The Pennsylvania State University

The Graduate School

College of Information Sciences and Technology

ADOPTION OF AGILE ENTERPRISE ARCHITECTURE IN LARGE ORGANIZATION: A CASE STUDY

A Thesis in

Information Science and Technology

by

Manoj Velumani

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2017

The thesis of Manoj Velumani was reviewed and approved* by the following:

Rosalie Ocker
Senior Lecturer of Information Science & Technology
Thesis Advisor

Eileen Trauth
Professor of Information Science & Technology

Steven Haynes
Senior Lecturer of Information Science & Technology

David Fusco
Lecturer of Information Science & Technology

Veeresh Thummadi
Postdoctral Scholar

*Signatures are on file in the Graduate School

Adoption of Agile Enterprise Architecture in a large organization: A case study

ABSTRACT

In the past decade, agile methodologies have become widespread, and the many organizations that employ it have proved successful. To deal with the volatility of the market and to meet dynamic business needs, agile methods seem to be a pragmatic approach. Organizations are keen to scale agile methods across the enterprise to reap the benefits seen in agile teams. Agile Enterprise Architecture can be defined as a process for handling Enterprise Architecture (EA) by applying agile development principles and methods. However, there is a research gap in the adoption of agile Enterprise Architecture. In this paper, I analyze the factors that influence agile EA adoption in large organizations using a grounded theory approach. The evolution of the Enterprise Architect role during the agile EA adoption is presented based on the complex adaptive system theory. A case study is conducted in a leading transportation company which transitioned to agile Enterprise Architecture from traditional methods.

This paper focuses on agile Enterprise Architecture adoption from the Enterprise Architects' viewpoint. My analysis shows that various significant factors influence agile Enterprise Architecture adoption in a large organization. Enterprise Architects are the key personnel in the Governance of Enterprise Architecture and deliver essential business value to an organization. My findings show the changing role of Enterprise Architects when agile Enterprise Architecture is adopted in a large organization. Furthermore, complex adaptive system theory assists in explaining this phenomenon.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Enterprise Architecture is a set of principles, models, and methods at an organizational level that is used in the realization of the enterprise's business goals. Enterprise Architecture (EA) offers a holistic view of the company and serves as a blueprint for an organization's current and future environments (Jonkers 2006). Enterprise architecture concentrates on planning and strategy, and includes building designs for business, infrastructure, and data to provide a solution to business problems (Fallmyr 2014). There are several traditional EA frameworks in use today such as TOGAF, Gartner and FEAF.

In the past decade, agile development methodologies have seen vast, widespread success, with many organizations moving towards agile development practices from traditional approaches. Recently, industry experts have been keen to scale agile practices to their larger projects and at the organization level to enhance performance with optimal resource utilization.

An agile firm stays more flexible and nimble; it is ready to embrace openness and to change directions for new developments, reassess the past, and develop better decision-making capacities which can reap benefits for an organization. Scaling agile practices at an enterprise level poses significant challenges to an organization, team and individual levels of the hierarchy, and remains a puzzle to be solved. However, the benefits of agile Enterprise Architecture seem to outweigh the costs of the existing challenges. The focus of this paper is on the agents responsible for implementing the adoption of agile Enterprise Architecture at firms that previously used traditional Enterprise Architecture methods, and how those agents can serve as a model for future adopters in practice.

1.1 Motivation and Research Focus

Recent interests of "agility" in EA research have led to cross-pollination of ideas between agile methods, and EA processes and frameworks (Isham 2008). Agile firms sense market changes and seize the opportunity to succeed by continuously enhancing and redefining the value of the company. Agility concentrates on the firm's interaction with customers, and helps an organization to achieve its business goals (Sambamurthy 2003). Scaled Agile principles and characteristics, as defined by Laanti, equip researchers and practitioners with a preliminary set of ideologies that can assist them in building scaled agile framework solutions and gain a better understanding of the major components of an organization. (Laanti 2013). One such agile approach is the Scaled Agile Framework (SAFe) that employs layers within an iterative approach. This is one of the major motivating factors in considering an agile Enterprise Architecture framework in an organization, and can help the organization to respond faster, react to dynamic market scenarios, and assist the company to act proactively.

1.2 Research Question

In this section, I will introduce the research question which I am trying to address.

Research question: What are the factors that influence the adoption of scaled agile methods in Enterprise Architecture in a large organization?

Specifically, the focus is on

a. Enterprise Architects' perspective on the adoption of agile adoption in a large organization

   In the research question, I try to analyze the role of Enterprise Architects when an organization is adopting agile Enterprise Architecture.

2. LITERATURE REVIEW

The role of Enterprise Architects is to help the leaders of an organization create a vision for the desired future state of the firm from multiple perspectives, including business goals, operations and information systems. Enterprise architecture is a framework that helps organizations to achieve their business objectives by integrating business strategy, information and data, applications, and technology. The literature review is organized as follows: the first section is focused on the agile methods and scaled agility. The next section discusses the evolution of existing traditional EA frameworks toward agile methodologies. The following section concentrates on agile and scaled agile adoption techniques, and identifies a research gap in the existing literature. The final section introduces complex adaptive systems theory and details the involved parameters.

2.1 Agile Methodologies

In the past decade, organizations have increasingly adopted agile methods to address key software development problems, namely longer software development times, higher costs to develop, and client dissatisfaction in software performance after delivery (Fitzgerald 2006). To avoid these pitfalls, many organizations have shifted toward agile modeling and agile methodologies such as Scrum, XP, Crystal, Lean software development, in the quest for faster and better performance (Conboy 2009). Scaled Agile frameworks reportedly improve productivity and quality as an organization aims to boost their results by handling new technology changes, the constant need for innovation, the greater unpredictability of the market, and reduced transaction costs for software (Laanti 2014).

The concept of being “agile” is not entirely new. Techniques such as iterative development were proposed in the 1980’s to address the ambiguous or volatile problems that resulted from sequential refinements. Agile methods are now deemed more pragmatic and they iterate more frequently to suit to rapidly growing market needs. Recent interests of agility in EA research has led to cross pollination of ideas between agile methods and EA processes and frameworks.

When the individuals in an organization are adaptable, the organization becomes more agile and adaptive. Individuals’ ability to be flexible also maximizes organizational responsiveness. The user's ability to adapt to new technology and build their speed of competence with new technology are important in optimizing the performance of an organization by minimizing work slowdowns, individual resistance, and user dissatisfaction. Agile firms sense market changes and seize the opportunity to succeed by continuously enhancing and redefining the value of a company (Sambamurthy 2003). Organizations see benefits with dynamic capabilities such as digital options, entrepreneurial awareness, agility and strategic processes that can mediate between IT resources and an organization’s performance (Sambamurthy 2003).

To reap the benefits of agile practices, firms have recently faced the challenge of scaling agile practices from smaller teams to span across the enterprise. This has resulted in confusion, as teams practicing different approaches and methods need to interface with each other (Vaidiya 2014). Agile methods have been successful on a smaller scale, but there are concerns and challenges when attempting to scale agile practices across an organization. This introduces the research gap at the heart of agile Enterprise Architecture: what are the factors in the adoption of agile EA in large organizations?

2.2 Enterprise Architecture Evolution

The inception of Enterprise Architecture dates back to the late 1960's and early 70's in IBM with the grandfather of architecture, Dewey Walker, initiating Business System Planning (BSP). BSP is a method for designing and analyzing the information systems architecture of organizations. The evolution of BSP commenced in the 1980's by John Zachman, who was Walker's student at IBM. In 1982 Zachman mentioned the term Enterprise Architecture when he was working with IBM and BSP; it was promoted with the goal of analyzing the current system issues and opportunities, and to develop a future state for migration. The main ideology was to provide Information Systems (IS) with a blueprint for development, and to assist business people with a decision-making framework.

In 1987 Zachman published an article "A Framework for Information Systems Architecture" in *IBM Systems Journal* which explains the framework for managing an enterprise. However, he did not call it Enterprise Architecture because the term was still a foreign concept to most people. According to Zachman, his framework is primarily intended for explaining a complex entity enterprise, and his framework provides a definition and structure for the enterprise. Later, in 1990, the term *Enterprise Architecture* was formally defined; Zachman’s framework has become the oncology of Enterprise Architecture. Zachman's framework was comprised of a blueprint for defining and controlling the integration of systems and focuses on six distinct perspectives of the stakeholder groups to provide a holistic view of the enterprise: who, what, when, how, where and why.

