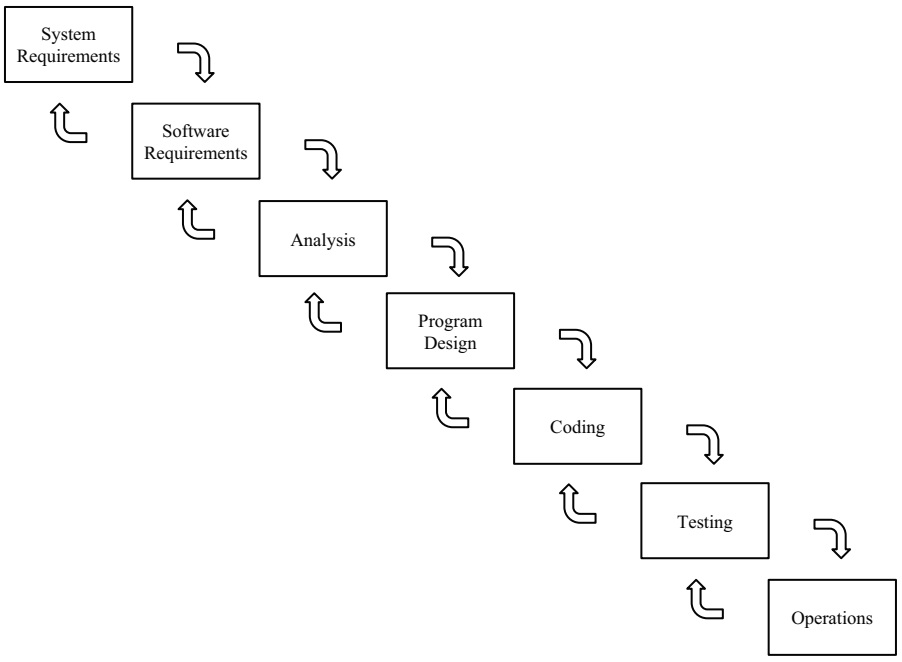

FIGURE 3.8: Iterative waterfall method as envisioned by Royce adapted from Managing the Development of Large Software Systems [40]

The phases from the ADM cycle can be executed in an iterative fashion, a number of predefined iterative cycles are given in the documentation. Due to these iterative cycles it is not a deterministic waterfall methodology according to the documentation. Iterative is not incremental, agile is a combination of both.

Using iterations makes it non-deterministic, but this does not change the fact that it is a waterfall approach. In the original paper from 1970 by Royce [40] iterations are already proposed as shown in figure 3.8, and are inherently part of the waterfall methodology.

Documentation driven

Another aspect of a waterfall approach is the extensive documentation. Each phase delivers documentation which is formally approved. Once approved the next phase is initiated. ADM follows a similar pattern.

TOGAF also relies on formal documentation. To ensure stakeholder involvement the finalization of most phases requires the conduction a formal stakeholder review. This consist out of presenting the given artifact to the stakeholder and ensuring that it aligns with their requirements.

Waterfall challenges

Besides looking at what is specified in the documentation, the EA challenges give an indication of the approach followed. This has the advantage of taking a broader view since it does not look at one particular framework. On the other hand it is difficult to substantiate that the challenges are the cause of the underlying project approach.

A number of problems identified in the literature are most likely the outcome of the underlying project management approach. Each of these issues or challenges are commented on below, with an explanation as to why these are the outcome of the project management approach.

A challenge identified is determining the architectural scope. The project management approach determines how to deal with the scope: with a waterfall approach scope is determined at the start of the project, this is possible since both the goal and solution are clear at that point. With an agile approach the scope can be slightly adopted per sprint or iteration. Over-scoping, attempting to model to much detail, is a common problem in EA development. The TOGAF ADM enforces a team to determine the scope beforehand, this is likely contributing to the difficulties surrounding the architectural scope.

Dealing with rapidly changing conditions is also considered to be a challenge. The formal and documentation heavy approach of TOGAF ADM makes it difficult to quickly cycle through the phases. This certainly does not help in accommodating rapidly changing conditions. Agile methodologies are focused on continuous delivery of valuable software. This could offer a solution to these rapid changing conditions.

Both stakeholder coordination and communication are also challenges that are mentioned. These could be the outcome of stakeholders not realizing the value of EA or having different goals that cause the lack of cooperation. On the other hand the development approach usually prescribe how to approach stakeholder management, these problems could very well be the result of the project approach.

Considering the three preconditions stated at the beginning of this sub-chapter: (1) examining the underlying drivers for a transition from waterfall to agile, (2) showing that EA follows a waterfall approach and (3) verifying that the drivers also applicable in the context of EA. As shown these preconditions are met. This is not an exact science

but it gives a strong indication that applying agile approaches to EA could be useful. Which is also emphasized by existing academic literature on the subject as shown in the next sub-chapter.

3.4 Current State of Agile in Enterprise Architecture

The section follows a similar structure. First the approach is explained after which different agile EA frameworks are discussed.

3.4.1 Approach

The approach described in figure 3.1 is followed again with the following goal, criteria and keywords.

Goal: identify articles which describe the application of agile to EA.

Criteria: articles should focus on the application of agile to EA and not introducing agility to the architecture (for example by introducing services).

Keywords: TITLE(agile AND (enterprise architecture))

Research on the application of agile within EA in scientific literature is limited. A search on Google Scholar delivered only 48 results. Of these results a significant amount is written by practitioners employed by (software) companies or offer little concrete insight into the adaption of agile techniques in EA.

A similar observation was also made in one of the articles [43] that dates from 2015: 'Although professionals are already trying to combine both Agile Software Development (ASD) and Enterprise Architecture Management (EAM), this combination has to the authors' knowledge been barely researched so far'.

Research on the application of agile principles in practice has been conducted [16]. This was done with the use of surveys to create an overview of which principles were applied to EAM. The article does not specify how the organizations implemented the principles.

3.4.2 Results

A selection of papers that offer concrete insights, i.e. propose or demonstrate the adoption of an agile approach in EA, are elaborated upon below. Each section describes an approach or framework from an article.

Integrating ASD and EAM

In the article Integrating Agile Software Development and Enterprise Architecture Man-

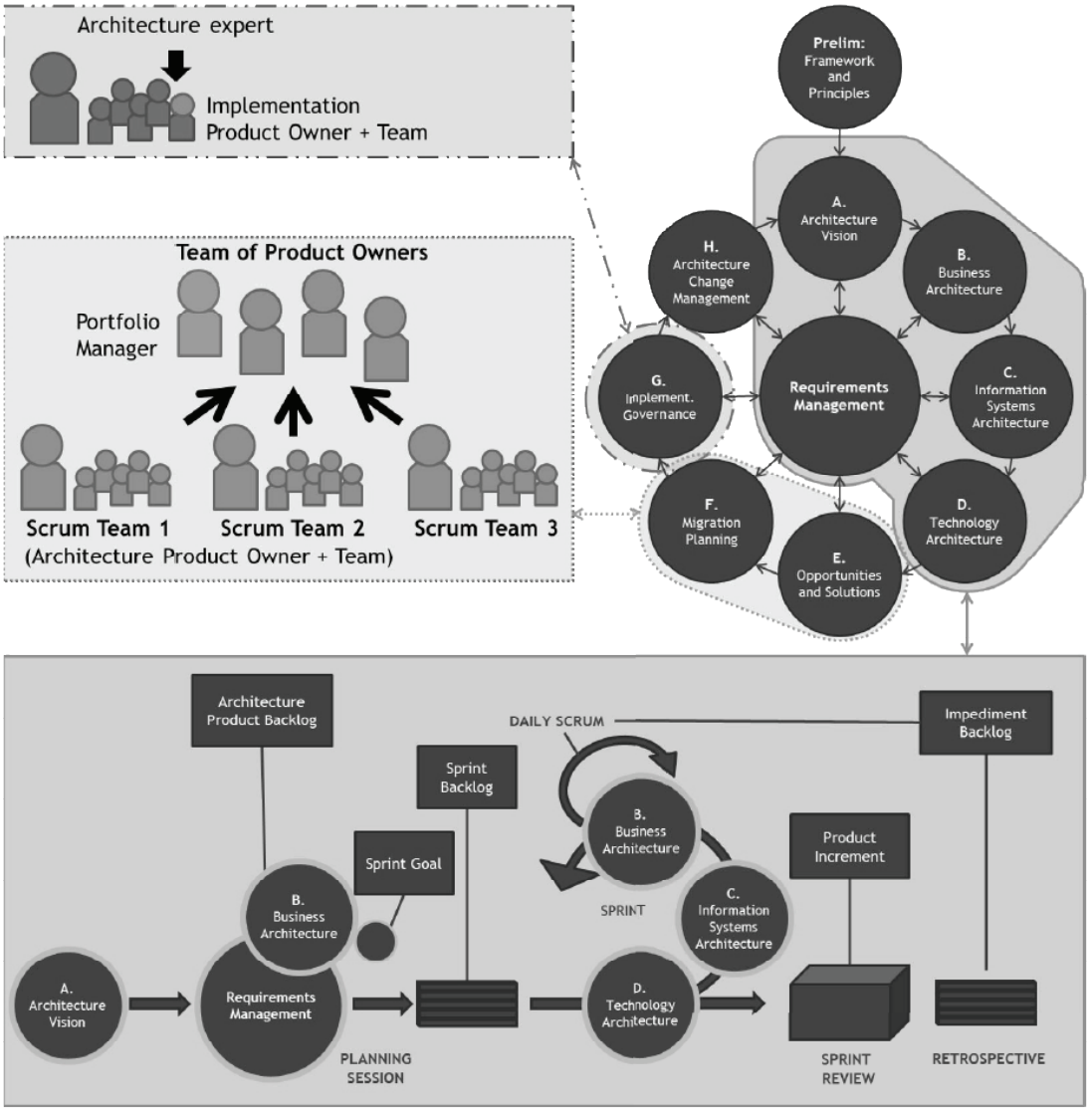

FIGURE 3.9: Integration of ASD and EAM [43]

agement [43] an adaption agile adoption of TOGAF ADM is realized by incorporating

SCRUM. In their model the architecture vision, business architecture, information system architecture and technology architecture are developed in sprints. The complete TOGAF ADM is done in four SCRUMs. This model is given in figure 3.9.

The framework was developed using a design science research approach. The solution had the objective of addressing identified problems: the ivory tower syndrome and the production of unneeded documentation. Issues such as architectural scope, rapidly changing conditions or outdated results were not considered.

The objectives of the solution lack a number of EA challenges. In the preliminary phase of TOGAF ADM the scope is already determined, although sprints are used during ADM, flexibility is lost due to the fixed scope. Dealing with rapidly changing conditions becomes difficult.

The demonstration & evaluation phase of the artifact is limited. The framework was presented to experts who gave their opinion. The experts found the idea of applying SCRUM promising next to that the idea of applying other agile methods is suggested.

Suggested is that EA teams should trigger implementation projects. No comments were documented about the lack of key points suggested in the agile manifesto, e.g. the ability to deal with changing requirements. One of the objectives, the creation of unneeded documentation is not addressed, but might be indirectly addressed by stand-up meetings which should bring down the need for documentation.

This article is not the first attempt to combine EAM and SCRUM, this was already done in 2011 [44]. That article does not propose an adaption of ADM, but does define roles and responsibilities in line with those from SCRUM.

The article proposes a number of interesting propositions, but leaves a number of gaps unanswered. The article leaves room for further research.

Iterative Approach to the EAM function

In figure 3.10 an other approach to EAM is given. This approach is classified as iterative, thus not as agile which indicates that it is not incremental. The model visualizes the interaction between two aspects: practice and research.

The model has an emphasis on stakeholders and the communication of artifacts. The model iterates over three steps: data gathering, stakeholder involvement and reflection.

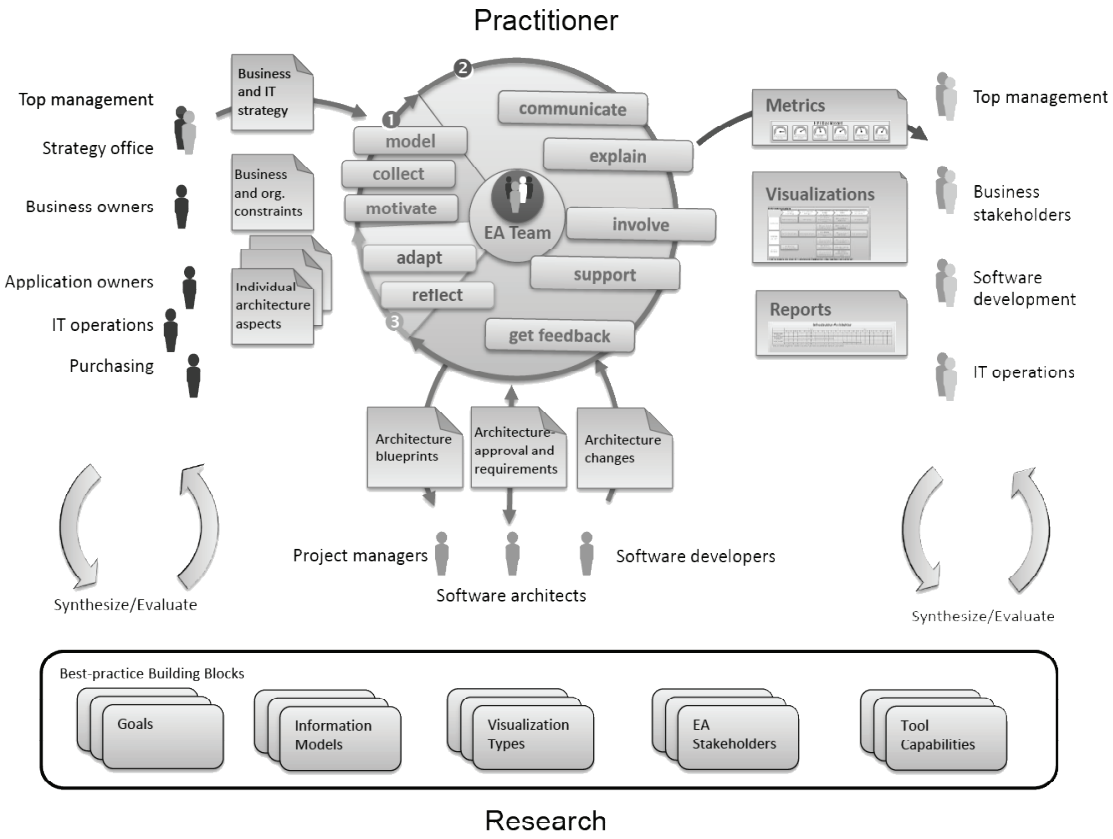

FIGURE 3.10: Iterative Process of the EAM function [45]

This is in contrast to TOGAF ADM were there is less focus on stakeholder management. Stakeholder management is part of each cycle in ADM and contains identifying their key concerns and position of power, but this is in contrast to the model given in figure 3.10. There is a strong emphasize on reflection, which is an inherent part of ASD.

The article from which the model is taken focuses on describing existing approaches for visualizing EA and tool support, not on the application of agile methodologies on EAP. No theoretical basis or explanation of development of the model is given, only an explanation of the model itself.

The article also details a number of issues brought forward by practitioners. The model is not directly related to the issues but in resolving certain issues early stakeholder involvement is desired [45]. This is reflected in their model and is also one of the practices from ASD.

Unfortunately the article does not further elaborate upon the design decisions regarding the model. It does however offer interesting insights that could possibly be used for the application of agile methodologies in EAP.