In 1993 Enterprise Architecture Planning (EAP), given by Stephen Spewak, outlined a process for defining architectures and a plan for implementing them. EAP has its roots in IBM Business System Planning (BSP) and is designed to solve business problems with information. Several years later, in 1996, Open group framework Architecture (TOGAF) developed a high-level approach to design at four levels: business, data, application, and technology. The U.S government initiated a federal enterprise architecture (FEAF) framework to understand the of enterprise architecture in U.S federal projects in the year 1999 with the passage of Clinger-Cohen Act. Though there are many enterprise architecture frameworks in existence today, it is reported that 80% of the global companies using TOGAF framework are highly successful. With the recent need for lean agile development, modularity and flexibility, Scaled Agile Framework (SAFe) came onto the scene in 2011. In today's world, Zachman believes that Enterprise architecture is too costly and time-consuming, and that there is a need to build Enterprise architecture incrementally and iteratively (Urbaczewski 2006). The evolution of Enterprise Architecture framework is depicted in the Table-1.

Enterprise Architecture (EA) offers a holistic view of the enterprise and serves as a blueprint for an organization's current or future environment (Jonkers 2006). Enterprise Architecture is a framework that helps to achieve business objectives of an organization by integrating business strategy, information, application, and technology (Marques 2004). Today, Enterprise Architecture can be seen as a kit of processes, tools, principles and structures which are required to support organization-level activities in a consistent manner, and to enrich business operations. Enterprise architecture concentrates on planning and business strategy, and encompasses building an architecture for a business infrastructure with data to provide a solution. (Fallmyr 2014). Unfortunately, there is no universal architecture representation, method or process which is accepted by researchers and practitioners. Frameworks are built for Enterprise Architecture encompassing viewpoints such as business, information, software and technology architecture. The goal of Enterprise Architecture principles is to provide assistance in describing the current state of an organization, evaluating Enterprise Architecture elements, and proposing the future target state of the organization. EA Principles help in formulating rules, guidelines, and criteria for evaluating and designing architectures to integrate business and IT principles in an organization. IT alignment can be defined as the best fit for business vision, strategy and Information Technology services in an organization. Historically, a large amount of projects have failed due to a lack of business and IT resources, and a lack of integration. In today's dynamic environment, agility is one of the main concerns for top management in an organization. Organizational agility is defined as the ability of an organization to move quickly in new directions without breaking the existing infrastructure and pressuring the organization with risk (Tallon 2011).

Feature	**Zachman**	**TOGAF**	**FEAF**	**SAFe**
Year	1987	1995	1999	2011
Founder	John Zachman	The Open Group	Federal CIO counsel	Dean Leffingwell
Enterprise Architecture Model	When, what, why, how, where who	Business, Application, Data and Technology	Performance, Business, Application, Infrastructure, Data and Security	Scaled Agile Development Team, Program and Portfolio level
Source	IBM Business System Planning (BSP)	U.S Department of Defence (DoD) TAFIM	Clinger-Cohen Act	Agile Software Development Framework

Core	Information System Architecture (ISA)	Architecture development method (ADM)	Collaboration planning methodology (CPM)	Scaled agile principles
Features	Two-dimensional matrix Theoretical model	Concentrates on 4 EA domains Process model	Concentrates on 6 EA domains Process model	3 levels Incremental and iterative model
Critics	Unrealistic	80% of companies use TOGAF	Only Federal projects	Lacks maturity
Description	Zachman provides a holistic view of the architecture with perspectives and stakeholders but lacks process	TOGAF focuses on best practices and set policies and procedures necessary for an organization.	FEAF helps to share Federal information and allows more flexibility for federal agencies	SAFe targets agile development practices and offered more freedom to the teams
Change	Build an Enterprise architecture framework for the organization to follow	TOGAF Blended more policies, procedures and best practices to follow	FEAF targets more flexibility in federal agencies	SAFe concentrates on iterative and incremental works in teams

Table 1. Enterprise Architecture comparison chart

2.3 EA framework analysis

Table 1 summarizes the key points of the following EA framework analysis: the Zachman framework focuses on provisioning views of an enterprise but does not provide the process for building an architecture (Cameron 2013). Zachman's Framework is heavily concentrated on the following questions: what, how, where, who, when and why. The framework does not offer guidance on sequence, process, methodology or implementation, but ensures that complete view of the enterprise is established. There is no standard followed in Zachman Framework, and it has no explicit compliance rules (Urbaczewski 2006).

The TOGAF Framework focuses on communicating best practices and a set of policies and procedures necessary for an organization. TOGAF is built around the Architecture Development Method (ADM) where requirements management is central. The ADM concentrates on a cyclic, iterative process to generate EA artifacts (Singer 2015). Unlike the Zachman framework, the TOGAF framework focuses on the process and standards in business, applications, data, and technology as major domains. TOGAF's strength lies in its three levels of rules for developing good principles: decision making support and the guidance of IT resources; support for development; and implementation of architecture principles (Zarvic 2014).

The Clinger-Cohen Act of 1996 required all US Federal Agencies to implement an integrated architecture to increase the value and of, and reduce the risks from, IT projects. This led to the development of the Federal Enterprise Architecture Framework FEAF (Jonkers 2006), which was built to develop the sharing of Federal information. FEAF allows for more flexibility in the use of methods, work products, and tools used by individual federal agencies, considered a major benefit of the framework (Urbaczewski 2006).

The Gartner Framework describes EA as a continuous process focused on assessing the present architectural state, stating objectives, strategies and plans to build a future state, and managing the enterprise continuously. According to Gartner, EA is a strategy used to meet the business needs of an organization (Pereira 2004).

Based on “business-IT alignment” as a proxy for speed in EA decisions, and from the above analysis, the Gartner framework is a hallmark for speed. However, the Gartner EA framework suffers from inflexibility, poor performance, and difficulty of use measured by its operational performance metrics. TOGAF and DoDAF frameworks do well regarding performance and flexibility, but lack speed. Despite being the pioneer, the Zachman EA Framework suffers from excessive attention to detail and therefore lacks speed and performance.

2.4 Agile adoption

In this section, I discuss the various agile adoption frameworks. The Agile Software Solution Framework is designed to map the correct ways to introduce agile methods into an organization (Qumer 2008). The two core components of this model are the agile conceptual aspect model and the agile toolkit. The agile conceptual aspect model focuses on managing various aspects of agility such as people, process, product and tools. Governance is defined as the management of agile methods and practices to align with an organization’s needs and achieve the business value.

Next, the Agile Adoption and Improvement Model was proposed to consider factors such as understanding the existing knowledge about the management of agile teams in an organization, and the agile approach toward software implementation (Qumer 2008). Agile adoption and an improvement framework assist top management and general managers with the adoption of agile methodologies into an organization.

James Arthur introduced the Agile Measurement Index to help identify the agile potential of projects in an organization to determine which agile practices best fit the organization. The system component is a four-stage process designed to measure the readiness of an organization for agile adoption by assisting the organization in making a decision, and helping them to identify the best practices to adopt (Arthur 2007). The four-stage agile adoption framework primarily concentrates on the decision making process of whether to adopt agile EA. The agile adoption framework was presented to a wide variety of audiences in an agile community and the feedback was collected. The outcome was positive and the framework was seen as a first step toward an organization adopting agility.

There is a strong need for understanding the factors that affect the successful incorporation of agile methods into an organization. Agile usage refers to two measures: horizontal usage, which identifies innovation across an organization; and vertical usage, which defines the depth of organization usage. Senapathi proposed the agile usage model which is comprised of five essential factors dictating agile adoption (Senapathi 2012). In the agile usage model, innovation factors describe the creative ability of an organization. Relative advantage signifies how agile methods are beneficial in comparison to traditional methods. Compatibility is a fit between an organization's innovation and the successful implementation of new methods. Sociological factors depict the behavioral aspects of individuals such as attitude of people, experience, technical knowledge and expertise levels. Technological factors are comprised of the tools that support implementation, and the agile practices an organization will adopt.

Agile methods demand collaborative work, hence leadership and team management play an important role in adopting scaled agile in the organization. Organizational factors are also key in how the management supports changes in a team as well as shifts in organizational work culture. Finally, agile usage and effectiveness in an organization are key measures to determine the successful adoption of agile practices in a firm. The study has identified essential factors for organizational adoption of agile EA, and demonstrates that support is required to overcome resistance and compatibility issues within a firm for successful agile adoption (Senapathi 2012).

Another new dimension to agile adoption in waterfall organizations is how firms face the challenge of resistance from development team members. The model stresses the scrum master perspective and elaborates on how social intelligence can be a crucial factor in dealing with the stiff resistance at the individual and team levels in an organization. The research question here is “How do scrum masters react to challenges when faced with resistance to adopt agile practices in a waterfall organization?” The research study focus is on teams that are transitioning from waterfall to agile methodologies in an organization. I interviewed the scrum masters to understand major issues faced when handling the teams, and what techniques they devise with to manage resistance. Social Intelligence can be used a method of tackling the resistance shown by employees of an organization (Robin 2016**.** Social Intelligence can be defined as the ability to understand emotional information and use it in effective ways. Various emotional factors contribute to social intelligence; Scrum masters have the best seat to observe the various emotional factors that contribute to social intelligence, and to act accordingly. The final stage was to respond to various team members’ resistance to agile adoption with appropriate actions.

Russo et al.(2013) performed a comparative study of the agile methods of adoption factors in two organizations. The organizations identified problems with their waterfall methods and were keen to transition to agile methods. Russo et al. performed an analysis of two organizations based on the adoption factors given by the Agile Usage Model of Senapathi et al (2012). The extension of the agile usage model here focuses on the effectiveness of agility based on horizontal and vertical agile usage. Horizontal usage defines the number of projects in a firm that are in transition to agile, which is an indicates the need of agile projects. The analysis from Russo et al. strongly indicates that benefits are achieved by moving to agile methods. Sociological and organizational factors are the prime factors that contribute to the successful adoption of agility. Sociological factors focus on both the behavior of the individual and the team in terms of their agile experience, attitude towards adopting agile methods. and competence in both technical and business domains.

2.5 Scaled Agile adoption

The wide success of agile methods has spurred interest in how to scale agile methods so that they can be used across an enterprise. Various aspects of agility in large organizations are discussed by Laanti (2013) regarding strategy, business, and people. Laanti anticipated that different disciplines across large organizations would be enhanced by agile thinking when solving problems. Scaled Agile can be defined as adapting agile methods across all levels of an organization. Adaptive planning and cumulative metrics can be set by organizations and teams to plan their work using increments and steer projects companywide (Laanti 2013). The scaled agile framework builds on lean flow thinking, Toyota principles, or blue ocean strategic principles.

The Scaled Agile Framework (SAFe) is designed by Dean Leffingwell (2010) and focuses on three levels of an organization: team, program, and portfolio. The three levels are bonded together, and each level has a set of activities. The SAFe framework incorporates agile practices at all three levels and provides a pattern for scaling agility across all levels of an organization. The Team level is the lowest level and defines the size of the agile teams and the tasks for team members. The next level is the portfolio level which comprises five to twelve agile teams comprised of 50 to 125 members. The portfolio level is the top level of an organization and is responsible for satisfying management needs and delivering business outcomes (Leffingwell 2010).

The Agile Software Methodology (Scott W. Ambler 2013) is well known, and it focuses on the delivery of project solutions. Scott W. Ambler designed Disciplined Agile Delivery (DAD), which is a combination of scrum and disciplined agile solution delivery and is scalable across an organization. One of the key aspects of the DAD framework is that teams are enterprise-aware and work in an organizational ecosystem. Enterprise awareness helps to reduce the overall delivery time of a product. Working closely with enterprise individuals helps to ensure that work proceeds in the right direction.

Individual awareness helps groups of people in an enterprise. gain new skills, insights, and experience. Team awareness helps to focus on how a team learns, achieves and improves together. People are motivated to achieve an organization's goals; enterprise awareness is a key to focuings on the higher goals. Lastly, community awareness helps to give back to the community by sharing knowledge and helping others, not just within the organization. The DAD framework's agile full lifecycle makes people enterprise-aware and helps to work with enterprise individuals to achieve]their targets. (Ambler 2013)

Empirical research on large-scale agile development is lacking, and organizations are keen to learn how to scale agile methods across enterprise (i.e., high) levels in the firm. However, scaling agile methods is complex, and there are many considerations regarding how agile practices fit largescale projects (Reifer 2003). There are notable integration and testing issues in a dispersed environment. Dingsoyr (2008) proposed a set of principles to overcome such challenges when moving an enterprise towards largescale agile adpotion. Architecture, inter-team co-ordination, portfolio management and scaling are the key areas for large-scale agile development (Dingsoyr 2008).

2.6 Organizational adaptation

Organization change is inevitable, and firms need to cope with environmental changes and structure an organization accordingly. Organizations are continually changing and cannot be arbitrarily controlled. The capacity of an organization to learn and adapt to change define the responsiveness and speed of the firm. (Hannan and Freeman 1984). Habits and routines can be defined as basic element of change in an organization which results from cultural, economic and environmental changes. (Knudsen 2007). Traditionally there is a misconception that inertia slows down organizational change. Inertia can be defined as the rate of change. However inertia influences the process of organizational adaptation and engenders useful organizational variation (Yi et al 2016)

Organizational adaptation happens over a long period of time with selection and adaptation driving change (Levinthal 1991). Organizational routines help stability and performance over period of time and play a major role in organization adaptation. The exploration and refinement tasks of an organization can be bring in flexibility and stability to the firm. Organizational routines serve as an enabler for change by making repetitive and routine efforts on the process. Reliability can be achieved by highly standardized organizational routines which are defined by Nelson and Winter.

Organizational routines play an important role in the adaptive dynamics of an organization and have three different facets: selection, variety and plasticity (Levinthal 2015). Organizational routines are defined by Pentland and Feldman as three approaches 1) Routines can be seen as a black box in the company 2) Routines as patterns of action and treat them in isolation 3) Understand the relationship between the components and identify parts of change (Pentland and Feldman 2005). Ostensive and performative aspects of organizational routines are discussed by the author. Organizational routines have great potential to change the behavior of an organization by enhancing the performance.

Uncertainty with new technology and its features is a major challenge for organizations. Problems posed by the new environment to the existing organization are complex and remain a major concern for the adoption of agility (Tyre 2011). Technical complexity increases the need for existing human resources to develop new skill sets and best practices, which is an ardent task. Adoption requires systematic shifts of many things including roles, relationships, and behaviors, which could lead to disruption. The greater the degree or size of process change, the greater the disruption (Tyre 2011). Innovation is in at the top of the priority list for the top management in many organizations because it produces competitive advantages. Process innovation with improved methods and optimized work process enhances the efficiency and responsiveness of an organization (Fichman).

2.7 Complex adaptive systems theory

Complex adaptive systems theory focuses on the study of management in organizations. Complex adaptive system theory is a study of the science on how complex systems can adapt to their environments to deal with various components (Vingen and Wang 2006). Complex adaptive systems reveal the evolutionary aspect in which sectors of the population adapt to environmental changes in an organization. (Hannan and Freeman 1977). Complex adaptive systems can be defined as the study of systems built around individuals who are capable of adapting when they interact with new environments (Janssen 2006)

The complex adaptive systems theory can be applied to agile methods and management. Organizations are seen as a complex system environments where independent individuals can work in self-organizing ways to produce innovative results (Highsmith and Cockburn 2001). Agile methods such as scrum, XP, and Kanban evolve in a complex adaptive system environment. Complex adaptive systems can be used to explain several practices in agile methods such as iterative development, retrospection and feedback mechanisms (Jain and Meso 2004).

Videgen and Wang discuss complex adaptive system principles and define them as, “Principle 1 Managing internal rate of change to match or exceed the relevant external change rates. Principle 2 Optimizing self-organizing, Principle 3 synchronizing concurrent exploration and exploitation”. The complex adaptive system theory principles deal with the chaos in an organization that begins to self-organize internally. Finally, the principles focus on exploiting the past and exploring the future (Vingen and Wang 2006).