Process for Agile Enterprise Development

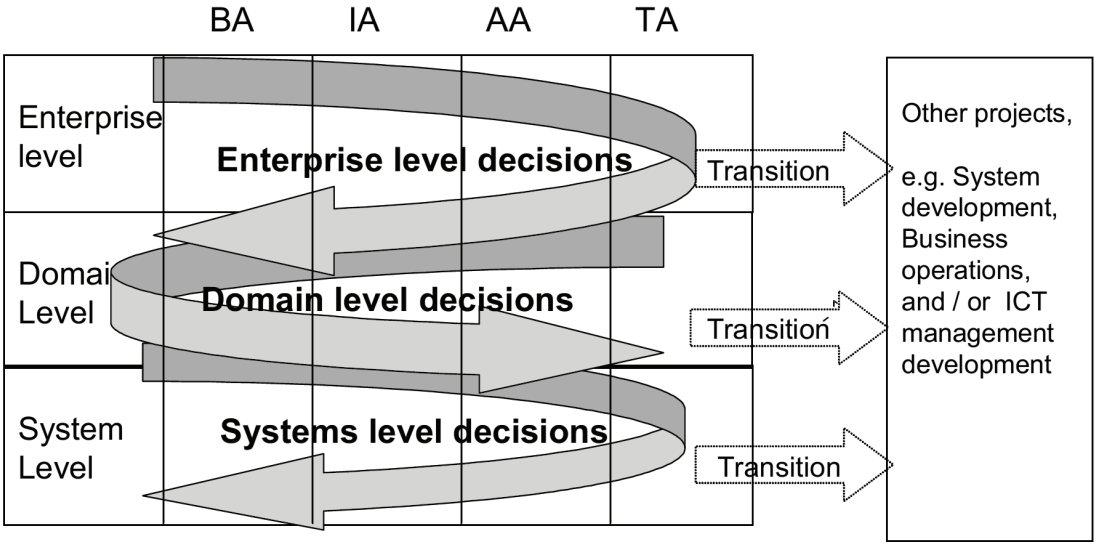

FIGURE 3.11: EA Management approach which supports a changing scope [46]

In EA Planning, Development and Management Process for Agile Enterprise Development [46] a process model is proposed that enables incremental development of EA. The model shown in figure 3.11 is suitable for for EA projects that have a limited time and scope.

This article proposes a process that should be capable of coping with a changing scope: 'The EA planning and development Grid has the basic idea that at each of the levels, decisions are made with a scope wider than the one(s) underneath, thus forming a continuum of decisions that become more and more focused, concrete and detailed' [46].

The article proposes moving through four types of architectures (business, information, systems and technology) on three different level (enterprise, domain and system level) as a tool of adapting and changing the scope. This approach does show resemblance to the leveled technique of TOGAF ADM.

A similar approach is proposed by Britton & Bye [47]. They propose levels of design. This is illustrated with the design of an airplane. First design the high-level structure, after which components of the airplane can be developed independently. The same reasoning can be applied to architecture.

Scaled Agile Framework (SAFe)

The SAFe [48] adapts various techniques from different agile approaches and scales these

up to the enterprise. The approach shows resemblance with SCRUM. The framework proposes an end-to-end agile approach to the enterprise. A part of the approach concerns EA development, which will be described here.

The framework acknowledges that some overview of the complete landscape must be present in the organization. But small agile teams are not capable of doing so: it is outside their environment and it is not their responsibility to understand the whole picture. Applying emergent design to architecture, only focusing on functionality needed for the next increment, is not feasible.

As a solution they propose the idea of intentional architecture. Intentional architecture is expressed in the architectural runway. The architectural runway represents the technological capabilities an organization has. The idea behind it is that the runway is continuously extended by introducing new technological capabilities that are required for features that will be implemented in the near future.

> **Intentional architecture** - a set of purposeful, planned architectural initiatives to enhance solution design, performance and usability and which provides guidance for inter-team design and implementation synchronization. [48]

By using the architectural runway SAFe attempts to combine enterprise architecture with emergent design. The approach is definitely not a big bang approach but could be described as a just-in-time way of developing architecture.

From the documentation available it is not entirely clear how these principles should be applied in practice.

Chapter 4

Conceptual Model Development

This chapter describes the development of the conceptual model. The model is described by defining the structural model and the measurement model, the former specifies the relationships between the variables and the latter describes how the items, i.e. questions on the survey, represent the variables. The sub-chapter on scale development describes how the survey items have been formulated and validated.

The main objective here is to provide an answer to the third sub-question: Does EA benefit from agile approaches? The literature shows, to a certain extent, that agile EA approaches have been researched and some theoretical basis exists for the application of agile approaches. One could say that other researchers consider it to be an area that offers potential. The effectiveness of such approaches has not been considered. Presumed is that an agile approach addresses EA challenges, but this relation has not been validated. With the use of surveys the relationship is validated.

An explicit choice was made to validate this assumption with the use of surveys and not with other methods, e.g. case studies or interviews. Organizations that explicitly state that an agile approach to EA development is used are limited. Conducting interviews or case studies is therefore difficult. By conducting surveys this is not necessarily an issue: distinct agile practices can be identified, without the organization explicitly following an agile approach.

Besides validating this relationship the data gives insights into which underlying agile practices and EA challenges are significant.

4.1 Structural Model

The survey measures two constructs: degree of agile practices applied and the perceived EA challenges. A description and the nature of each construct are given below.

- Degree of agile practices applied: this construct measures the extent to which an organization applies agile practices to the development of their EA. The entity the construct describes is the organization and the general property is the agile practices applied to EA development.
- Perceived EA challenges: this construct measures which EA challenges an organization perceives. The entity the construct describes is again the organization and its general property is the perceived problems regarding EA.

The relationship between the two constructs is researched with the use of a survey. Both constructs cannot be measured directly, these are also referred to as latent variables. Measures that have a relationship to these constructs should be used in order to measure these constructs. The results of the first two sub-questions are of use in determining indicators for these constructs.

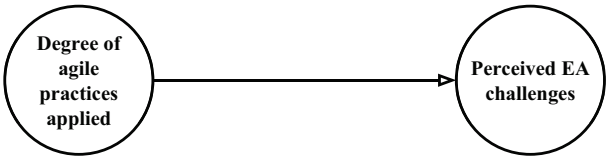

FIGURE 4.1: Structural model for interviews

In figure 4.1 the structural model, which shows the relationship between the latent variables, is given. The model contains only two variables. The model is kept simple for two reasons: (1) research agile EA is lacking and (2) the time available for this research limits what is possible.

The relationship represents the impact of agile practices applied to EA on the problems perceived in the EA organization. A higher degree of applied practices will result in fewer problems perceived, or at least that is expected.

It is difficult to add more constructs to the structural model since current research on agile EA development is limited. Basis for adding more constructs is limited although

adding certain constructs does seem like a logical step. Possibilities that could be considered are:

- Next to solving existing EA challenges, agile should introduce new capabilities. Adding the construct EA performance or capabilities seems logical.
- The impact of agile on the EA challenges is very likely influenced by the EA tooling/methods used and the maturity of the EA organization.
- Applying an agile approach will not only impact EA performance, but is likely to impact overall business performance.

An explicit choice has been made not to incorporate these constructs in the structural model. No literature was found supporting the above claims. Beginning with the most basic structural model is the most logical starting point.

4.1.1 Reflective or Formative

Before continuing with defining the measurement model it is necessary to explain the difference between to types of constructs: reflective and formative [49]. In research reflective constructs are mostly used, but the conceptual model that is shown here uses formative constructs, which require explanation.

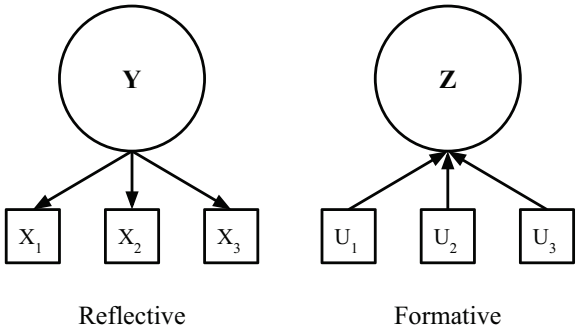

FIGURE 4.2: Difference between a formative and reflective construct

The difference between the constructs is given in Figure 4.2. With a formative construct, correlation between the indicators is not to be expected. Thus a change in U_1 does not result in a change in U_2. A commonly used example is the socioeconomic status (SES). Consider the construct SES has education and neighborhood as indicators. When a

person graduates his SES increases, but the neighborhood does not change, in this case there is no direct correlation between the indicators and the construct does not have to impact all the indicators.

With a reflective indicator there is a correlation between the indicators and the construct directly influences the indicators. A change in Y will result in a change in $X_{1..3}$. Reflective indicators are mostly used in research, although this choice is not always correct [50].

The measurement model specifies the indicators. In the case of both agile practices and EA challenges the indicators are formative. Both define and describe the construct. Correlation between the indicators is likely to be low. Examples of indicators of agile characteristics are the dependence on face-to-face communication and the frequent delivery of working software. An organization that relies for the most on face-to-face communication does not automatically realize frequent delivery of working software.

The same is true for the EA challenges. Examples of EA challenges are the lack of experienced architects and stakeholder communication. Although organizations might be facing both challenges simultaneously, they can exist independent of one another. An increased perception of one of the challenges does not automatically lead to an increased perception of the other challenge.

It seems to be a better idea to use reflective measures because these are less problematic compared to formative measures. An explicit choice was made to use formative measures. Reflective measures would have been possible if we are only interested in the impact of agile on EA development, and not interested in which underlying factors are significant in this relationship.

4.2 Measurement Model

In figure 4.3 the components used for the measurement model are given.

The measurement model describes how the items on the survey represent the variables in the structural model. Each section describes one of the first-order constructs. The items are grouped by indicators which have been derived from the literature. The numbers in

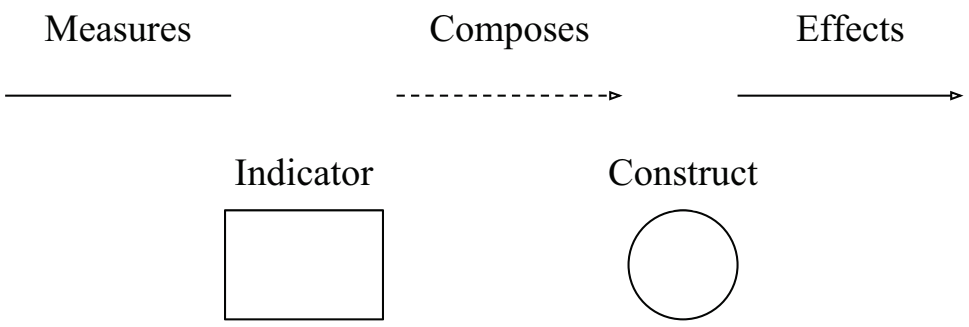

FIGURE 4.3: Components used for the measurement model

the diagrams refer to the questions on the survey. The development of the actual items is explained in the sub-chapter on scale development.

Figure 4.4 contains the measurement model that has been developed by just taking the results of the literature review. All practices and challenges are represented as individual indicators. The model contains 26 indicators, measuring each indicator will require a number of questions in the interview. To ensure that participants will complete the survey the amount of questions must be kept to a reasonable amount. The target is to use two to three questions per indicator.

The initial model needed reconsideration. Having this amount of individual indicators in a measurement model is problematic. Figure 4.5 contains the revised measurement model. The model in figure 4.4 needed adjustment because the indicators contained redundancies and conceptual overlap. Both could later form problems at the analysis of the data. Conceptual overlap can cause multicollinearity among the indicators [51].

Another driver is reducing the total amount of indicators. Seven indicators for one construct is already considered to be a large amount [51]. Having more indicators has the risk of indicators being non-significant when they are potentially not. In the case of figure 4.4 this could result in indicators not having any impact on the constructs, while this is not necessarily the case. The model has become complex. The model now contains first and second-order latent variables.

Grouping of indicators was done based on conceptual overlap. Expected is that the grouped indicators will correlate. In the new model all the indicators are still traceable,

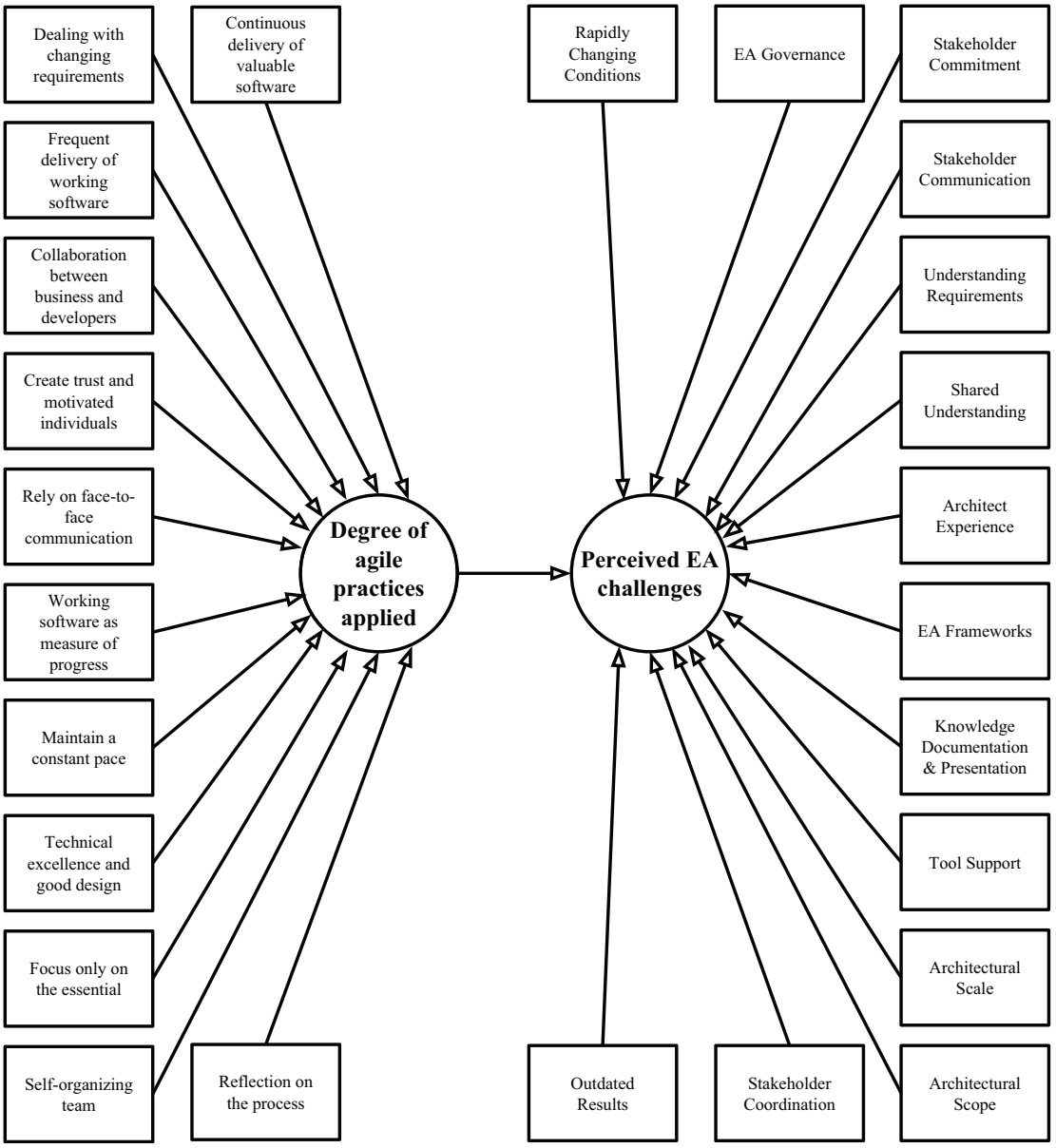

FIGURE 4.4: Measurement model for interviews

i.e. indicators from the initial model are still reducible from the items. Analysis of the data will show whether the grouping was actually grounded.

The grouping of the indicators for the degree of agile practices applied was done based upon the values from the agile manifesto [15] Continuous delivery in figure 4.5 aligns with responding to change. Collaboration aligns with customer collaboration and focus on individuals & interactions. Parsimony aligns with working software over comprehensive documentation.

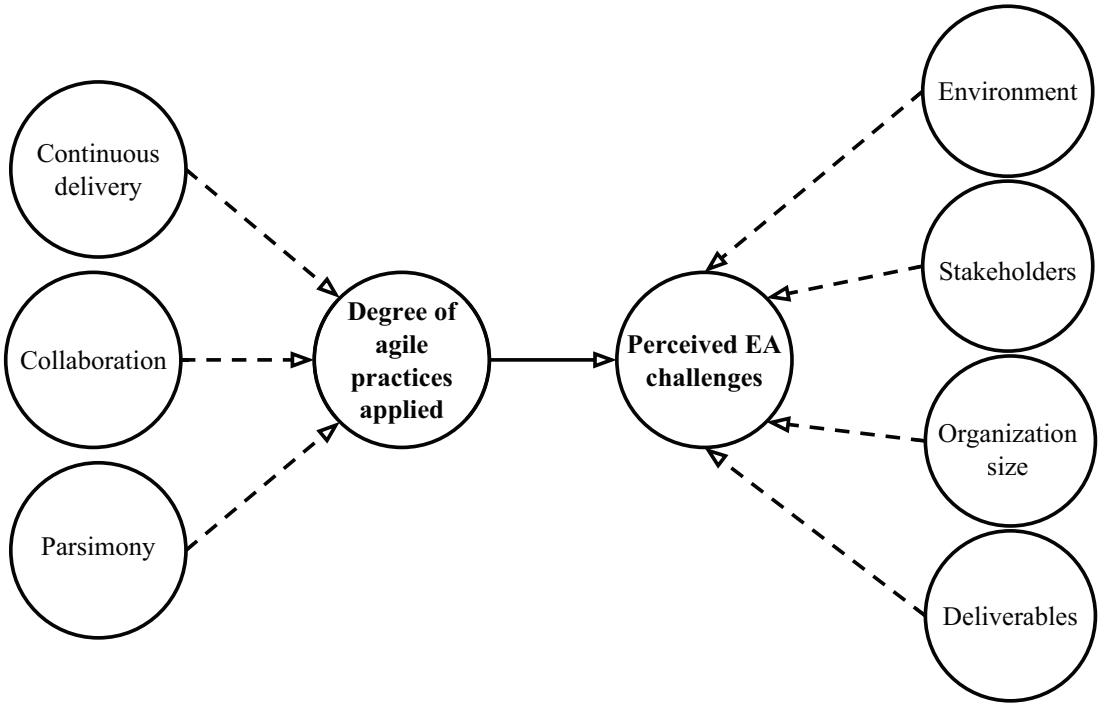

FIGURE 4.5: Revised measurement model for interviews

The grouping of the indicators for perceived EA challenges mostly follows the categorization of Lucke et al. [9]. Stakeholders aligns with people and semantics. Except EA governance but was placed under environment because it pertains to stakeholders. Environments aligns with Dynamics and Organizations size with extent. Deliverables focuses on semantics.

A number of indicators have been removed, this decision was made because of two reasons: (1) strong indication that these indicators will be non-significant and (2) reducing the amount of indicators is preferable, as opposed to reflective were the amount of indicators has little effect. For formative there is a limit in order to retain significant weights [51].

Two indicators from the agile construct were removed: technical excellence & good design, and motivated individuals. The first indicator is excluded because it is closely related to software design, and is therefor more difficult to generalize to EA development. The second indicator is removed due to ambiguity. Motivation is a complex phenomenon, although it is an important cornerstone of agile development it is difficult to operationalize and compare this indicator between organizations.

For the perceived EA challenges a number of indicators have been combined and some have been removed. Stakeholder coordination has been removed from the model. According to the literature challenges with coordination are driven by two things: difference in geographical location and time differences. These are unlikely to be addressed by introducing agile practices.

Similar reasoning applies to the indicators EA frameworks, tool support and architect experience. Here again the project management approach is unlikely to have significant impact caused by shortcomings of the tooling used, the EA framework or lack of experience of the architect(s).

4.2.1 Environment

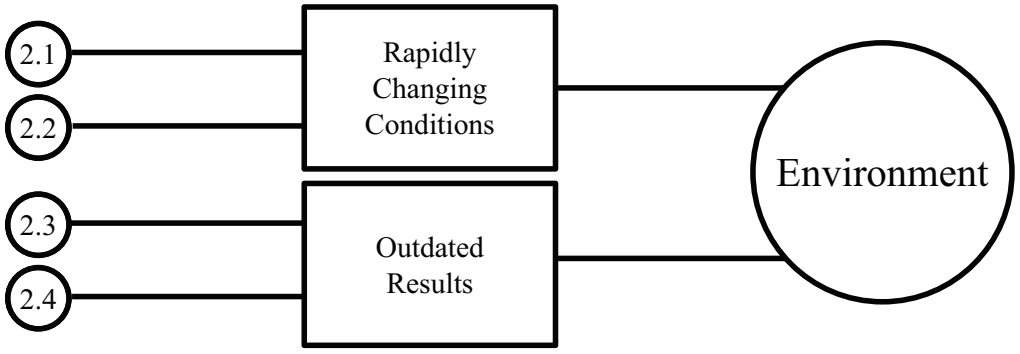

FIGURE 4.6: Indicators and items supporting environment construct

The environment construct in figure 4.6 encompasses all challenges that are related to or are a result of the environment in which EA is conducted. Two indicators have been positioned under environment: rapidly changing conditions and outdated results. The indicators show overlap, outdated results are likely the outcome of rapidly changing conditions.

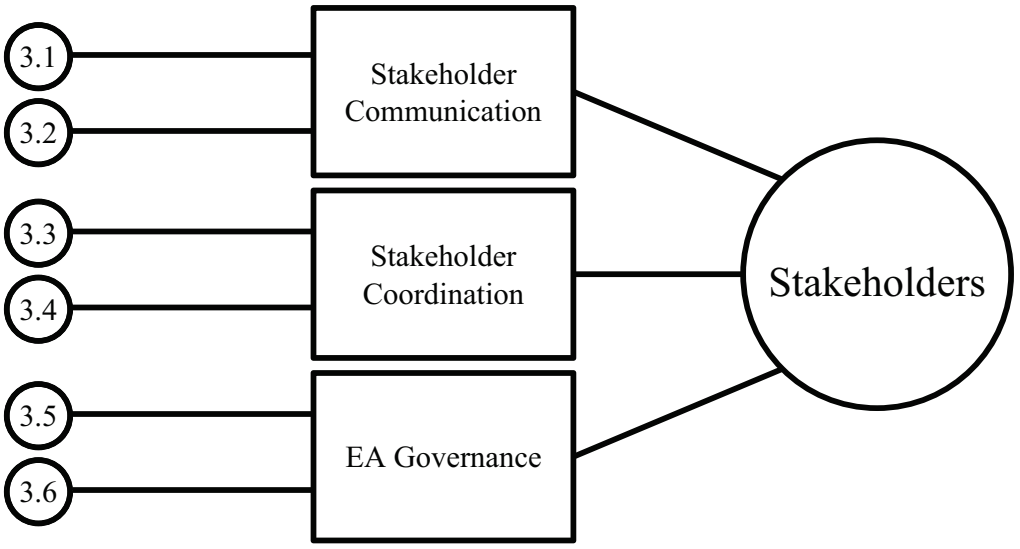

FIGURE 4.7: Indicators and items supporting stakeholders construct

4.2.2 Stakeholders

The stakeholders construct in figure 4.7 encompasses all indicators that concern challenges related to stakeholders of EA: stakeholder communication, stakeholder coordination and EA governance. Governance has also been placed under stakeholders since it is expected that the lack governance structure will be closely related to problems with stakeholder communication and coordination due to the lack of roles and responsibilities.

4.2.3 Organizations Size

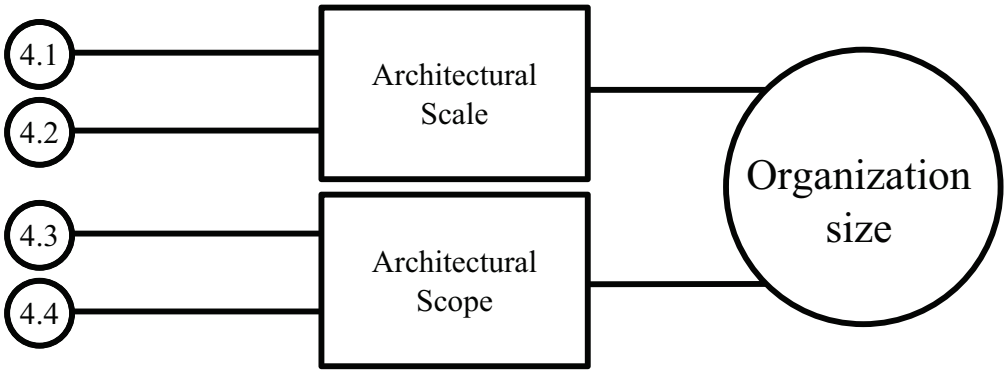

FIGURE 4.8: Indicators and items supporting organizations size construct

The organization size construct in figure 4.8 encompasses all indicators that are related to the size of the organizations: architectural scale and architectural scope. As an organization grows the architectural scale increases and determining the scope can be more difficult.

4.2.4 Deliverables

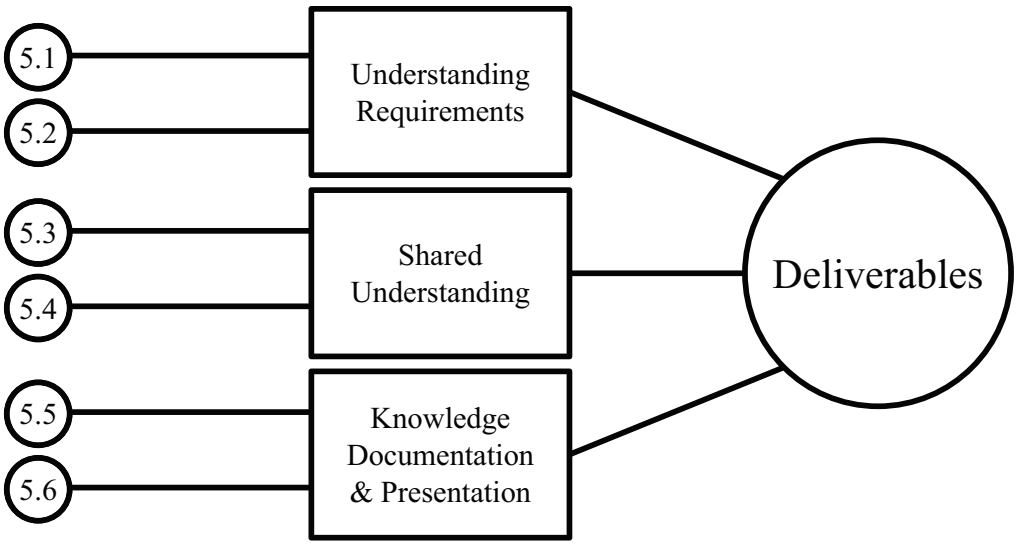

FIGURE 4.9: Indicators and items supporting deliverables construct

The deliverable construct in figure 4.9 encompasses all indicators that are related to EA deliverables or artifacts, e.g. diagrams, lists and matrices. The indicators describe the understanding and documentation of the deliverables.

4.2.5 Continuous Delivery

The continuous delivery construct in figure 4.10 encompasses agile practices that enable the continuous delivery of software. In the case of EA it would not be the delivery of software but that what delivers value for EA. Frequent delivery of working software, dealing with changing requirements and maintaining a constant pace are all practices that enable continuous delivery.

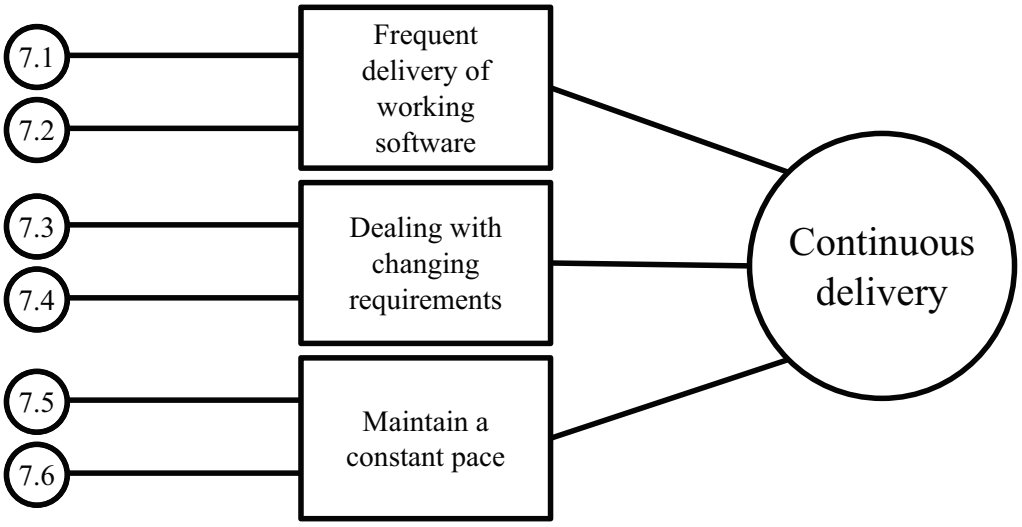

FIGURE 4.10: Indicators and items supporting continuous delivery construct

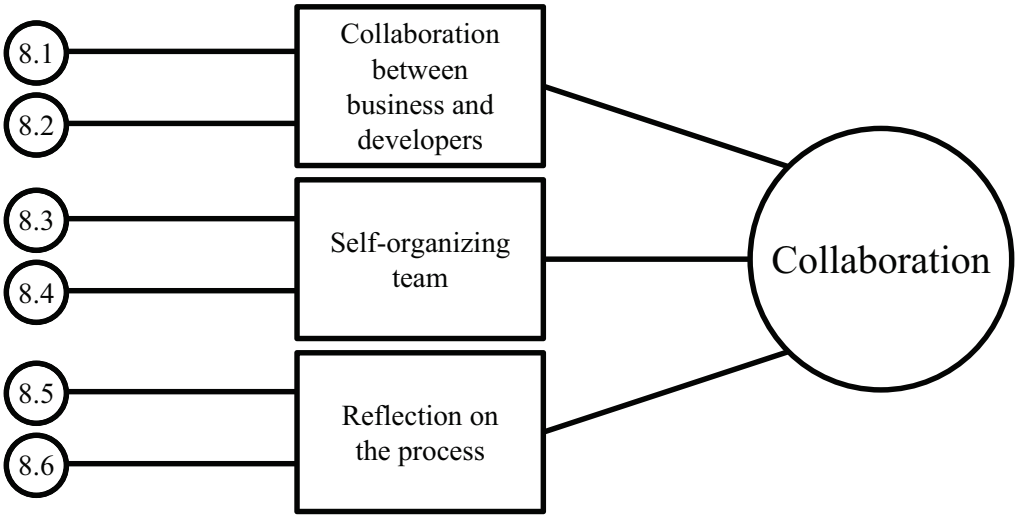

FIGURE 4.11: Indicators and items supporting collaboration construct

4.2.6 Collaboration

Collaboration construct in figure 4.11 contains practices that emphasize the collaboration aspect of agile development: collaboration between business and developers, self-organizing teams and reflection on the process. These have been grouped in one construct: if an organization is addressing collaboration then they are likely to focus to focus on these three practices.

4.2.7 Parsimony

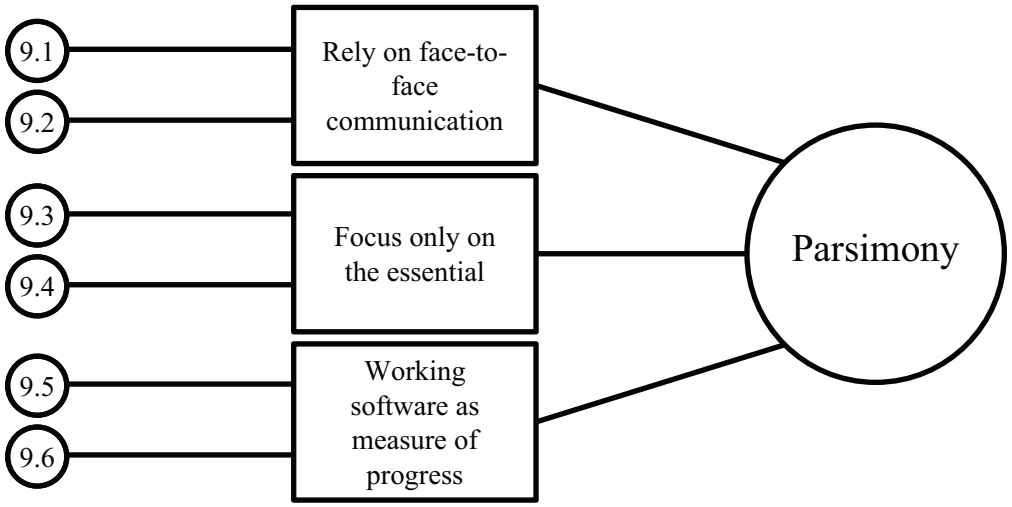

FIGURE 4.12: Indicators and items supporting parsimony construct

The parsimony construct in figure 4.12 captures all practices for reducing use of resources: rely on face-to-face communication, focus only on the essential and working software as measure of progress.