Self-organizing is a key theme in the complex adaptive system theory. Multiple agents interact with each other in a complex adaptive system, and they are capable of self-organizing and evolving over time. Self-organizing can be defined as the process in which individuals adjust and make changes to their behavior to deal with internal and external environment changes (Cilliers 1998). There are few principles which define self-organizing behavior as described by Stacey 1996. The parameters for self-organizing, Stacey wrote, are “The rate of information flow throughout the system, the nature of connections among people, and diversity of cognitive schema” (Stacey 1996). Higher-level parameters help an organization to understand the conditions and opportunities that exist in a system, the connections among people across the system, and the creative and innovative ideas that emerge for people to self-organize and adapt.

Management practices are seen as catalysts for self-organizing parameters to fully function and where the new behaviors can emerge. Management practices offer individuals a platform to identify opportunities and develop ways to adjust to environmental changes (Anderson 1999). Self-organizing has no command and no control structure. There is no planning or managing required when individuals can sense environmental change and adapt to evolving needs. Complex adaptive systems emerge over time in a coherent fashion without the need for any singular entity to control them (Janssen 2006).

3.RESEARCH DESIGN

3.1 Research study methodology

I conducted a research case study on one of the leading transportation companies in North America, here named “Alpha,” to investigate the effects of Scaled Agile Framework (SAFe) and understand emerging work practices in Enterprise Architecture. In these projects, I investigated modernizing services at the company “Alpha," referred to as their “Service Modernization” initiative. The Service Modernization initiative includes service planning and freight movement with advanced technology. The new Transportation Support System (TSS) will enable “Alpha” to be agile in responding to changing business demands and give them immediate access to better information to support decision-making. The projects in this initiative are mission critical and are carried out to modernize outdated transportation support systems. To this end, “Alpha” had adapted agile methods such as SAFe to reduce cycle times in the implementation of Enterprise Architecture in the company. To understand these variations and the differences that agents and artifacts bring about in projects, I studied two projects of roughly similar sizes and timeframes that used the SAFe framework. The focus of my research case study is on the Enterprise Architect perspective during agile adoption in an organization. Grounded theory approach is followed in this case study.

3.2 Organization overview

“Alpha” is a transportation company over a century old that is a leader in freight services. Technology Services is the name of the IT department at “Alpha”. The Transportation Support System (TSS) is at the core of its operations. The Transportation Support System was formulated in 1992 with mainframe systems of 30 million lines of code hits and 3.6 billion queries, and has been in operation for 25 years. The existing system and the technology it used are mainframe legacy systems, which are facing extinction. The company previously had several unsuccessful modernization efforts. This time around, their motives and business needs were stronger. The company follows a traditional Enterprise Architecture approach, which is not formalized to follow any specific framework and has borrowed principles from TOGAF. During the planning process for the new system, “Alpha” realized that they not only needed to modernize the code, but they also needed to modernize the functionality and technology base of the system as a whole and embark on a new way of doing business.

3.3 Research study settings

I selected the company “Alpha” to study because it is a large organization with a history of traditional Enterprise architecture that has recently evolved to scaled Agile EA. Learning about the process of transition in a single or multiple organization can create a picture of the factors involved in adopting Agile EA, and assist in proposing a framework for its adoption. Research design components involved identifying an organization with a history of Enterprise Architecture that was willing to participate in a case study. For this study, I followed a qualitative research approach with a series of interviews conducted at various levels of the organizational hierarchy. The interviewees held positions such as Director, Enterprise Architect, Senior Architect, Manager and agile team member. The interviews are followed by data collection and analysis, and propose the findings for Agile EA adoption.

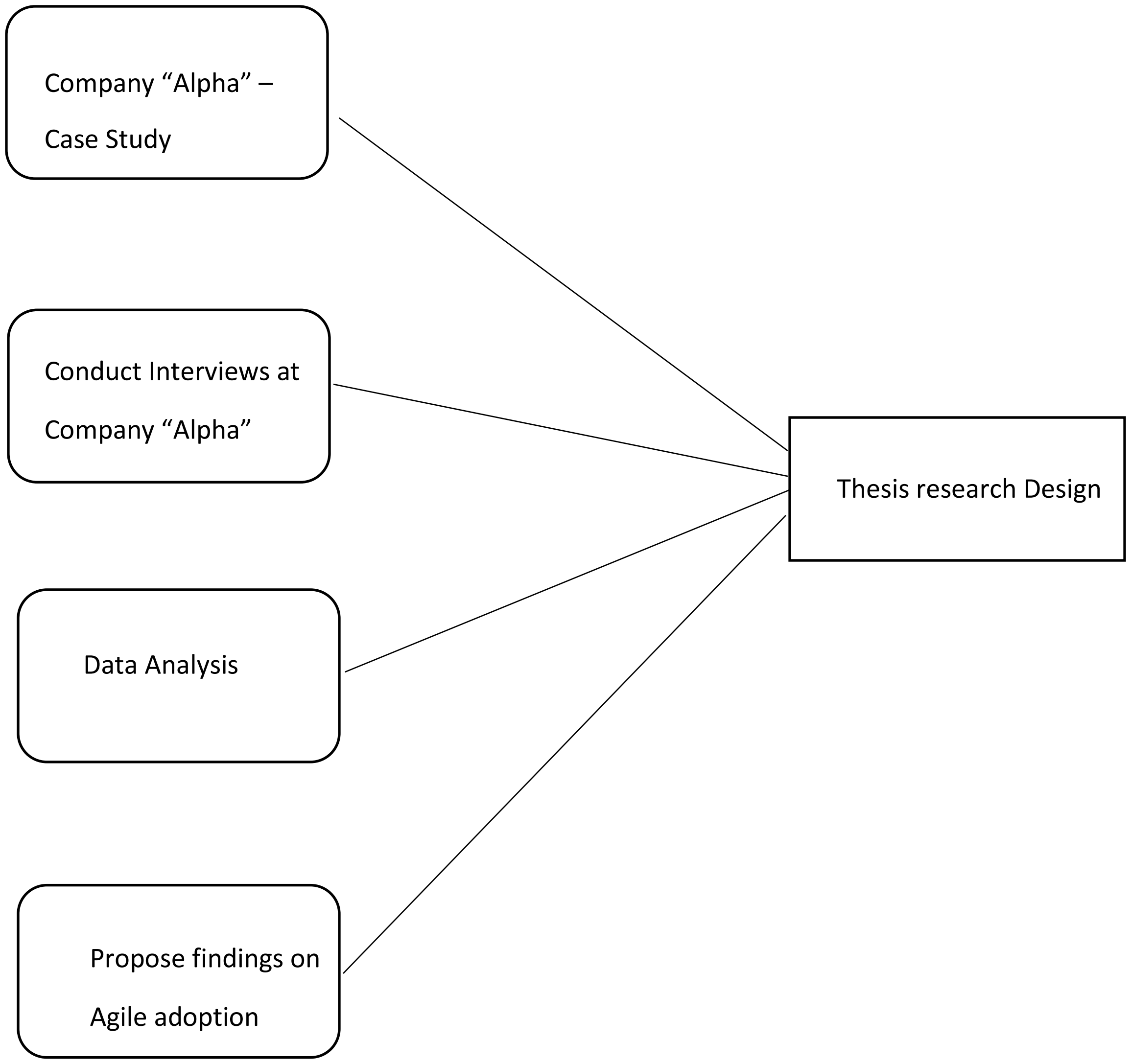
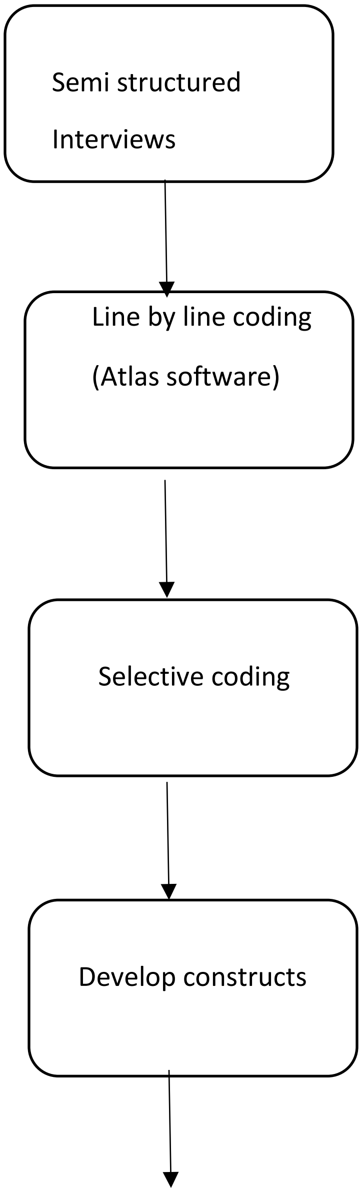