4.3 Conceptual Model

The structural model together with the measurement model form the conceptual model behind the surveys. The conceptual model is given in figure 4.13.

4.4 Scale Development

The scales are available in appendix A. This also shows the scale development: old question, reason for changing the question, and the final questions used on the survey.

The figures in the sub-chapter describing the measurement model already contain the items (circles with numbers) that refer to questions on the survey. This sub-chapter will explain how these items were developed and validated. First will be described how these items were derived from the literature, secondly how these were validated and lastly the final items used on the survey.

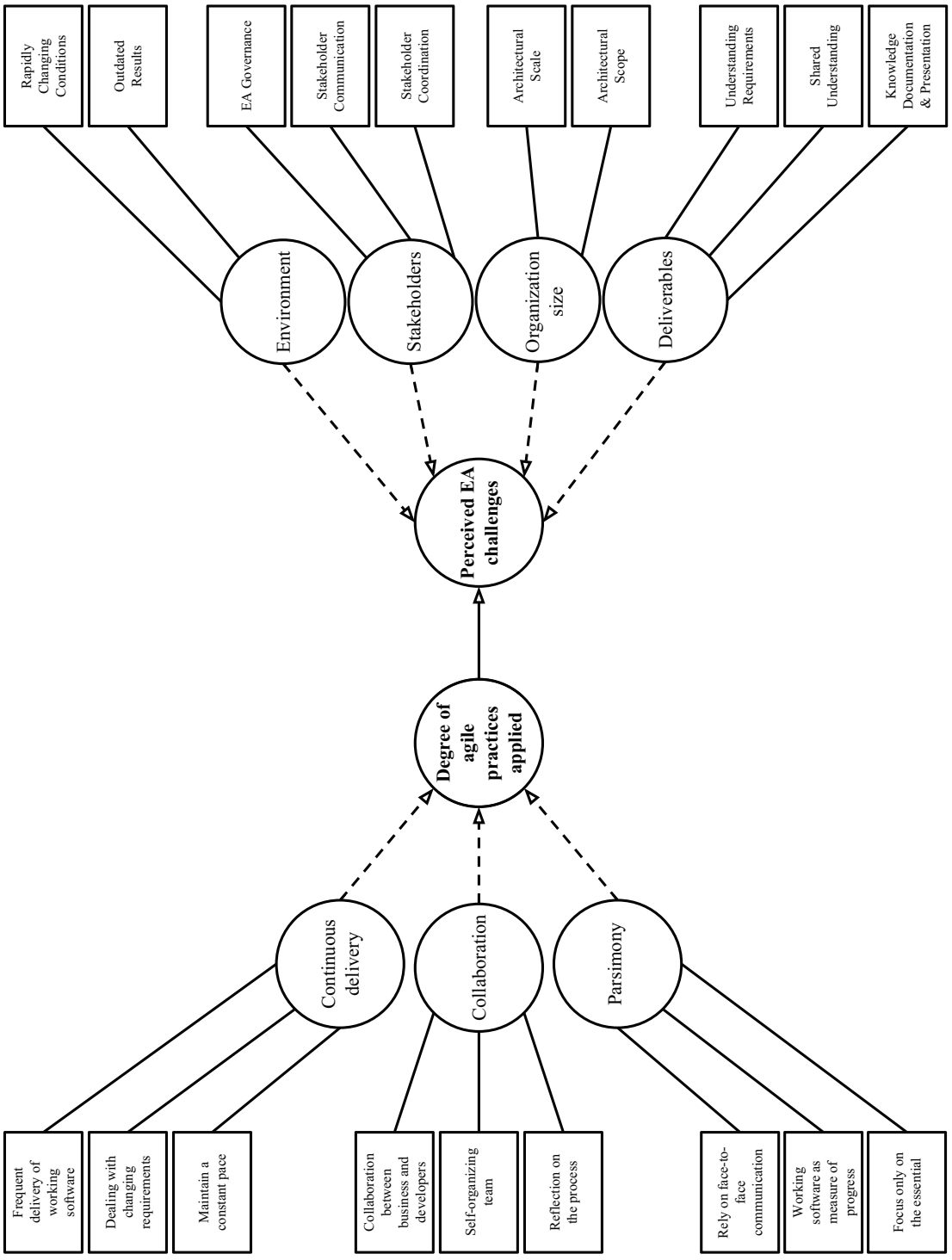

FIGURE 4.13: Conceptual model showing all the constructs and indicators

Based upon the conceptual model and literature questions were formulated. Unfortunately no questions were available that were previously used. Since the questions have not been used before it is unknown whether these are valid. This introduces uncertainty: the items are not measuring the given indicator or the questions are unclear for the respondent. Pre-testing the survey is necessary.

Most indicators are measured by two items. Which items are measuring which indicator is specified in the measurement model. The questions are available in appendix A, the relationship between the questions and the literature is given in table 4.1.

The questions are formulated as statements and are answered on a five-point Likert scale (strongly disagree, disagree, neutral, agree, strongly agree), next to these options respondents can also indicate that they do not know the answer to the question.

The sections below comment on choices made regarding the scales.

Conscious or Unconscious

The survey items measure agile practices applied to EA development. With this approach individual practices can be observed in an organization.

This approach also has a potential downside. It isolates the individual practices of agile development. It has less regard for the relationships between these components. Measuring individual practices enables analyzing agile development beyond a conceptual level. Possible limitation is that it does not necessarily regard agile development from a holistic perspective.

Reflective indicators

Besides the questions directly related to the indicators, six extra questions were added to the survey. These questions are required to conduct an analysis on the data. Without these it would not be possible to validate the formative constructs [49]. For both the degree of agile practices applied and perceived EA challenges three questions are added.

Defining reflective measures for the construct perceived EA challenges is difficult. There are no reflective indicators which directly measure EA challenges. The statements are based on the assumption that more challenges result in lower EA performance. When conducting the statistical analysis the scores will need to be reversed coded. The following three statements are meant to measure the perceived EA challenges.

Benefits of EA include organizational alignment and resource portfolio optimization which can result in reduced costs, business IT alignment and reduced complexity [8].

1.1 EA results in lower IT costs.

1.2 EA contributes to reducing complexity in the organization.

1.3 EA enables better alignment between business and IT.

The same approach has been followed for the reflective measure for the degree of applied agile practices. Again, directly measuring the with a reflective construct is not possible. The approach taken here to use statements reflecting the waterfall approach. This should validate the formative indicators because, on certain elements, agile and waterfall have opposite practices. The scores of these statements will be reversed coded.

These statements are based upon management principles behind waterfall approaches. These are taken from the following article [52], but are generally accepted as being distinct to waterfall development approaches.

6.1 Requirements are fixed at the beginning of the process.

6.2 Each step and/or stage in EA projects and/or processes are documented extensively.

6.3 Steps and/or stages are planned in detail early on in the process and/or project regarding EA.

4.4.1 Pre-Testing

Within Deloitte Consulting the survey was distributed. Consultants were asked to complete the survey based upon the situation of a client. The survey was specifically targeted at those who have experience with EA assignments. Besides completing the survey, they were asked to give feedback on the questions. A total of eight surveys were completed.

With a pre-test it is preferable to conduct a statistical analysis in order to determine the reliability and validity of the questions. This does require having a significant sample

size. Eight samples are not enough, even with double the amount of samples a statistical analysis would be questionable. Results would be too dependent upon individual cases.

The pre-test uncovered a number of issues with the survey. These are listed below. This resulted in changing a number of the questions, which questions were changed is shown in appendix A.

- Questions about EA challenges were formulated negatively. This is considered a bad practice since it can be confusing, but was initially done because problems have a negative annotation from themselves, and were formulated as such. After feedback these were reformulated: not because they were confusing but because it introduces the chance that respondents give a too favorable view of the situation.
- Terminology was made simpler. The survey contained numerous concepts that were not necessarily mutually exclusive: EA project, EA process, EA deliverable and EA artifact.
- One of the validation questions for the EA challenges constructs was replaced. The question was ambiguous and interpretations of the respondents differed resulting in inconsistent answers. This could be problematic when analyzing the data, as a precaution the question was replaced.
- Some superficial adjustments: numbering of the questions was adjusted, the term parsimony was changed to simplicity (although this changes the meaning it is more familiar for the respondents) and the accompanying texts were slightly changed.

4.4.2 Final Scales

The final survey is available in appendix A. The appendix contains the survey used for the pre-test and the final survey. For each question the initial question is given, and if a question was changed the reason for the change is also given.

Item	Question	Source
1	The EA team is able to keep up with current release cycles in the organization.	[9]
2	The EA team is able to keep up with changes in the business demands.	[20] [9]
3	EA artifacts are delivered on time to be useful for projects.	[20] [30]
4	EA artifacts are mostly on time for business or IT transformations.	[20]
5	Conveying EA artifacts to stakeholders requires little effort.	[27]
6	Communication with stakeholders, even those with a different background, is effortless.	[25]
7	Stakeholders are usually available when information from them is required.	[20] [9]
8	Stakeholders are forthcoming in providing necessary information. information	[20]
9	The policies and procedures in place for EA are adequate. EA	[5] [25]
10	Roles and responsibilities of EA are clear for the stakeholders.	[5] [25]
11	The size of the organization does not pose any problem when modeling the EA.	[9]
12	The amount of different architecture views is manageable.	[9]
13	The initial scope set at the beginning of EA development* is rarely changed during the course of development.	[22]
14	Determining the scope at the start of EA development is effortless.	[22]
15	Stakeholders are clear on what is expected from the EA team.	[20]
16	Different departments have similar expectations from the EA team.	
17	Goals from the business are clear for the EA team.	[20] [9]
18	The EA team has relatively more focus on the IT than the business.	[27]
19	All produced EA artifacts are perceived as useful by stakeholders.	
20	EA artifacts meet the needs of the stakeholders.	
21	An iterative approach is followed regarding EA development.	[37]
22	An incremental approach is followed regarding EA development.	[37]
23	The EA team is capable of incorporating new requirements of the architecture during development.	[35]
24	Requirements of the architecture are only defined at the highest level required.	[35]
25	Business and IT are both closely involved in planning EA development.	
26	The workload can be described as being equal over time.	
27	At various points during EA development the business is involved.	[32]
28	The business regularly supplies the EA team with feedback.	[32]
29	Roles and responsibilities are determined by the team members themselves.	[33]
30	Team members assign themselves work.	
31	Meetings are organized to reflect on work done.	[33]
32	Feedback is an inherent part of the process.	[33]
33	The amount of EA artifacts is kept to a minimum.	[33]
34	Communication with stakeholders is primarily done face-to-face.	[38]
35	All EA artifacts produced are used by the organization.	[33]
36	Achieving transformations would be possible with less EA artifacts.	[33]
37	On a daily basis progress on EA development is discussed.	[33]
38	Progress on EA development is closely tracked.	[33]

Table 4.1: Grouping of indicators for EA challenges

Chapter 5

Results

This chapter describes the results that were obtained by the survey. The structure of this chapter is as follows:

- Chapter 5.1 (Data transformation) – Before any analysis of the data is possible, it needs to be transformed. Transformation of the data is described in this section.
- Chapter 5.2 (Data description) – Descriptive statistics are used to describe and summarize the data obtained by the surveys.
- Chapter 5.3 (Checking assumptions) – Checking of assumptions before applying statistical analysis.
- Chapter 5.4 (Statistical analysis) – Statistical analysis of the surveys which are obtained by applying multiple linear regression.

These sections together result in the following outcome: the three agile practices listed below are capable of explaining the variance in the EA challenges in organizations, i.e. focusing on these challenges is likely to reduce the EA challenges perceived.

- Dealing with changing requirements
- Reflection on the process
- Focus on only the essential

5.1 Data Transformation

The data extracted from the surveys is transformed. Measures are combined, missing values are addressed and items are reverse coded. This was done in the following sequence. For each step is explained what was done and why, the outcome of the steps is available in appendix B

1. **Removing incomplete cases**
   After removing cases that were not fully completed. There are 34 cases left.

2. **Reverse coding of question 5.4 and 9.4**
   Answers of both questions needed to be reverse coded since the phrasing is negative as opposed to the other questions which are formulated positively, i.e. answering strongly agree to these questions indicates that a problem is perceived whereas with the other questions strongly agree indicates a problem is not perceived at all.

3. **Calculate Cronbach's alpha**
   Most of the indicators on the survey are measured by two questions. This makes it possible to determine the reliability with the use of Cronbach's alpha.

4. **Remove problematic measures**
   Some measures have a low Cronbach's alpha which needs to be addressed. This can be done by removing the indicator or an individual measure if there is a reason to do so.

5. **Missing values and combining measures**
   Respondents had the possibility of indicating that they do not know the answer to the question or that it is not applicable. Before analyzing the data these missing values need to be addressed. Two methods were used for addressing the missing values: if there was a value available from another measure which measures the same underlying concept it was replaced with that, else the mean was calculated for the given construct was used.

As can be seen in appendix B the data transformation resulted in removing two items, and merging two items. Combining measure was done by using the mean and not the sum. This overcomes the possibility that missing values impact the measure.

5.2 Data Description

This section visually describes the data obtained by the surveys. Demographics on the respondents are given and an aggregation of the answer on the agility of EA development and EA challenges perceived. The sample size is 34.

5.2.1 Demographics of respondents

In figure 5.1 the respondents per sector are given. Respondents had the possibility of selecting one of five sectors: (1) Consumer Business, (2) Financial Services Industry, (3) Manufacturing, Energy & Resources, (4) Public Sector and (5) Technology, Media & Telecommunications. Some respondents did not supply the sector their organization operates in.

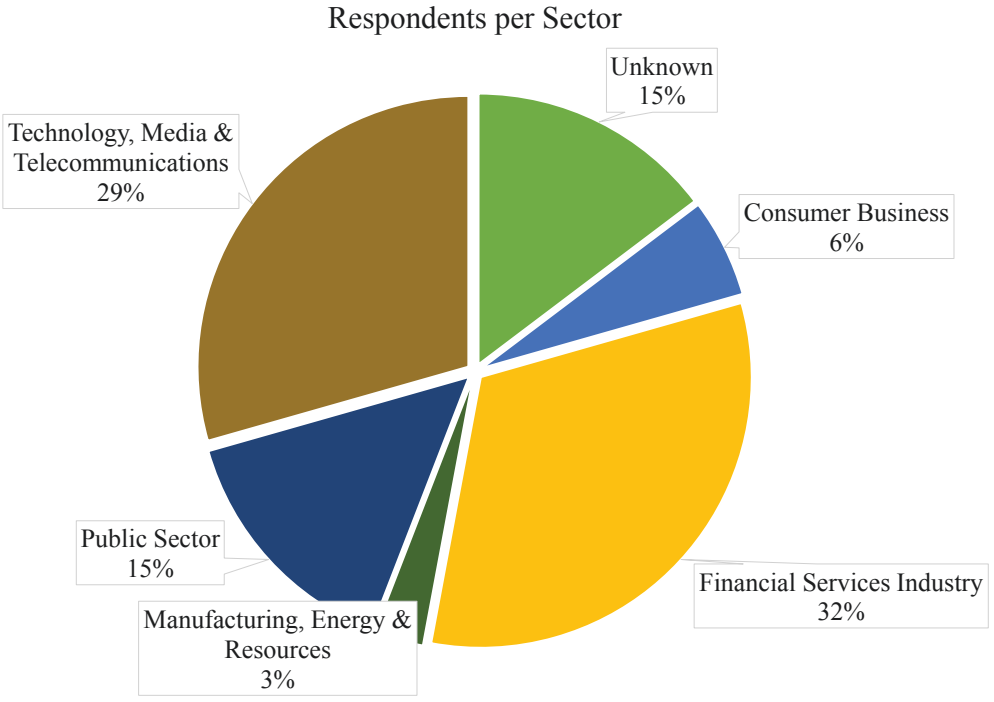

FIGURE 5.1: Respondents per Sector

5.2.2 EA Challenges

In figure 5.2 a break down of the answers per question on EA challenges is given. A small footnote about the visualization: *Strongly Agree* indicates that a challenge is **not** perceived, *Strongly Disagree* indicates that a challenge **is** perceived. This difference is

caused by the phrasing of the questions on the survey. The numbers on the horizontal axis refer to the questions on the survey, which are available in appendix A.

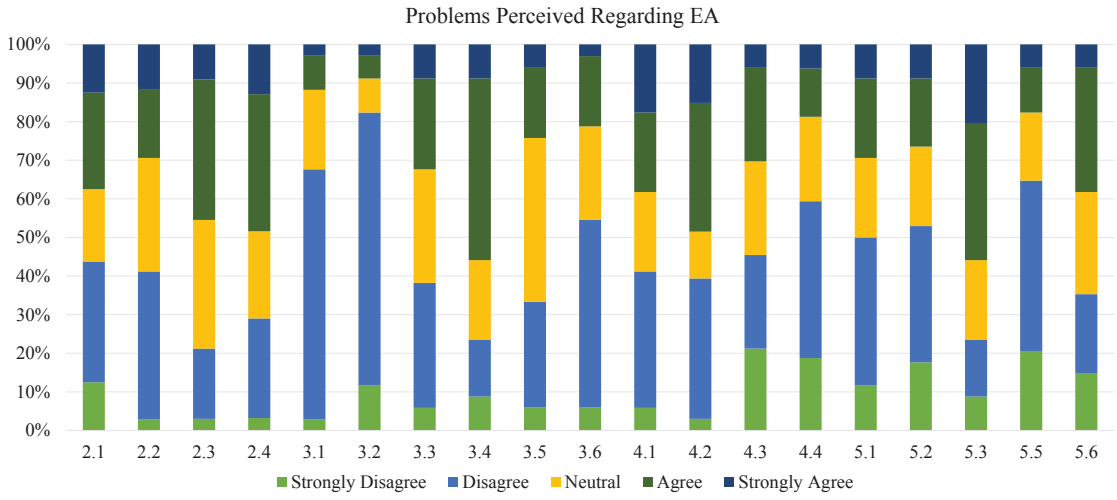

FIGURE 5.2: Problems perceived regarding EA

5.2.3 Agile Practices

In figure 5.3 a break down of the answers per question on agile practices is given. In this graph *Strongly Agree* indicates that an agile practices is applied within the organization.

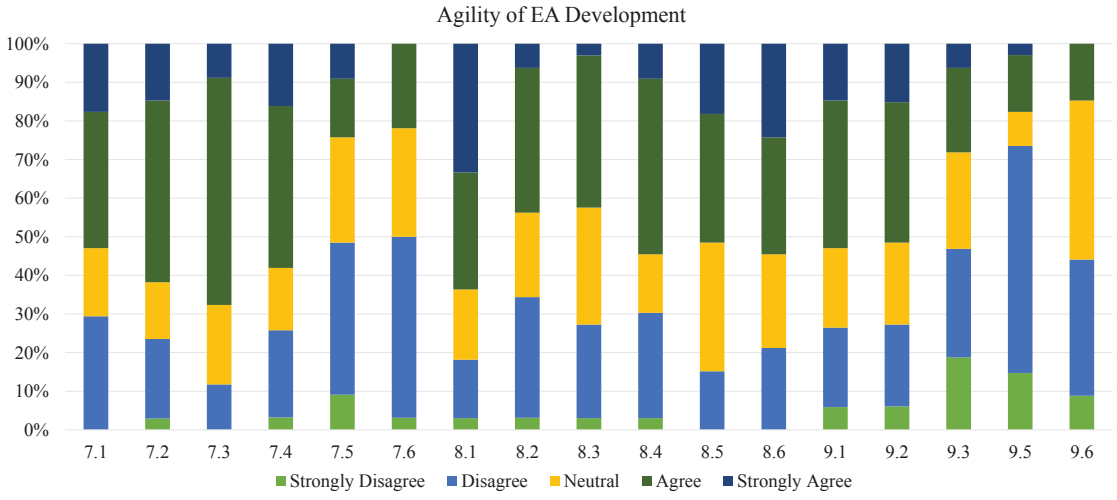

FIGURE 5.3: Agility of EA development

5.3 Checking Assumptions

The statistical analysis, chapter 5.4, contains two types of analyzes. The initial idea was to analyze the survey results using Structural Equation Modeling (SEM). More

specifically PLS-SEM would be used, as opposed to CB-SEM, since it better suited for exploratory approach and smaller sample sizes [49]. Assumptions for SEM are described first after which the assumption for multiple linear regression are explained.

5.3.1 Structural Equation Modeling

The SEM analysis cannot be used because the assumptions do not all check out. It is included because it is the reason a different statistical method was used: multiple linear regression. This gives others insights as to how this research was developed.

In determining the validity there are four steps that are taken into account [49]: content validity, convergent validity, collinearity and significance of the constructs. First each concept will be explained, after which it will be applied to the data set. By applying these steps the reliability and validity of the data can be assured.

- **Content validity**

  Content validity is about whether the items are measuring what they are supposed to measure. Achieving content validity requires thorough literature review and theoretical basis. The conceptual model is based upon the literature review which was done on agile practices and EA challenges. It is reasonable to presume that this thorough literature search resulted in measures which cover, most of, the content domain that they are intended to measure.

- **Convergent validity**

  Convergent validity, which is also referred to as redundancy analysis, is using multiple indicators to measure the same construct. Convergent validity determines for a given group of indicators whether these are measuring the same underlying construct. When R^2 for the formative measure is below 0.64 then there are convergent problems. In that case the reflective and formative measures do not measure the same underlying construct. This is a strong indication that the one of the measures does not contribute sufficiently to the construct.

- **Collinearity analysis**

  The variance inflation factor (VIF) is a measure to determine collinearity. VIF measures the amount the standard error has increased due to collinearity. A value

above 10 is considered to be problematic, although some set the limit at 3 [50]. When collinearity is present it is advisable to remove the measure. Removing measures requires careful consideration, since the combination of the measures defines the formative construct. It should be ensured that the remaining measures still cover all the aspects of the construct domain.

- **Significance of measures**
  Formative measures are the building blocks of the construct. These buildings blocks do not all necessarily contribute equally to the construct. This introduces the possibility that certain measures may not contribute to the construct at all. It is therefore of importance to validate whether each of the measures are significant.

The above assumptions were checked with the use of SmartPLS [53]. The results are available in appendix C. The results indicate that the formative constructs are correctly defined, but are not significant in the model. The exact cause of this is unclear, a possibility is the sample size which is on the low side for the number of constructs in the model.

As shown in appendix C not all the assumption for SEM check out. A different method is used for the data analysis.

5.3.2 Multiple Linear Regression

The conceptual model was built with SEM in mind, i.e. reflective measures were added in order to validate the formative measures. Applying multiple linear regression requires extra transformation of the data. The dependent variables will be the predictors of the agile practices. The independent variable for the model, EA challenges, is acquired by taking the mean of all the predictors of perceived EA challenges.

As with SEM, before applying multiple linear regression a number of assumptions [54] need to be validated. These are the following assumptions. A short description of each assumption is given. Appendix D contains the outcome of the validation.

- Variable types: all the variables should be measured on an interval level. All items on the survey, which are used for this analysis, are measured on a Likert scale. This assumption checks out.

- Non-zero variance: as can be seen in the appendix none of the predictors have a standard deviation which equals zero. No non-zero variance is present in the data.
- Predictors are uncorrelated with 'external variables': states that certain variables, external variables, which are not included by do have an influence. Considering both constructs (EA challenges and agile practices) were defined by an extensive literature search it can be reasonably assumed that no variables are left uncovered.
- Normality: whether the distribution of the data conforms to a normal distribution. This is checked by a visual representation of the distribution which is given in a histogram. This data exhibits a normal distribution.
- Linearity: linear relationship between the dependent and independent variables. This assumption can be evaluated by determining the presence of non-linearity. This can be done by plotting the observed against the predicted values. As shown in Appendix D, this assumption checks out.
- Homoscedasticity: similar amount variance between dependent and independent variables. Can be checked by plotting the standardized residual values against the standardized predicted values.
- Independence of errors: errors should be independent, i.e. not correlate with each other. This can be be validated by calculating the Durbin-Watson value. The values are well within limits.
- Multicollinearity: predictors should not correlate. This can be checked by either looking at the variance or the VIF (Variance Inflation Factor). The values are all well within limits.
- Independence: there is no measure or technique available to determine the independence. Independence is achieved when all variables in the model are the outcome of separate entities. For the predictors used this assumption is most likely valid: the items on the survey are all measuring different underlying concepts. EA challenges and Agile practices are completely different entities.

The assumptions all checked out. The data is suitable for multiple linear regression. The outcomes of the steps taken for validating the assumption are available in appendix D.

5.4 Analysis of Data

The results are divided in three parts. The first section describes the influence of agile practices on EA challenges. The second section describes the influence of EA challenges on agile practices. The last section describes a correlation analysis between the agile practices and EA challenges.

5.4.1 Influence of Agile Practices on EA Challenges

A multiple linear regression analysis was used to validate the model. The research is exploratory and there is a limited sample size. Validation of the model will be limited to identifying agile practices which significantly impact the problems concerning EA. Thus linear regression will identify agile practices which significantly explain the problems in organizations.

Forced entry was used as regression method. This method forces all predictors in the model at once. A stepwise method could also be applied. This method selects predictors based upon their significance, and omits all other predictors. Stepwise methods are recommended for exploratory approaches. An explicit choice was made not to employ stepwise methods, even though the approach is exploratory. Automatically omitting variables by an algorithm leaves less room for interpretation. If the analysis shows that a predictor is not meaningful, i.e. not significant, that provides meaningful insights. It creates new questions as to why a particular predictor does not impact EA challenges.

The analysis unveils that three predictors are significant in predicting EA challenges with p <0.12. These are B73 (β = 0,242), B85 (β = 0,235) and B93 (β = 0,464), this translate into the following predictors: dealing with changing requirements, reflection on the process and focus only on the essential. A visual overview of the results is given in figure 5.4.

Other predictors are not significant. These include the following: frequent delivery of working software, collaboration between business and developers, self-organizing team, rely on face-to-face communication and working software as measure of progress.

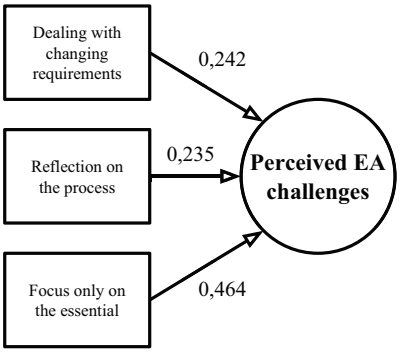

FIGURE 5.4: Significant agile practices in predicting EA challenges

5.4.2 Influence of EA Challenges on Agile Practices

The conceptual model does not contain this relationship: the relationship is only modeled in one direction. Multiple regression was done on this relationship. This shows that none of the EA challenges were capable of predicting the agile practices. P-values indicated that none of the predictors are significant.

5.4.3 Correlation between Agile Practices and EA Challenges

The regression analysis does not show which EA challenges are addressed by agile practices. A correlation analysis was done on the significant agile practices and all the EA challenges.

The Spearman correlation coefficient was calculated with SPSS. The Spearman's correlation is non-parametric and does not assume normally distributed data [54]. Spearman's correlation is used because it is suited for the type of data used.

The results are available in appendix E. The following significant correlations between agile practices and EA challenges are present.

- Dealing with changing requirements: rapidly changing conditions and stakeholder communication.
- Reflection on the process: outdated results and shared understanding.

- Focus only on the essential: rapidly changing conditions, EA governance, architectural scope, understanding requirements and knowledge documentation & presentation.

Chapter 6

Discussion

The statistical analysis showed that the following agile practices are predictors for the EA challenges perceived (in order of significance): focus on the essential, reflection on the process and the ability to deal with changing requirements. These three practices are able to explain 50% of the variance in EA challenges among the organizations.

This is a strong indication that EA benefits from agile approaches. The significant agile practices are areas an agile EA approach should focus on, and are essential when developing such an approach.

Besides the statistical analysis the description of the data contains some noteworthy outcomes. The perception of individual EA challenges and agile practices in an organization results in some interesting points of discussion. Similar data is also available in previous research, results of this thesis can be compared to previous research.

The descriptive statistics are first interpreted, explained and compared to previous research. After which the outcomes from the statistical analysis are discussed. Numerous agile practices were non-significant. Although these are non-significant they offer insights, and might have implications.

6.1 Descriptive Statistics

The following highlights EA challenges and agile practices which are outliers: predictors which are predominately observed or rarely observed.

6.1.1 EA Challenges

The list below contains EA challenges with more than 50% of the respondents perceive in their organization. Most of the EA challenges that stand out are related to stakeholders: stakeholders communication & conveying artifacts, unclear roles and responsibilities for stakeholders and perceived usefulness of artifacts. When applying the classification scheme from Lucke et al. [9] most of the problems can be placed in the category: understanding and management of EA. Only scoping of EA development can be placed in the other category, modeling complex systems.

EA challenges more than 50% of the respondents perceive in their organization:

- Stakeholder communication & Conveying artifacts
- Unclear roles and responsibilities for stakeholders
- Scoping of EA development
- Perceived usefulness of artifacts

The following challenges are perceived by less than 30% of the respondents. The data shows that more than 50% agrees with the statement that the goals from the business are clear. This contradicts earlier research [20], which states that almost 85% agrees that business goals are unclear. The same goes for: stakeholders are forthcoming in providing information, the data shows that a little over 20% has reluctant information providers. Previous research [20] indicates that almost 65% has reluctant information providers.

EA challenges more than 30% of the respondents perceive in their organization:

- Artifacts are on time
- Stakeholders are forthcoming in providing information
- Clear goals from the business

The large differences with previous research is remarkable. It is unlikely that all the organizations that participated in the survey have very different stakeholders which are

both forthcoming in providing information and have clear goals. The use of EA frameworks was not measured, but that could be an explanation as to why these challenges are perceived less. The frameworks used might prescribe processes which reduce the perception of these challenges.

Although these challenges are perceived less compared to previous research, other challenges related to stakeholders are not. Most of the challenges that more than 50% of the respondents perceive in their organization are related to stakeholders. Challenges related to stakeholders are still perceived. This stresses the importance of agile methodologies which put a focus on collaboration.

In the problem statement it was assumed that architects have difficulty with adapting to the environment due to the pace of change. Items related to this are agreed upon by approximately 40% of the respondents. That is a significant amount but are not challenges that architects are most affected by. On one hand this validates the assumptions made in the problem statement: dealing with change is an area that architects can definitely approve upon but is not the most agreed upon challenge.

Combining measures delivers interesting observations. Architects deliver EA artifacts mostly on time (45% of the time), but stakeholders rarely perceive the artifacts as useful (18% perceived as useful). Architects are producing deliverables which are on time but not useful. Over-sized and too difficult documentation is a common EA challenge. Architects should focus on learning what stakeholders need.

6.1.2 Agile Practices

Following agile practices more than 60% of the respondents agreed upon.

- Incremental approach
- Incorporating requirements during EA development
- Involvement of business during EA development

Incremental is the most used agile practices for EA development. This is line with earlier research [16]. It is unclear what is being developed incrementally. The survey

specifically asked about the delivery of a usable product, but that is still ambiguous. That could be artifacts, transformation or interactions with stakeholders.

Following practices more than 40% of the respondents disagreed upon.

- Progress tracking of EA development
- Involvement of business and IT in planning EA development
- Equal workload over time

A strong agile practice is the involvement of the business during EA development. This is not striking, they are most likely the stakeholder that initiated the process. Although the business is involved in the process they are not involved in the planning of EA development.

Architects do not perceive an equal workload over time but at the same time there is no tracking of the progress. The planning is done independently but does result in an unequal workload. If the business, and possibly other stakeholders, are involved in the planning process this challenge could be addressed.

6.2 Linear Regression

To obtain sufficient exploratory power with the conceptual model requires a sample size of approximately 80. This total has not been reached. A different problematic aspect is the amount of uncertainty in the model and questions. All the predictors have a theoretical basis, but the relationships in the model have not been previously tested. This limits what can be done with the data and the conclusions made based on the data. The analysis is not focused on theory testing, but has an exploratory approach.