Figure 1. Research Design

3.4 Data Collection method

The grounded theory approach was followed in this research case study. The interviews were semi-structured, and the duration of the interviews was between 30 mins to 1 hour. Overall, I conducted 27 interviews as part of this research study at "Alpha". To get wide perspective of the EA transition process, participants were spread across top management positions such as

Enterprise Architects, Directors, Project Managers, Business Analysts and Development team members. The interviews were recorded and transcribed for analysis purposes. The approach for data analysis was to begin with line by line coding and then perform selective coding on the Enterprise Architect's perspective. Coding was performed using Atlas software, a preferred tool for coding analysis. The research study phases were: 1) problem formulation 2) case study design 3) open coding and data collection 4) selective coding and data collection 5) process analysis and data collection 6) theoretical coding and data collection 7) scaling up and 8) theoretical integration.



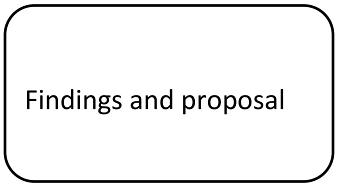


Figure 2. Data collection methodology

Interview Requirements

Phase 1

I conducted 13 interviews with directors, enterprise architects and senior architects from the top management.

Focus

The focus of phase 1 was to understand the service modernization initiative, organization change and strategies at the top level. Another aspect of the interviews with the enterprise architects was to understand their transition from traditional to agile methods.

Phase 2

In Phase 2, two project teams were selected for analysis. Each team was from a different project (Shipping Project Management and Work Force Management). A total of 14 interviews were conducted: 7 in Shipping Project Management and 7 in Work Force Management.

Focus

These interviews were conducted to gain insight into the agile teams and their work practices. The interviews with the agile teams helped me to understand the agile adoption process and how the team went through this transition phase.

The table below displays the research steps involved in this case study, and the outcomes. I followed grounded theory approach at the outset of the research to formulate the problem, gather data and perform coding analysis of the collected data. With some initial findings from the data, I then went back to the literature to relate relevant existing theories that support the findings. Complex adaptive system theory supports the theory of the Enterprise Architect's role in the evolution that emerged from the data.

Research Step	**Tasks**	**Outcomes**
Problem formulation	• Filter out domain and explore research problems and gaps. • Identify the problem from theory and practical perspectives. • Find the research gap based on the existing literature.	• Agile Enterprise Architecture is identified as the research domain. • Problem area to focus on is agile Enterprise Architecture adoption. • Learned more from the existing literature and formulated a research question.
Case study design	• Start with a research methodology. • Identify a single case study design on which to base the research. • Layout of the research steps and data collection methods.	• Grounded theory approach is selected as a research methodology approach. • The case study is to be based on an organization in transition to agile EA. • The research design steps are laid out in detail for Company "Alpha". Interview protocol is shown in the appendix section.
Data collection methods	• Start with the grounded theory approach to collect data. • Sketch out interview phases and protocols	• Conducted multiple interviews at the company "Alpha" • Completed line by line coding to analyze the interview data • Identified initial codes and devised category groupings.

	- Gather preliminary and secondary interview data and perform line by line coding.	
Selective coding	- Delimiting further coding based on emergent categories. - Perform selective coding and label the categories	- The emerged categories depicted some of the factors that influenced agile adoption. - Narrowed the categories towards agile adoption factors and Enterprise Architect perspectives shown in appendix section,
Theory building	- Start analyzing and identifying the relationship between emerging categories. - Build constructs and aggregate dimensions and arrive at a theory. - Try to relate the emerged theory with the existing theories in the similar field	- Group the categories and find relationships that depict Enterprise Architect factors. - Enterprise Architect role evolution is emerged as shown in figure 4. - The emerged theory is related to the complex adaptive system theory
Theory Integration	- Find relevant existing theory and relate to it. - Integrate the existing theory to compare the findings	- Adaptive system theory is related to the Enterprise Architect theory. - Compare the findings with the Adaptive system theory and make findings more concrete.

Table 2 Research Steps and Outcomes

4. RESEARCH FINDINGS

4.1 Findings

The findings are organized based on the two phases of conducted interviews. The data analysis was based on line by line coding and then the selective coding approach was used to address the research question. Constructs were developed based on the selective coding.

The **top management** triggered a service modernization initiative to rewrite and redesign the existing code to scale for the future. In the process, they realized that they need to modernize their transportation system as a whole to reap the benefits. The organization's management decided to take a new approach to the way they do business by changing the current process and making it more scalable and efficient.

The firm hired **e**xternal consultants to review their IT operations, architecture, and the operations of the company. The outcome of the assessment from the external agents was the recommendation to build a formal Enterprise Architecture program to succeed in modernization efforts. The current system in the firm was more of traditional EA approach with waterfall design and agile teams in development. The validation of external agents triggered the need for a strong Enterprise Architecture program in the minds of the top management.

To combat pitfalls in the existing system and to ensure profitable service modernization efforts, a **tiger team** was formulated in the organization with the goal of establishing an appropriate enterprise architecture framework. The tiger team had specialists from various internal divisions of the company with decades of experience working in the organization. The tiger team was tasked with understanding the existing architecture in detail, processing the feedback by the external agents, and **building a formal enterprise architecture framework** which could serve as the foundation for the service modernization project and support the organization's future initiatives. **Technology Services leadership** played a major role in the decision to kick off the transformation to agile EA and gave immense support to the tiger team. **Organizational necessity** was one of the prime factors that triggered the transition phase to agile teams for the organization. The company "Alpha" had one of the largest monolithic mainframe systems in place, dating to the 1990s. The maintenance of legacy systems was proving to be expensive and time consuming. There were multiple attempts in the company to modernize the legacy systems which were unsuccessful due to a lack of motivation. The "Alpha" company decided to live with the legacy mainframe systems considering the challenges and complications involved in replacing the entire system while it was in use. In 2011, the Alpha Company decided to modernize the legacy systems, a decision that was strongly driven by the leadership.

The mainframe systems were large, with 30 million lines of code that handled the load of 3.6 billion queries a day, quite the arduous task for legacy systems. The performance of the transportation systems could not be enhanced without migrating the latest technologies. The newer technology offered advanced features, and faster, lightweight transactions when compared to the large legacy systems. From the business perspective, the risk of legacy systems were also higher, with increasing user load. The transportation support system needed to be rewritten and re-designed to make it sustainable for the future. The aim of modernizing the transportation support systems wass to make the system flexible and react quickly to changing business needs. A few of the quotes below highlight the organizational necessity.

> *The core transportation system that runs the railroad is all mainframe based. We actually have 5 main frames, but we have 3 that are together as part of our core. What we call our "transportation support system." It's 10,000 MIPs that run 3.6 billion sequel queries a day. It's really a pretty transactionally intestive system. (Interviewee A)*

> *We understood if we wanted to do this properly and not just end up with the same highly coupled, highly integrated hairball that we have today…we needed to raise that up and start with it being a business-driven modernization. (Interviewee B)*

People competence is about the ability of the individuals to work together as a team to build agile methods in an organization. Agile teams are self-organizing and independent in nature, which is one of the biggest differences compared to waterfall methods. Building agile teams is the primary objective to drive projects and team forward, where the competence of individuals in the team plays a key role. One of the interesting things I observed across multiple interview sessions is the importance of agile training and education, emphasized by multiple people. Agile Education's focus is on agile training and creating awareness among team members of the agile principles and processes that are required. The following quotes indicate changes in the agile setup and the level of competence required from the team and individuals to help the organization transition smoother.

> *Well how the project has evolved…I can definitely say we've improved in terms of having a better roadmap and better planning basically and better coordination among the teams. We have different teams. We have platform teams. We have feature teams. Initially, we struggled with the coordination between the platform team and the feature teams, aligning their goals with each other. (Interviewee C)*

In 2014, the company decided to embrace agile architecture and scale it to the enterprise level. The road map of the company can be visualized in terms of program increments which stand at the top of the hierarchy. Each program increment is then split into chunks of sprint cycles and the agile teams work in sprints to deliver results. Product owners are responsible for backlog management and the agile teams follow scrum methodology to achieve project deliverables. The switch from traditional to agile methodologies is an arduous task and the company provided sufficient agile training to make the process smoother. Finally, at the lower level are the user stories which are assigned to individuals to work on during the sprint cycle. Daily scrum, sprint reviews and the retrospection process help to evolve and enhance the agile process and delivery. The Management must undergo a metamorphosis, which is the process of adopting new practices and processes in the organization to meet business needs.

Agile ingestion is about the people in the team getting to know the agile principles, receive training from the agile coaches and then trying to ingest the entire agile methodology into their work culture. Creating awareness among people for agile terminologies and helping them to put the agile methods into practice is the key outcome for successful agile education in a company. The “Alpha” company indicated the importance of agile education, which was evident for the **work culture shift** among individuals and team members of the company.

There are new roles associated with agile teams such as scrum master and product owner, and the responsibilities of individuals in the teams also changes to suit the agile nature. Learning roles and responsibilities is vital for agile team members. The work culture itself undergoes change because the agile teams are self-organizing and more independent. Agile process includes scrum meetings and sprint cycles, so the agile work culture is different and needs to be understood. After receiving the agile training and getting to know the agile work culture, the next important step is for the teams to clearly understand agility. The agile ingestion process can take time initially, but coming to terms with agile processes is essential to the success of agile teams and the organization.

> *Traditional is difficult to ... It doesn't really seem to meet the needs ... I've seen more projects fail ... I've seen different projects fail using traditional methods than I do in agile methodologies ... Sometime ... some way I’ve seen 3-4 times as many of the projects fail using the big waterfall techniques. The projects in agile have typically ended up with higher amounts of customer satisfaction and development satisfaction. (Interviewee D)*

4.2 Enterprise Architecture perspective on agile adoption

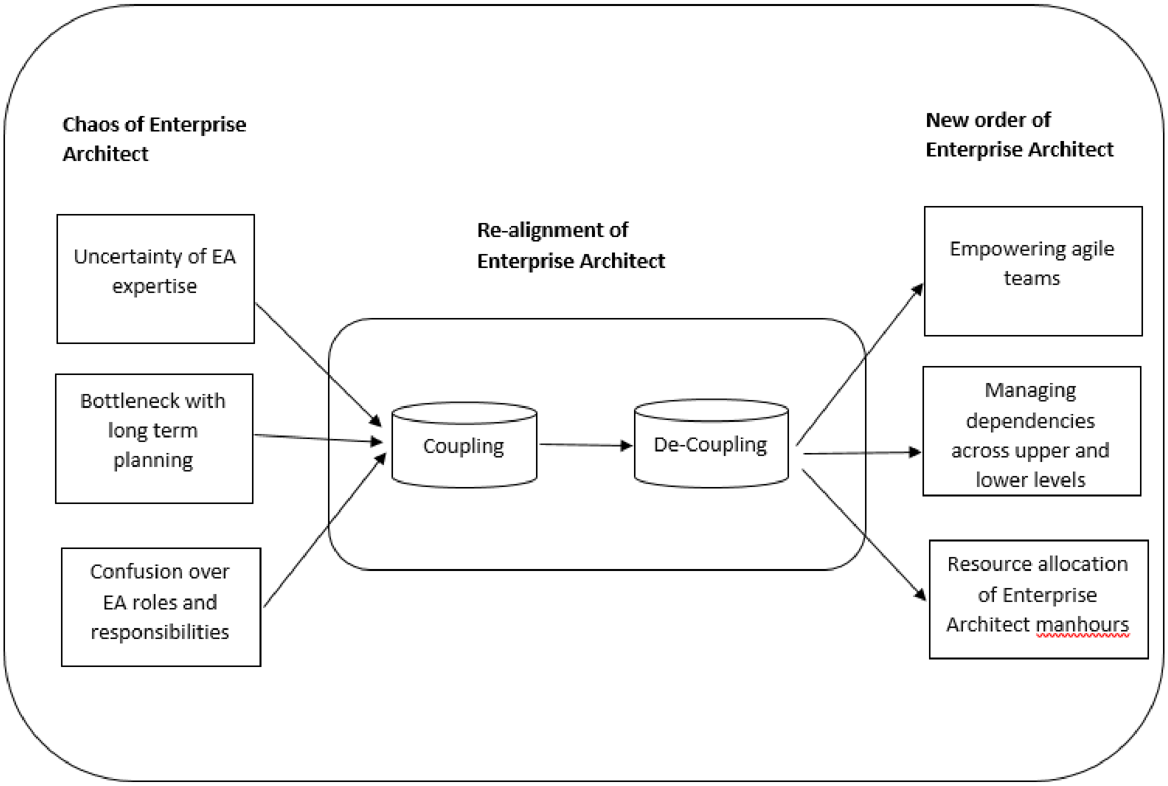

Chaos of Enterprise Architect
Uncertainty of EA expertise
Bottleneck with long term planning
Confusion over EA roles and responsibilities
Re-alignment of Enterprise Architect
Coupling
De-Coupling
New order of Enterprise Architect
Empowering agile teams
Managing dependencies across upper and lower levels
Resource allocation of Enterprise Architect manhours

Figure 3 Enterprise Architect role evolution

After undergoing initial assessments, in 2011 the "Alpha" company started the initiative to modernize with the strong support of leadership. Experienced Enterprise Architects sent out positive signals to top management in support of the service modernization effort. Slowly, the company spread the message about the need for modernization, from both the development and the business standpoint, for the organization to perform effectively. The research question's focus is on the Enterprise Architect's perspective during the scaling process of agile EA adoption. The role of the Enterprise Architects seems to have evolved during the transition phase. In this section, the focus is on the findings from the data that used the grounded theory approach. The data reveals the roles and responsibilities of Enterprise Architects and how they transitioned during the adoption phase in the organization, depicted in Figure-3, During the agile EA adoption phase, the Enterprise Architects in the organization were uncertain about their roles and responsibilities. The organization was moving from traditional waterfall methods to agile Enterprise Architecture, which involved a tremendous amount of management change. The Enterprise Architects experienced a situation in which they were unclear about whether to follow traditional or agile methods**. "Just in time architecture"** was one of the key things that emerged from the top management, which is completely a different approach from Enterprise Architecture. The organization hired few Enterprise Architects to bring more expertise to the team. Overall, **internal chaos was concentrated among the architects in the organization** who dealt with the situation, as shown by the below quotes:

> *At the same time, moving from a traditional requirements-gathering waterfall environment the way we always have been, and changing that to Agile, there's a ton of change management going on in this, helping folks not just come up to speed on, but truly understand what these are and how they function. Then marrying the two together when they are traditionally I think there's more of what you're looking for, these are ... traditionally you would see these at odds with each other.*

> *With enterprise architecture being very large, slow moving, want to understand, broad and deep across domains. How do you ... that's where we are right now is, how do we find that balance? That's what we're really working hard on right now is what is ... what we have termed as just enough, just in time. How do we measure the ... or how do we balance the desire to get product out the door, from an architecture standpoint, taking that up to a high enough level and being able to look far enough across domains, and understand it across the different domains of architecture to where we don't end up in the same place that we*

> *are today. (Interviewee E)*

> *What Vishal is referring to is that I came up with a concept of marking an item on the artifact as "just in time", which also meant we would really only create it if, and when, it was needed. Sometimes it may have been at the discretion of the architect themselves, as to whether they felt that it was needed as a work product or not. Other times it was, we may or may not need it at this point in time. We may or may not need it later. We will create that if, and when, we need it at the point in time it's needed.*
>
> *(Interviewee Top management)*

The **role of Enterprise Architects** was fluctuating, as shown in figure 2, from a resource manager who has the capability to manage and drive the team. The managerial aspects of the Enterprise Architects were expected to solve some of the problems. People were seeking technical expertise from Enterprise Architects, which is a completely different dimension to managerial aspects. The Enterprise Architects experienced a plethora of different scenarios and were expected to solve problems at different levels. The expectations were unclear; the Enterprise Architects did not understand their roles and responsibilities, which lead to confusion in the Enterprise Architect group.