Below each of the significant agile practices are elaborated upon.

Dealing with changing requirements
In software development the ability to deal with changing requirements is enabled by the combination of incremental and iterative development. Developing in short cycles and continuously delivering working software enables a development team to incorporate new

requirements. In the case of EA the predictor for incremental and iterative development is non-significant, but the ability to deal with changing requirements is somehow enabled.

This difference might very well be result of a fundamental difference between EA development and software development. In software development the piece of software is what delivers value to the customer. In achieving that piece of software various methods and tools, ranging from technical tools such as integrated development environments (IDEs) to practical methods such as user stories are used. In EA development the artifacts are the methods and tools to deliver value, the artifacts themselves are not what deliver value.

In software development that what delivers value to the customer is developed incrementally and iteratively: this is the piece of software. Following this line of reasoning the same should apply to EA development. This does raise the question what is delivering the value for EA. The process that creates resulting in the EA artifacts could very well be what is delivering value.

The survey asks specifically about EA artifacts which are defined as deliverable or product from the architect. But if for example stakeholder interaction is what delivers value to the customer then that should be done incrementally and iteratively.

Reflection on the process

One of the agile principles pertains to reflecting on behaviour in order to become more effective. Having frequent meetings to reflect on work done and having feedback as an inherent part of the process results in less EA challenges.

That reflecting on work done has a positive impact on challenges that are perceived might be obvious. Reflecting on the process is one thing, but translating this into improvements might be more difficult. The data shows that reflection on the process results in less EA challenges.

Focus on only the essential

Focus on only the essential relates to whether the artifacts produced by an organization are actually used. Agile principles emphasize parsimony. This includes limiting documentation by focusing on face-to-face communication and closely tracking development progress. Although this practice is significant in predicting the perceived EA challenges,

it seen rarely among the organizations. Little below 30% of the respondents indicate that all produced artifacts are actually used by the organizations.

TOGAF is considered to be documentation driven [17]. The data indicates that focusing on the essential reduces the EA challenges perceived. Adapting current methodologies such that there is less focus on the production of documentation is likely to result in benefits. This is in line with one of the agile values: working software over comprehensive documentation [15].

6.3 Correlations

Correlations between the significant agile practices and EA challenges were calculated. It is expected that an agile practice addresses certain EA challenge(s). An agile practice that correlates with an EA challenge is an indication of this.

Some critical side notes must be placed. The sample size is 34 and 30 correlations are calculated. With such a small sample size and high number of correlations the results must be interpreted with caution. Correlations are likely to be always found. A number of the correlations described in chapter 5.4 are discussed.

Dealing with changing requirements addresses rapidly changing conditions and stakeholder communication. The first challenges is unsurprising: being able to take on new requirements enables you to better deal with changing conditions. The latter does not have a theoretical basis.

Reflection on the process addresses outdated results and shared understanding. You would expect that reflection on the process addresses more than just two EA challenges. Reflection should result in overall improvements. The data does not show this. This is an indication that achieving improvements by reflection is difficult in practice.

A focus on the essential addresses numerous EA challenges: rapidly changing conditions, EA governance, architectural scope, understanding requirements and knowledge documentation & presentation. The survey specifically measured the use of EA artifacts. Organizations that have put thought to what is essential, i.e. the requirements from stakeholders for EA artifacts, might be more mature. This could be the reason that a

focus on the essential addresses so many EA challenges. But with the amount of data available this is difficult to say.

6.4 Validating Outcome Survey

Results obtained by analyzing the survey have been validated by experts. This validation was done with the use of a round table which was organized in the light of this master thesis. Architects from various organizations, which participated in the survey, were present at the round table.

The goal of the round table was to validate the results that were obtained by the statistical analysis done on the survey. To ensure that round table would indeed result in a proper validation, a predefined structure was followed. Based on the data four statements were formulated. The statements were discussed during the round table. A short summary of the discussion per statement is given:

- **Architects produce deliverables that are on time but not useful for stakeholders**

  The data indicates that EA artifacts are rarely perceived as useful, but are delivered on time. The regression analysis shows that a focus on the essential has an impact on EA challenges.

  During the discussion a number of points were put forward. Artifacts are not necessarily what delivers value. The process resulting in the artifact is what delivers value. Artifacts are merely a tool for documenting.

  Creating a definition of done addresses usefulness and timeliness. It defines when an artifact is useful and can create a sense of time pressure for architects.

- **Architects understand the needs of stakeholders but are unable to communicate artifacts to stakeholders**

  Some consider this a challenge which is not addressed by the development approach. Communicating artifacts is a skill of the architect. The architect should have the ability to communicate the artifacts properly.

  This is also influenced by the amount of creativity that is given to the architect and the formalization of artifacts. An agile practice that could address communication

challenges is a focus on face-to-face communication. In practice this could prove more difficult due the amount of stakeholders and in the end decisions must be documented.

- **EA development should not be done incrementally but iteratively**
  A purely incremental or iterative approach to EA is not possible. You always need some rough frame before taking an incremental or iterative approach. First create abstract boundaries and principles which can be made more concrete level-by-level.

  Multiple interpretations exist to what can be developed incrementally or iteratively. Both the layers of the architecture as the communication with stakeholders can be done incremental or iterative.

- **EA development progress is rarely tracked, it does not result in benefits**
  Daily tracking of EA development is not useful. Tracking of EA development over a longer time frame could be. The low amount of organizations that track EA development could be due to ambiguity. The questions on the survey do not state what is being tracked.

  The tracking of EA development progress can be changed by introducing a definition of done. This defines an objective that much be reached for an artifact to be finished. In turn this enables time boxing of development. This simplifies the tracking of EA development progress.

6.5 Requirements for Agile EA Approach

The results of the survey and round table result in a number of requirements for an agile EA approach.

- 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.

Chapter 7

Conclusion

This chapter is divided in four sections. The first section describes the conclusions of this research by providing answers to each of the research questions. After which the contributions to theory are detailed. The last two sections describe the limitations of this research and areas possible future research could focus on.

7.1 Conclusions

1. What are the current challenges with EA development?

The literature review in chapter 3.1 revealed the following EA challenges.

- EA Governance: there are unclear roles and responsibilities for stakeholders of EA development.
- Stakeholder coordination: consists out of the willingness of stakeholders to cooperate and stakeholders which are unavailable or reluctant to provide information.
- Stakeholder communication: challenges with communication of EA artifacts due to different background and needs of stakeholders.
- Understanding requirements: both business & IT requirements are important. Mostly there is a too strong focus on IT, and too little attention for the business.
- Shared understanding: unclear understanding of the business goals and unclear demands from stakeholder for the architects.

- Knowledge documentation & presentation: inability to produce the right amount of documentation. Over-production and too complex EA artifacts are the result.
- Architectural scale: difficulties of modeling the various different domains of the architecture and the size of the organization.
- Architectural scope: challenges with setting the scope at the beginning of EA development. During development it is discovered that the scope was poorly chosen.
- Rapidly changing conditions: a quickly changing enterprise environment. Both business demands and software lifecycles are moving faster. Keeping up with the pace of change introduces new challenges.
- Outdated results: the inability to deliver EA artifacts on time. IT or business transformations have begun without the EA artifacts.

2. What characterizes agile software development?

Chapter 3.2 provides the answer to this sub-question. Based upon the agile manifesto and literature review resulted in the following list of characteristics supported by agile practices.

- Dealing with changing requirements: the ability to incorporate new or changed requirements during development.
- Frequent delivery of working software: developing software incrementally and iteratively to enable short cycles. Each cycle delivers working software.
- Collaboration between business and developers: driven by collocated teams and having the business involved the planning.
- Create trust and motivated individuals: agile development emphasizes the individual. This is achieved by practices such as empowering the individual and careful team composition.
- Rely on face-to-face communication: documentation should be kept to a minimum. This can be achieved by putting a focus on face-to-face communication thereby reducing documentation.

- Working software as measure of progress: delivery of working software should be almost continuous. Having daily progress meetings and progress tracking are practices to ensure continuous delivery.
- Maintain a constant pace: iterations should be time-boxed. Each iteration contains the same amount of work. The workload should be constant over the iterations.
- Technical excellence and good design: an emphasize on the right processes, people and tools. This practice was not incorporated further on in research due to the fact that it is difficult to measure and apply to EA development.
- Focus on only the essential: related to relying on face-to-face communication. Delivering working software is key. This is achieved by limiting formal documentation and the number of resources needed to achieve goals.
- Self-organizing team: emphasize on interactions between individuals. A team should be self-managed and have collective ownership of the products produced.
- Reflection on the process: reflecting on work done by continuous feedback and retrospective meetings.

3. Does EA benefit from agile approaches?

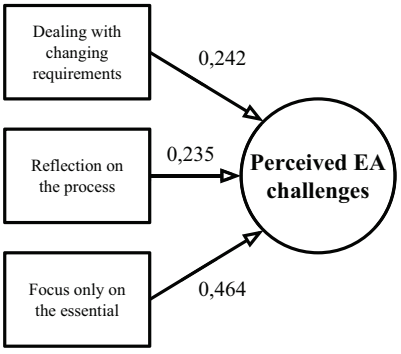

FIGURE 7.1: Significant agile practices in predicting EA challenges

Based upon the literature review in chapter 3.3 & 3.4, current EA challenges, current approaches to EA development and an empirical study it was shown that EA can benefit from an agile approach. With the use of literature it was shown that current approaches to EA resemble that of a waterfall methodology and the drivers for a transition from waterfall to agile are applicable in the context of EA.

Analysis of the surveys unveiled that three agile practices are significant in predicting the EA challenges perceived in an organizations. This together with the evidence based upon the literature shows that EA can benefit from agile approaches. The agile practices are shown in figure 7.1, for each agile practices the standardized coefficient is given.

All other agile practices were not significant. Noteworthy is that both incremental and iterative, important agile practices, are also not significant. The correlation analysis shows that reflection on the process only correlates with two EA challenges. You would expect to see this agile practices correlate with more EA challenges.

4. What are the requirements for an agile EA approach?

The empirical study in chapter 4 & 5 used to determine whether EA benefits from agile approaches gives an indication what an agile EA approach should focus on. The significant predictors in the model should be part of an agile EA approach. The requirements are based upon the agile practices from conceptual model which are significant.

- 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.

5. Do the requirements result in an applicable agile EA approach?

The round table offers a different perspective on the data obtained by the survey. The four statements discussed offer background on which agile practices could be useful in practice. The following key points have been taken into account:

- 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.
- 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.
- Incremental and iterative development are possible but require rough borders: high-level principles and boundaries.
- 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.

How can agile approaches originating from software development be incorporated in the development of an enterprise architecture?

This research shows that EA development can benefit from agile approaches. More specifically the following agile practices are of importance: the ability to deal with changing requirements, reflecting on the process and a focus on the essential. An emphasize on these agile practices is very likely to result in less EA challenges perceived in an organization.

Creating a focus on the essential can be achieved by carefully considering EA artifacts. The research shows that EA artifacts are rarely perceived as useful and that communication with stakeholders is difficult. Creating a definition of done could address such issues. It creates a common understanding of useful EA artifacts, a sense of time pressure for architects and enables the tracking of EA development progress.

Dealing with changing requirements has a positive impact on EA challenges. By setting high-level boundaries and principles an incremental and iterative approach can be taken. Increments and iterations can be done both on EA artifacts and stakeholder communication.

Reflection on the process addresses EA challenges. Which EA challenges this will solve is difficult to say. Data shows that only two EA challenges are addressed. Considering the sample size this could be different in practice.

7.2 Contributions

This section describes the contributions of this research. These have been divided in contributions to theory and practice.

7.2.1 Contributions to Theory

This research contributes to theory by validating that agile principles have a positive impact on EA. In previous research agile principles were already being applied to existing or new EA frameworks, based on the assumption that this is beneficial. By validating this assumption a gap in the literature was filled.

The conceptual model that was developed is an addition to current research. Both agile software development and EA challenges are extensively described and researched. The combination of these concepts in one model is new. Due to the limitation of this research the model cannot be completely validated, but does give new insights.

These insights consist out of the significant impact of the three agile practices on EA challenges and non-significant relationships in the model. The comparison of the data gathered in this research and that of previous research on EA challenges contradicts each other. This contributes to the theory by showing that this area is under-researched.

7.2.2 Contributions to Practice

This research has identified agile practices which are able to address EA challenges in an organization. For practitioners this offers two insights: (1) agile principles can benefit EA and more specific (2) which agile practices should be focused on to ensure benefits.

The findings support the current trend in which agile methodologies are becoming an area of focus. The findings of this research give a strong indication that this focus of practitioners is just.

Besides the positive impact of agile practices on EA challenges the research shows that numerous EA challenges are related to stakeholders. Although this is not related to the application of agile practices to EA it does give useful information to practitioners. Apparently focus on stakeholders is lacking.

7.3 Limitations

The limitations of this research come forth from the methods used. Uncovering the potential effect of agile methodologies on EA was done by decomposing both concepts into smaller parts which could be measured with the use of a survey. By decomposing both concepts the interrelated relationships become less important. This combined with the sample size limits the conclusions that can be drawn.

The approach to gathering data was not chosen at random, but was driven by the fact that the amount of organizations that employ an agile approach to EA development are limited. Decomposing the phenomena enabled the targeting of more organizations. They did not have to explicitly follow an agile approach to EA development. Choosing a different method of data collection, e.g. use cases or simulations, might deliver different insights.

The data lacks information on how EA is used with an organization: the scope (e.g. organizations wide or project specific) and the purpose (e.g. controlling or driving transformations). Suppose that an EA team is in place to approve projects and ensure that projects stay within scope and at a different organization the EA team is responsible for translating business goals to IT projects. Both teams are likely to have different approaches to EA development. With the data used for this research, this distinction cannot be made

7.4 Future Research

This research provides a number of directions for future research. The requirements formulated based upon the surveys could be used to devise a conceptual framework for an agile approach to EA. This would enable the possibility to continue and finalize the design cycle.

As already suggested in the chapter on the development of the conceptual model it is kept as basic as possible. Extending the model with moderating variables might increase the exploratory power. Possible variables which have an impact are the maturity or EA framework that is currently being used. These could be incorporated in future research.

The combination of agile and EA has two possible combinations. One is applying an agile methodology to the development of EA, as done with this research. The other is the agility of the architecture, achieved by techniques such as service-oriented architecture (SOA), this has been the topic of previous research. Researching the relationship between these two might be interesting.

Appendix A

Questionnaire Development

This appendix gives an overview of the questionnaire used for data gathering. The questionnaire was pre-tested, which resulted in changes. If a question was changed, both the new and the old question are given with a reason as to why they were changed.

For the questionnaire a five-point scale was used with the possibility of answering not applicable or unknown. The scale is given below.

o	o	o	o	o	o
strongly agree	agree	neutral	disagree	strongly disagree	not applicable or unknown

Five-point scale used for interviews

A.1 Problems Perceived Regarding EA

The following questions are about problems perceived in the organization regarding EA. The statements are based upon problems commonly seen in organizations. Please try to answer these questions such that they reflect the current situation as close as possible. The answers serve as a baseline for determining the possible impact of an agile approach.

General questions

1.1 *Old:* Enterprise architecture is responsible for lower IT costs.

New: EA results in lower IT costs.

Reason: phrasing and abbreviation.

1.2 *Old:* Enterprise architecture contributes to reducing complexity in the organization.
New: EA contributes to reducing complexity in the organization.
Reason: abbreviation.

1.3 *Old:* Enterprise architecture enables faster system development.
New: EA enables better alignment between business and IT.
Reason: ambiguous.

Environment

2.1 *Old:* The EA team is unable to keep up with current software
New: The EA team* is able to keep up with current release cycles in the organization.
*group of people responsible for the architecture
Reason: negative phrasing.

2.2 *Old:* Changes in business demands are moving too fast for the EA team.
New: The EA team is able to keep up with changes in the business demands.
Reason: negative phrasing

2.3 *Old:* Results or deliverables from EA projects are too late for the organization.
New: EA artifacts* are delivered on time to be useful for projects.
*deliverable or product from the architect(s)
Reason: negative phrasing and terminology.