The Enterprise Architects were groomed from engineers, who are different in their thinking. The Architects were expected to have a deductive reasoning mindset with a framework for long-term planning, which is evident from the quotes in the below section. The Enterprise Architects groomed from the background of engineers have a different mindset all together, which can hamper the speed of delivery. The Enterprise Architects were also struggling to find the balance between the agile and traditional approaches. The agile approach required less documentation and a shorter turnaround time for delivery. The **Enterprise Architects could also be seen as a bottleneck with a vision of long range planning**, which could negatively impact the agile team focus. However, the Enterprise Architects were given the role of transitioning to agile Enterprise Architecture in the organization.

> *In our organization we take engineers and we walk them up through the ranks into architect without taking the engineer out of the engineer. In building hopes you have architects and engineers, and they're different for a reason. Architects think big. They think innovatively. In many cases, scientist and engineers have been groomed to pinpoint. Architects use deductive reasoning. Engineers use inductive reasoning. Scientists use deductive reasoning but they use it at the detail level. When you pick an architect, you need someone who can use deductive reasoning very well. You don't generally ... It's hard to find in the ranks of the engineers. If you've got a program where you can groom someone into an architect and they're effective.*
>
> *The biggest challenge to me right now is changing the mindset of everybody. When you're only really focused on 2 weeks' worth of work or planning shorter increments of work to develop a plan for a whole year is very challenging. Our executive leadership, they still expect to see what your plan is for a whole year. Or if it's a whole project, what's your plan for 5 years. It's to me very difficult to be able to produce those types of reports when we're really focused on planning small increments of work. That's been my biggest challenge.*
>
> *(Interviewee Top management)*

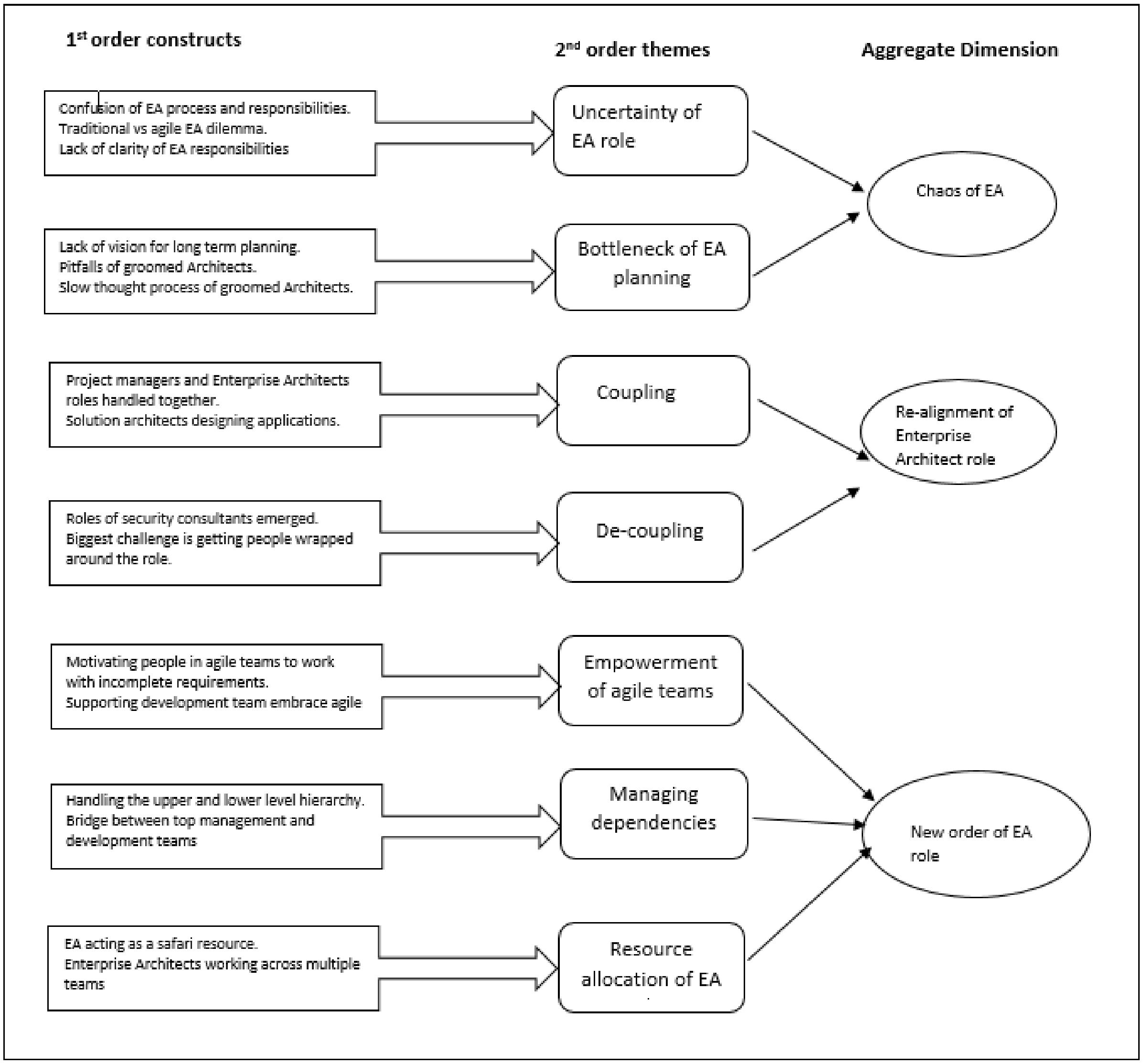

1st order constructs
2nd order themes
Aggregate Dimension
Confusion of EA process and responsibilities.
Traditional vs agile EA dilemma.
Lack of clarity of EA responsibilities
Uncertainty of EA role
Chaos of EA
Lack of vision for long term planning.
Pitfalls of groomed Architects.
Slow thought process of groomed Architects.
Bottleneck of EA planning
Project managers and Enterprise Architects roles handled together.
Solution architects designing applications.
Coupling
Re-alignment of Enterprise Architect role
Roles of security consultants emerged.
Biggest challenge is getting people wrapped around the role.
De-coupling
Motivating people in agile teams to work with incomplete requirements.
Supporting development team embrace agile
Empowerment of agile teams
Handling the upper and lower level hierarchy.
Bridge between top management and development teams
Managing dependencies
New order of EA role
EA acting as a safari resource.
Enterprise Architects working across multiple teams
Resource allocation of EA

Figure 4 Findings Enterprise Architecture perspective

To achieve the goal of transition, the **Enterprise Architect group strived to re-align** and meet the expectations set by the organization, as depicted in figure 3. The Enterprise Architect group went through a phase of self-organizing themselves and identifying roles and responsibilities for the individuals and the group. In the process, there were dual responsibilities handled by individuals; for instance, a Project Manager playing the role of Enterprise Architect that also handled management responsibilities, or an Enterprise Architect entrusted with the responsibilities of a Business Architect.

Coupling of roles and responsibilities were visible to a larger extent among the Enterprise Architect group dealing with additional responsibilities. The key for the Enterprise Architects was to re-align with the goals of the company and to help create a smooth transition process to agile Enterprise Architecture adoption in the organization. The modernization initiative of the company opened the scope for various changes in the organization. There was a tiger team formed with a group of senior architects to create a solution for the Enterprise Architecture in the organization. Here, the Enterprise architects were playing the role of designing the solutions, which is demonstrated in the interviewee quotes below:

> *As far as assigning responsibilities and making sure we're progressing, that's really on the team themselves. The teams are empowered to make those decisions. What we've had to do is really rethink what the manager does, and so what we've done is in a lot of places where it's appropriate, we've had managers now playing the role of PMs but in the same point they're no longer a resource manager. They've become an individual contributor and now they're a PM. We've had some that retained the resource manager capability but they also take on the responsibility of an enterprise solution architect.*
>
> *I think a lot of the terminology is some of the problems, too. They don't understand what they're supposed to do or how they're supposed to handle it. We're supposed to self-organize and self-decide how to do things as a team, but then we also have leadership and product owners that are trying to say, "This is how we have to do it," and, "This is how we want it done." That doesn't give the team the ability to decide some of those things.*
>
> *(Interviewee Top Management)*

Self-organizing aspects of the Enterprise Architects are witnessed here, where they evolved to meet the needs of the organization. There were internal and external factors that played a role in shaping the Enterprise Architects. The external push was from the technology emergence, with a whole new set of technologies that were pushed for a service modernization initiative. Internally, the success of the agile development teams portrayed the need for scaling agile methods across the enterprise in the organization. To meet the needs of the organization and deliver results, the Enterprise Architects coupled roles and responsibilities.