2.4 *Old:* Business or IT transformations have begun before results or deliverables from the EA team are available.
New: EA artifacts are mostly on time for business or IT transformations.
Reason: negative phrasing and terminology

Stakeholders

3.1 *Old:* Conveying EA artifacts* to stakeholders takes considerable effort.
*deliverable or product from the architect(s)
New: Conveying EA artifacts to stakeholders requires little effort.
Reason: negative phrasing.

3.2 *Old:* Different backgrounds of stakeholders create a barrier for communication.
New: Communication with stakeholders, even those with a different background, is effortless.
Reason: negative phrasing.

3.3 *Old:* Retrieving information necessary for EA is difficult due to unavailable stakeholders.
New: Stakeholders are usually available when information from them is required.
Reason: negative phrasing.

3.4 *Old:* Stakeholders are reluctant in providing necessary information.
New: Stakeholders are forthcoming in providing necessary information.
Reason: negative phrasing

3.5 The policies and procedures in place for EA are adequate.

3.6 Roles and responsibilities of EA are clear for the stakeholders.

Organization size

4.1 *Old:* The modeling of the EA is complicated due to the size of the organization.
New: The size of the organization does not pose any problem when modeling the EA.
Reason: negative phrasing.

4.2 *Old:* The modeling of EA is complicated due to the amount of different architectural views.
New: The amount of different architecture views* is manageable.
*representation of the architecture which is meaningful to a stakeholder
Reason: negative phrasing.

4.3 *Old:* During the course of EA projects it is determined that the initial scope was poorly chosen.
New: The initial scope set at the beginning of EA development* is rarely changed during the course of development.
*all activities and projects that are executed to realize EA artifacts (transformations or change that follow from these activities or projects are not included)
Reason: negative phrasing and terminology.

4.4 *Old:* Determining the scope at the start of the project is difficult.
New: etermining the scope at the start of EA development is effortless.
Reason: negative phrasing and terminology

Deliverables

5.1 *Old:* Stakeholders are unclear on what is expected from the EA team.
New: Stakeholders are clear on what is expected from the EA team.
Reason: negative phrasing.

5.2 *Old:* Departments have different expectations from the EA team.
New: Different departments have similar expectations from the EA team.
Reason: negative phrasing.

5.3 *Old:* Goals from the business are unclear for the EA team.
New: Goals from the business are clear for the EA team.
Reason: negative phrasing.

5.4 The EA team has relatively more focus on the IT than the business.

5.5 *Old:* The EA team produces documentation which is not used by the organization.
New: All produced EA artifacts are perceived as useful by stakeholders.
Reason: negative phrasing and terminology.

5.6 *Old:* Documentation does not meet the needs of stakeholders.
New: EA artifacts meet the needs of the stakeholders.
Reason: negative phrasing and terminology.

A.2 Agility of EA Development

The following questions are about the agility of enterprise architecture development. The context is of importance here: please answer the questions in light of the development or planning process of enterprise architecture.

General questions

6.1 Requirements are fixed at the beginning of the process.

6.2 Each step and/or stage in EA projects and/or processes are documented extensively.

6.3 Steps and/or stages are planned in detail early on in the process and/or project regarding EA.

Continuous delivery

7.1 An iterative* approach is followed regarding EA development.
*development is done in cycles: each cycle delivering a useable product which is improved upon with each cycle

7.2 An incremental* approach is followed regarding EA development.
*opposed to a monolithic approach, product is not developed at once but in chunks

7.3 *Old:* EA teams are capable of incorporating new requirements during development.
New: The EA team is capable of incorporating new requirements of the architecture during development.
Reason: more specific phrasing.

7.4 *Old:* Requirements are only defined at the highest level required.
New: Requirements of the architecture are only defined at the highest level required.
Reason: more specific phrasing.

7.5 *Old:* Business and IT are both closely involved in planning regarding EA.
New: Business and IT are both closely involved in planning EA development.
Reason: more specific phrasing.

7.6 The workload can be described as being equal over time.

Collaboration

8.1 At various points during EA development the business is involved.

8.2 The business regularly supplies the EA team with feedback.

8.3 Roles and responsibilities are determined by the team members themselves.

8.4 Team members assign themselves work.

8.5 Meetings are organized to reflect on work done.

8.6 Feedback is an inherent part of the process.

Simplicity

9.1 *Old:* The amount of EA documentation is kept to a minimum.
New: The amount of EA artifacts is kept to a minimum.
Reason: terminology.

9.2 Communication with stakeholders is primarily done face-to-face.

9.3 *Old:* All artifacts produced are used by the organization.
New: All EA artifacts produced are used by the organization.
Reason: more specific phrasing.

9.4 *Old:* Achieving transformations would be possible with less documentation and/or artifacts.
New: Achieving transformations would be possible with less EA artifacts.
Reason: terminology.

9.5 On a daily basis progress on EA development is discussed.

9.6 Progress on EA development is closely tracked.

Appendix B

Data Transformation

B.1 Reliability of Measures

Reliability of the measures on the survey has been determined with the use of Cronbach's Alpha. Table B.1 contains the results. Some measures were not reliable, these measures were addressed, this is explained below.

Items	**Cronbach's Alpha**	**Remark**
2.2 & 2.2	0,783	Although the value is on the low side, the predictor is kept.
2.3 & 2.4	0,883	
3.1 & 3.2	0,742	
3.3 & 3.4	0,719	
3.5 & 3.6	0,718	
4.1 & 4.2	0,562	
5.1 & 5.2	0,688	
5.3 & 5.4	0,818	
5.5 & 5.6	0,448	Measure is unreliable. Question 5.4 was reversed coded which is likely the cause. The choice was made to remove question 5.4, and retain question 5.3 as a single measure.
7.1 & 7.2	0,777	
7.3 & 7.4	0,338	
7.5 & 7.6	0,051	Questions 7.3, 7.4, 7.5 and 7.6 have been combined in one measure. Analysis indicates that they are measuring the same underlying concept.
8.1 & 8.2	0,67	
8.3 & 8.4	0,837	
8.5 & 8.6	0,754	
9.1 & 9.2	0,532	Although the value is on the low side it is kept.
9.3 & 9.4	0,168	Measure is unreliable. Question 9.4 was reversed coded which is likely the cause. The choice was made to remove question 9.4, and retain question 9.3 as a single measure.
9.5 & 9.6	0,801	

Table B.1: Reliability of measures

Appendix C

Structural Equation Modeling

The inner and outer VIF values are given in table C.1 and C.2. These values are all below the threshold of 3.3. This indicates that no multicollinearity is present in the model [49]. This is of importance because the constructs are formative, these values are strong indication that the constructs are being measured correctly.

Construct	**VIF**
Collaboration	1,111
Continuous Delivery	1,041
Deliverables	1,554
Environment	2,102
Organization Size	2,146
Parsimony	1,118
Stakeholders	1,328

TABLE C.1: Inner VIF values

Table C.3 contains the outer weights of the path between the variables and constructs. Considering the p-value of numerous weights, the model is unreliable. This could be a reflection of the current situation, or just a distorted image due to the small sample size.

Figure C.1 shows the path coefficients the R square value for both 2nd order constructs. The R squared values are far below 0.64 which indicates that the formative constructs insufficiently contribute to both agile and EA [49]. This value determined by the reflective measures which were used.

Measure	VIF
A11	1,553
A12	1,409
A13	1,365
A21	1,521
A23	1,521
A31	1,329
A33	1,181
A35	1,348
A41	1,134
A43	1,134
A51	1,526
A53	1,800
A55	1,567
B61	1,246
B63	1,246
B71	1,322
B73	1,322
B81	1,329
B83	1,057
B85	1,266
B91	1,474
B93	1,487
B95	1,222

TABLE C.2: Outer VIF values

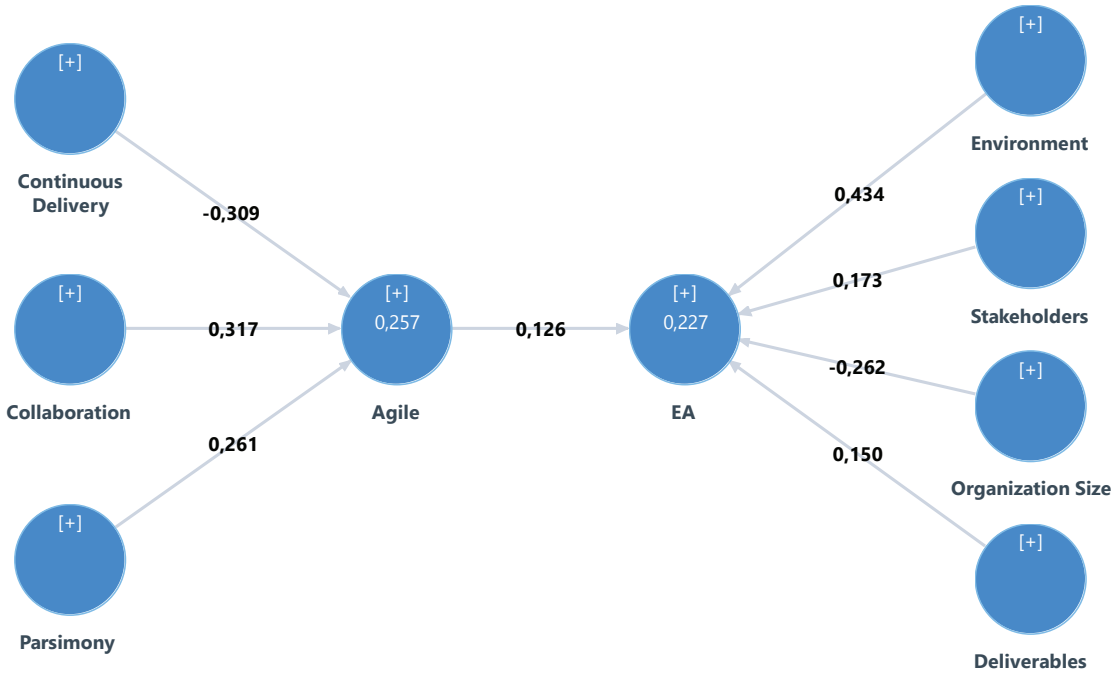

FIGURE C.1: Path loadings

Path	**P-value**
A11 <- EA	0,05
A12 <- EA	0,02
A13 <- EA	0,35
A21 ->Environment	0,06
A23 ->Environment	0,99
A31 ->Stakeholders	0,18
A33 ->Stakeholders	0,39
A35 ->Stakeholders	0,32
A41 ->Organization Size	0,93
A43 ->Organization Size	0,10
A51 ->Deliverables	0,65
A53 ->Deliverables	0,96
A55 ->Deliverables	0,17
B61 <- Agile	0,18
B63 <- Agile	0,02
B71 ->Continuous Delivery	0,13
B73 ->Continuous Delivery	0,69
B81 ->Collaboration	0,21
B83 ->Collaboration	0,67
B85 ->Collaboration	0,05
B91 ->Parsimony	0,41
B93 ->Parsimony	0,12
B95 ->Parsimony	0,49

Table C.3: Significance of paths

Appendix D

Regression Results

D.1 Checking Assumptions

D.1.1 Agile Practices Predictors

In top-right of figure D.1 a histogram of the standardized residual values is given. The histogram has a normal distribution. The P-P plot in the top-left of figure D.1 shows dots plotted closely among a straight line. Both figures combined indicate normality [54].

In the bottom of figure D.1 a plot of the standardized residuals against the the standardized predicted values. No pattern is visible in the plot. The dots are randomly distributed which indicates homoscedasticity and linearity [54].

In figure D.2 in the top the VIF values are given. These implicate that no perfect multicollinearity is present.

In figure D.2 on the bottom the Durbin-Watson value as calculated by SPSS is given. The value is between 1 and 3, which indicates that the errors in the model are independent [54].

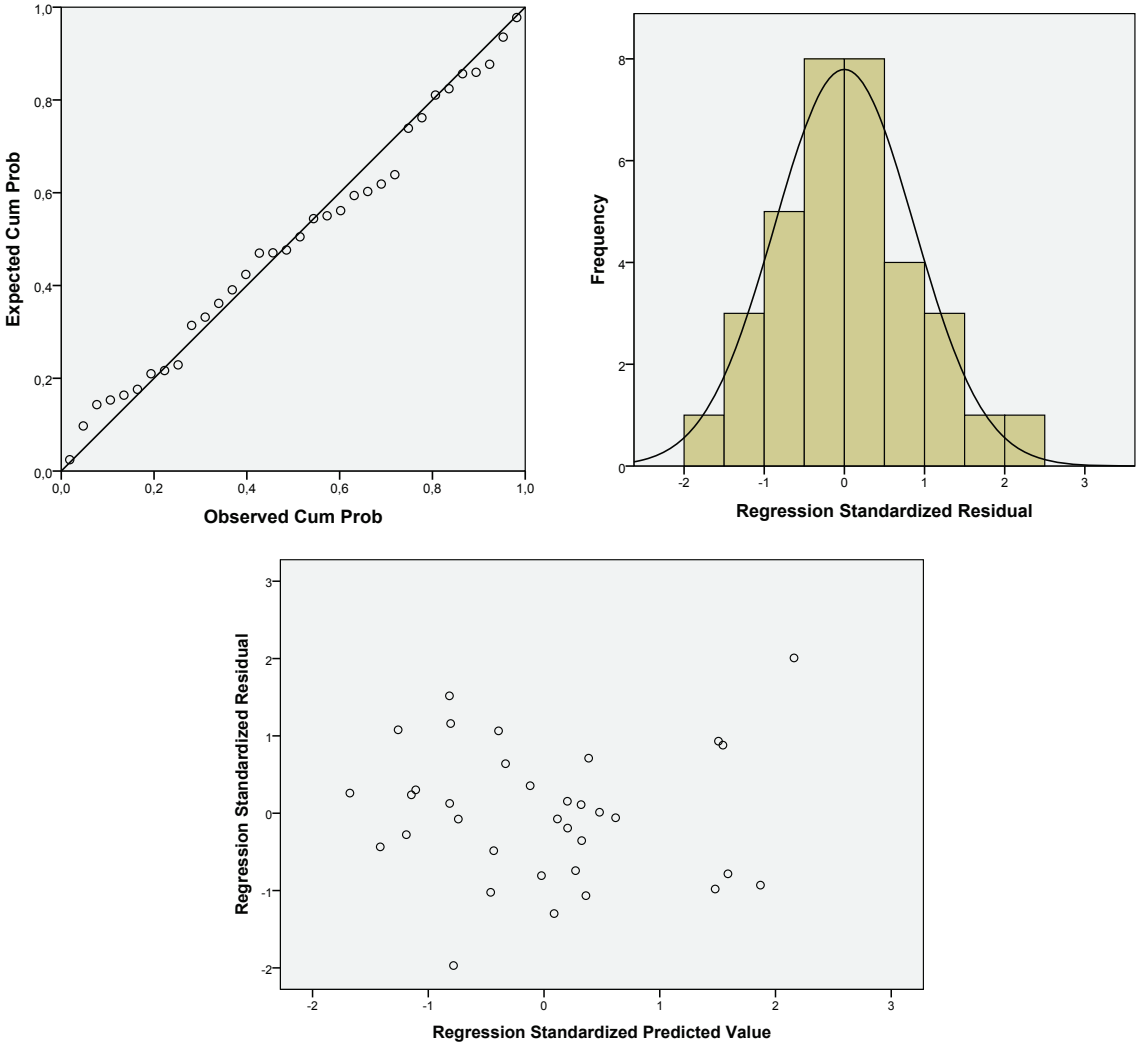

FIGURE D.1: Visual overview for checking assumptions

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	VIF
		B	Std. Error	Beta			
1	(Constant)	,102	,557		,183	,856	
	B71	,082	,097	,117	,845	,406	1,609
	B73	,252	,155	,242	1,620	,118	1,875
	B81	-,019	,097	-,027	-,199	,844	1,549
	B83	,057	,093	,076	,614	,545	1,277
	B85	,178	,105	,235	1,703	,101	1,596
	B91	,018	,107	,025	,170	,866	1,793
	B93	,276	,087	,464	3,173	,004	1,796
	B95	,069	,104	,084	,660	,515	1,373

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	,838[a]	,703	,607	,43596	1,514

FIGURE D.2: Collinearity and Independence

D.1.2 EA Challenges Predictors

In top-right of figure D.3 a histogram of the standardized residual value is given. The histogram has a normal distribution. The P-Plot in figure D.3 on top-left shows dots plotted closely among a straight line. Both figures combined indicate normality [54].

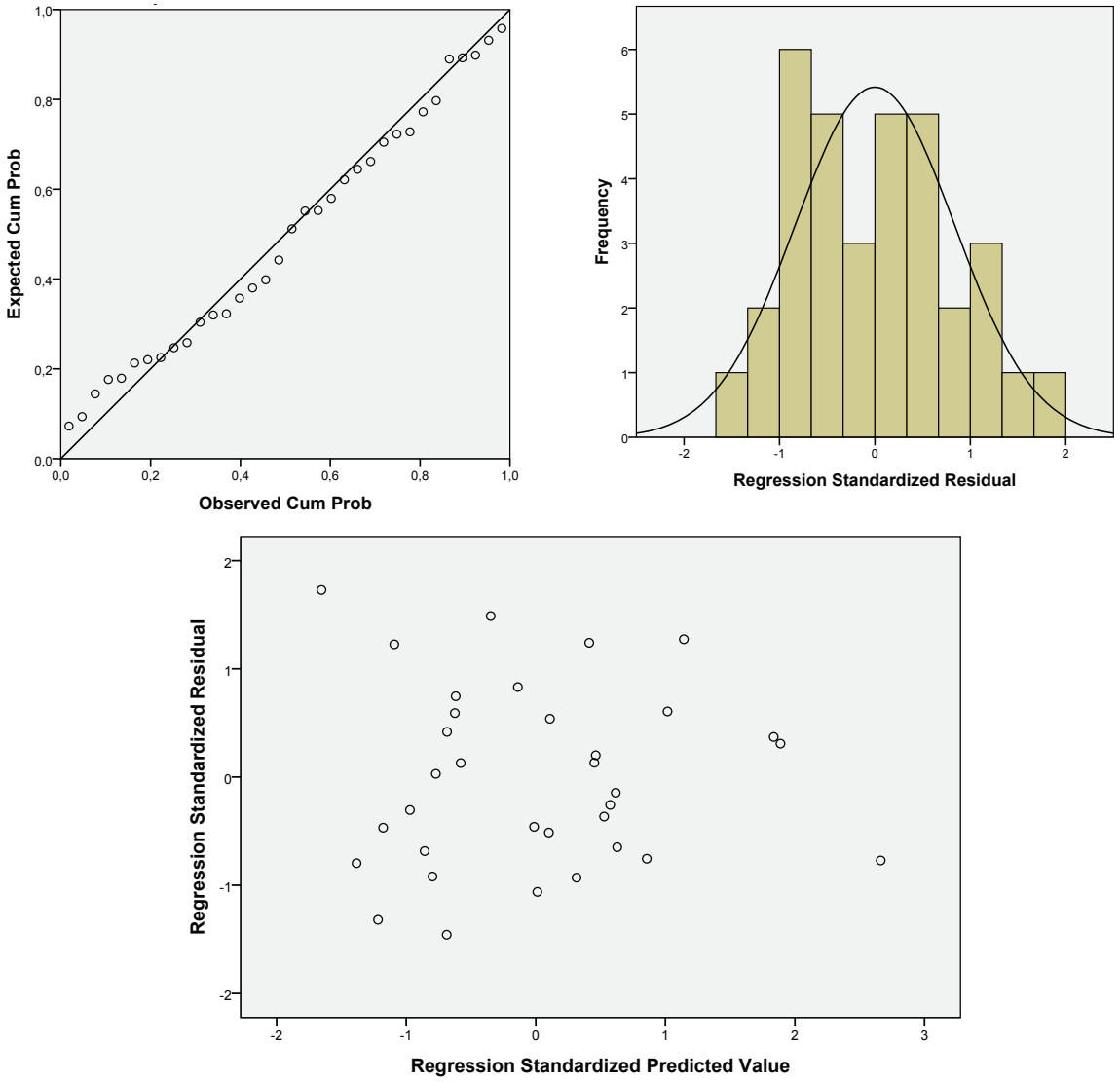

FIGURE D.3: Visual overview for checking assumptions

In the bottom figure D.3 a plot of the standardized residuals against the standardized predicted values is given. No pattern is visible in the plot. The dots are randomly distributed which indicates homoscedasticity and linearity [54].

In figure D.4 on the top the VIF values are given. These implicate that no perfect multicollinearity is present [54].

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	VIF
		B	Std. Error	Beta			
1	(Constant)	1,454	,308		4,728	,000	
	A21	,142	,131	,283	1,088	,288	4,706
	A23	,017	,098	,036	,178	,860	2,888
	A31	-,027	,153	-,036	-,179	,860	2,890
	A33	,021	,103	,038	,201	,843	2,420
	A35	,046	,093	,081	,496	,625	1,845
	A41	,135	,095	,264	1,416	,170	2,404
	A43	,007	,102	,014	,072	,943	2,641
	A51	,080	,110	,159	,727	,475	3,324
	A53	,040	,107	,089	,375	,711	3,908
	A55	,120	,087	,234	1,374	,183	2,010

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	,817[a]	,668	,524	,38274	2,070

FIGURE D.4: Collinearity and Independence

In figure D.4 on the bottom side the Durbin-Watson value as calculated by SPPS is given. The value is between 1 and 3, which indicates that the errors in the model are independent [54].

D.2 Regression results

In figure D.2 and D.4 in the top table the results of the regression are available.

Appendix E

Correlation Results

The matrix in figure E.1 contains the Spearman correlation coefficient for the significant agile practices and all the EA challenges.

			B73	B85	B93
Spearman's rho	B73	Correlation Coefficient	1,000	,062	,371
		Sig. (2-tailed)	.	,729	,031
		N	34	34	34
	B85	Correlation Coefficient	,062	1,000	,385
		Sig. (2-tailed)	,729	.	,024
		N	34	34	34
	B93	Correlation Coefficient	,371	,385	1,000
		Sig. (2-tailed)	,031	,024	.
		N	34	34	34
	A21	Correlation Coefficient	,526	,342	,694
		Sig. (2-tailed)	,001	,048	,000
		N	34	34	34
	A23	Correlation Coefficient	,239	,511	,424
		Sig. (2-tailed)	,174	,002	,012
		N	34	34	34
	A31	Correlation Coefficient	,528	-,181	,320
		Sig. (2-tailed)	,001	,306	,065
		N	34	34	34
	A33	Correlation Coefficient	,339	,215	,171
		Sig. (2-tailed)	,050	,222	,335
		N	34	34	34
	A35	Correlation Coefficient	,362	,163	,516
		Sig. (2-tailed)	,036	,357	,002
		N	34	34	34
	A41	Correlation Coefficient	,410	,384	,240
		Sig. (2-tailed)	,016	,025	,171
		N	34	34	34
	A43	Correlation Coefficient	,381	,290	,630
		Sig. (2-tailed)	,026	,096	,000
		N	34	34	34
	A51	Correlation Coefficient	,251	,215	,530
		Sig. (2-tailed)	,152	,223	,001
		N	34	34	34
	A53	Correlation Coefficient	,321	,517	,466
		Sig. (2-tailed)	,064	,002	,005
		N	34	34	34
	A55	Correlation Coefficient	,249	,442	,676
		Sig. (2-tailed)	,156	,009	,000
		N	34	34	34

FIGURE E.1: Correlation matrix agile practices and EA challenges

Bibliography

[1] John Zachman. A framework for information systems architecture. *IBM Systems Journal*, 26(3):276–292, 1987.

[2] Brian H. Cameron and Eric McMillian. Analyzing the current trends in enterprise architecture frameworks. *Journal of Enterprise Architecture*, 9(February):60–71, 2013.

[3] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems*. Springer US, 2009. doi: 10.1007/978-0-387-39940-9.

[4] The Open Group Architecture Framework (TOGAF). Other architectures and frameworks, 2006.

[5] The Open Group Architecture Framework (TOGAF). Architecture development method, 2011.

[6] Merriam-Webster Inc. *Merriam-Webster's collegiate dictionary*. Merriam-Webster, 2004.

[7] Brian Torrey, M Boster, Simon Liu, and R Thomas. Getting the most from your enterprise architecture. *IT Professional*, 2(4):43–51, 7 2000. doi: 10.1109/6294.869382.

[8] Toomas Tamm, Peter B. Seddon, Graeme Shanks, and Peter Reynolds. How does enterprises architecture add value to organisations? *Communications of the association for information systems*, 28(1):141–168, 2011.

[9] Carsten Lucke, Sascha Krell, and Ulrike Lechner. Critical issues in enterprise architecting a literature review. *16th Americas Conference on Information Systems (AMCIS) 2010*, pages 1–11, 2010.

[10] IEEE. Iso/iec/ieee 42010:2011: Systems and software engineering - architecture description. Technical report, 2011.

[11] Gartner. Gartner it glossary - enterprise architecture (ea), 2015.

[12] Asif Qumer and Brian Henderson-Sellers. Measuring agility and adoptability of agile methods : a 4-dimensional analytical tool. *Proceedings of IADIS International Conference Applied Computing 2006*, pages 503–507, 2007.

[13] a. Qumer and B. Henderson-Sellers. An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology*, 50(4):280–295, 2008. doi: 10.1016/j.infsof.2007.02.002.

[14] I Sommerville. *Software Engineering*. Addison-Wesley, ninth edition, 2009.

[15] Martin Fowler and Jim Highsmith. The agile manifesto. *Software Development*, 9 (August):2835, 2001. doi: 10.1177/004057368303900411.

[16] Matheus Hauder, Sascha Roth, Christopher Schulz, and Florian Matthes. Agile enterprise architecture management: An analysis on the application of agile principles. *Fourth International Symposium on Business Modeling and Software Design (BMSD)*, page 10, 2014.

[17] Asif Qumer Gill, Stephen Smith, Ghassan Beydoun, and Vijayan Sugumaran. Agile enterprise architecture: A case of a cloud technology-enabled government enterprise transformation. In *18th Pacific Asia Conference on Information Systems*, 2014.

[18] Hugo Ramos and André Vasconcelos. extreme enterprise architecture planning. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 1417–1419, 2014. doi: 10.1145/2554850.2555130.

[19] Tore Dybå and Torgeir Dingsøyr. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50:833–859, 2008. doi: 10.1016/j.infsof.2008.01.006.

[20] Matheus Hauder, Sascha Roth, and Florian Matthes. Organizational factors influencing enterprise architecture management challenges. *21st European Conference on Information Systems (ECIS)*, 2013.

[21] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77, 2007. doi: 10.2307/40398896.

[22] Frank J. Armour, Stephen H. Kaisler, and Simon Y. Liu. Building an enterprise architecture step by step. *IT Professional*, 1(4):31–39, 7 1999. doi: 10.1109/6294.781623.

[23] Frank J. Armour, Stephen H. Kaisler, and Simon Y. Lui. A big-picture look at enterprise architectures. *IT Professional*, 1(1):35–42, 1 1999. doi: 10.1109/6294.774792.

[24] Carsten Lucke, Marco Bürger, Thomas Diefenbach, Jan Freter, and Ulrike Lechner. Categories of enterprise architecting issues an empirical investigation based on expert interviews. *Multikonferenz Wirtschaftsinformatik*, pages 999–1010, 2012.

[25] J. Alberto Espinosa and Wai Fong Boh. Coordination and governance in geographically distributed enterprise architecting: An empirical research design. *Proceedings of the 42nd Annual Hawaii International Conference on System Sciences, HICSS*, pages 1–10, 2009. doi: 10.1109/HICSS.2009.133.

[26] Ville Seppänen. *From Problems to Critical Success Factors of Enterprise Architecture Adoption*. PhD thesis, 2014.

[27] Betsy Burton. Enterprise architecture seminar workshop results : Top ea challenges. Technical Report January, 2010.

[28] Steven Spewak and Michael Tiemann. Updating the enterprise architecture planning model. *Journal of Enterprise Architecture*, 2(May):11–19, 2006.

[29] Stephen H. Kaisler, Frank Armour, and Michael Valivullah. Enterprise architecting: Critical problems. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 00(C):1–10, 2005. doi: 10.1109/HICSS.2005.241.

[30] Scott Ambler. Agile enterprise architecture, 2009.

[31] Greg Smith and Ahmed Sidky. *Becoming Agile in an Imperfect World*.

[32] Shvetha Soundararajan, J D Arthur, and O Balci. A methodology for assessing agile software development approaches. *Agile Conference (AGILE), 2012*, page 68, 8 2011. doi: 10.1109/Agile.2012.24.

[33] Shvetha Soundararajan. *A Methodology for assessing Agile Software Development Approaches*. PhD thesis, 2011.

[34] Luigi Buglione. Light maturity models (lmm): An agile application. In *Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement*, Profes '11, pages 57–61. ACM, 2011. doi: 10.1145/2181101.2181115.

[35] Noura Abbas, Andrew M. Gravell, and Gary B. Wills. Using factor analysis to generate clusters of agile practices (a guide for agile process improvement). *Proceedings - 2010 Agile Conference, AGILE 2010*, pages 11–20, 2010. doi: 10.1109/AGILE.2010.15.

[36] Laurie Williams, Kenny Rubin, and Mike Cohn. Driving process improvement via comparative agility assessment. *Proceedings - 2010 Agile Conference, AGILE 2010*, pages 3–10, 2010. doi: 10.1109/AGILE.2010.12.

[37] Granville Miller. The characteristics of agile software processes. *International Conferences and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS)*, pages 10–12, 2001.

[38] Samireh Jalali and Claes Wohlin. Agile practices in global software engineering - a systematic map. *Proceedings - 5th International Conference on Global Software Engineering, ICGSE 2010*, pages 45–54, 2010. doi: 10.1109/ICGSE.2010.14.

[39] Lucas Layman, Laurie Williams, and Lynn Cunningham. Motivations and measurements in an agile case study. *Journal of Systems Architecture*, 52(11):654–667, 2006. doi: 10.1016/j.sysarc.2006.06.009.

[40] W W Royce. Managing the development of large software systems. *IEEE WESCON (Reprinted in Proceedings Ninth International Conference on Software Engineering.)*, (August):1–9, 1970.

[41] Robert K. Wysocki. *Effective Software Project Management*. John Wiley & Sons, Inc., 3 2006.

[42] Daniel J. Fernandez and John D. Fernandez. Agile project management - agilism versus traditional approaches. *Journal of Computer Information Systems*, 49:10–17, 2008. doi: Article.

[43] Sebastian Hanschke, Jan Ernsting, and Herbert Kuchen. Integrating agile software development and enterprise architecture management. In *48th Hawaii International Conference on System Sciences*, pages 4099–4108, 2015. doi: 10.1109/HICSS.2015.492.

[44] Sabine Buckl, Florian Matthes, Ivan Monahov, Sascha Roth, Christopher Schulz, and Christian M. Schweda. Towards an agile design of the enterprise architecture management function. *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC*, pages 322–329, 2011. doi: 10.1109/EDOCW.2011.33.

[45] Sascha Roth, Marin Zec, and Florian Matthes. Enterprise architecture visualization tool survey 2014. Technical report, 2014.

[46] M. Pulkkinen and a. Hirvonen. Ea planning, development and management process for agile enterprise development. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 00(C):1–10, 2005. doi: 10.1109/HICSS.2005.220.

[47] Chris Britton and Peter Bye. *IT architectures and middleware: strategies for building large, integrated systems*. Pearson Education, 2004.

[48] Dean Leffingwell. Safe: Scaled agile framework, 2013.

[49] Joseph F Hair Jr, G Tomas M Hult, Christian Ringle, and Marko Sarstedt. *A primer on partial least squares structural equation modeling (PLS-SEM)*. Sage Publications, 2013.

[50] Scott B Mackenzie, Philip M Podsakoff, and Nathan P Podsakoff. Construct measurement and validation procedures in mis and behavioral research: Integrating new and existing techniques. *MIS Quarterly*, 35(2):293–334, 2011.

[51] Ronald T. Cenfetelli and Geneviève Bassellier. Interpretation of formative measurement in information systems research. *MIS Quarterly*, 33(4):689–707, 2009.

[52] Walker Royce. Improving software economics. *IBM Corporation Software Group*, (May), 2009.

[53] Christian M. Ringle, Sven Wende, and Jan-Michael Becker. Smartpls 3.

[54] Andy Field. *Discovering Statistics using SPSS*. Sage Publications, third edition, 5 2009.