The roles of the Enterprise Architects were also decoupled into functionality based architect roles in the organization. Several of the noticeable decoupled roles of the Enterprise Architects were Enterprise solution architects who were responsible for building solutions to design and developmental problems, Business architects who were primarily involved in envisioning the roadmap for the future, Enterprise Data Architects and so on. One of the key challenges for the Enterprise Architect group was to understand the roles of people with a variety of titles and re-align them with new roles and responsibilities, as seen in the quotes below:

> *I think probably the biggest challenge from an enterprise architecture perspective has been getting people's heads wrapped around their role. For whatever reason, a lot of people put importance on titles, right? For instance, when we first defined the roles, we had enterprise architect, solution architect, data architect, security architect, application architect.*
>
> *The titles were like senior consulting systems engineer. That's what the titles were. They worked as enterprise architect, the team's name was enterprise architect, the group was names as enterprise architect but they didn't have the title. Everything started evangelizing from that basic crux of it. That's how it was earlier. It was mainly involved in innovation and emerging technologies. That's where EA got involved.*

> *(Interviewee Top Management)*

The Enterprise Architects had to deal with a challenge of agile adoption in which they had to educate traditional teams about agile methods and their usefulness. One of the new roles which the Enterprise Architects assumed was to help the agile teams understand the importance of agile ceremonies for the timely delivery of projects. **Enterprise Architects played a large role in empowering the smaller development teams in the organization to adapt to the agile practices**. Enterprise Architects play a massive role in supporting smaller development teams, and helped them understand the importance of agile methods and how they their teams help achieve the business value of the organization as a whole. The following interview quotes showcase the phenomenon discussed above:

> *Agile training and education is one of the key aspects that help the teams to embrace agile methods and focus of quick project delivery. The mindset shift of the people is a major factor in the success of the agile teams. Enterprise Architects empowered the agile teams and offered them necessary training to adopt and transition to agile methods smoothly.*
>
> *Our biggest challenges so far have been having folks understand that how you function differently in this model, instead of having multiple ... each team being its own basically micro-silo, and not really working together cross-team as I was saying earlier, and being able to move that up and understand how you work in an Agile fashion, and then how those tie together cross-program, across our program.*
>
> *The biggest change is, and I have to tie this back again to our OCM piece, the biggest change that I have seen is, we moved from very much a ... we're moving from very much a command and control structure to, if you're going to do Agile and introduce the concept of*

> *dev ops at the same time, we're ... we've been given much more freedom to make decisions at our team level.*
>
> *The biggest one is you keep telling people is you can start working when you have incomplete requirements. A lot of developers have a very hard time understanding that you do something and that you may have to revisit it to make changes as the requirements as what the customer needs come into focus. It's hard on customers because they are used to not being bothered.*

The Enterprise Architect resource acted as a safari resource where they can pitch in and contribute to the teams based on dynamic needs. The man-hours of the Enterprise Architects were split up to assist various teams during this scaled agile adoption process. The usage of Enterprise Architects as a shared resource greatly benefitted the company during this transition process. The quote below explains this concept with the example of a chicken and a pig. The evolution of the Enterprise Architects' role during the adoption phase is shown in Figure-3.

> *Safari means basically you are dedicated for a sprint not for that entire program increment and if you are a dedicated resource means like you are dedicated for the entire program increment. You will not change your alliance to the team for that entire 10 sprint time. Okay. Me being a safari resource, my alliance or my allegiance to the team is only for one sprint not for the entire program increment. Infrastructure architects are also like that, solution architects are always dedicated to the team. They are always available. The enterprise architect, security architect or infrastructure architects we are safari.*
>
> *The people who are committed to the sprints are called pigs and the people who are not are called chickens. We as enterprise architects we are chickens for most of the times for*

the team but if there is a specific work is needed for the team, we are pigs for that one sprint. We are not pigs for the entire program increment. For an example a team needs my dedicated attention, my dedicated time for that one sprint or there is a one story that they are playing, I can be a pig for that sprint for the team but I would not be there for all their entire 10 sprints. I am a chicken. They can always call me, consult me, get my time on ad-hoc basis, but I am not committed to them. I am not required to be with them all the time.

(Interviewee Top management)

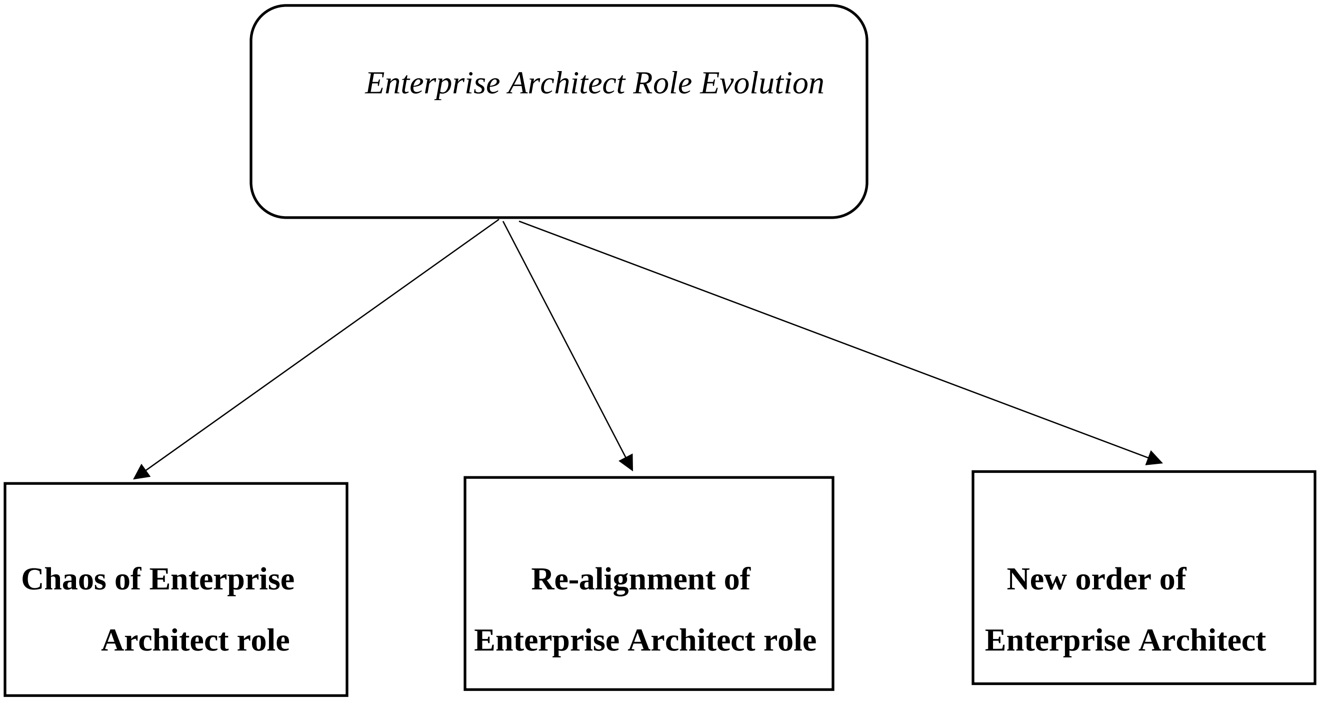

Enterprise Architect Role Evolution
Chaos of Enterprise Architect role
Re-alignment of Enterprise Architect role
New order of Enterprise Architect

Figure 5 – Enterprise Architects' role evolution during agile EA adoption phase

4.3 Quotes table

Dimensions	Company Alpha	Examples from interviews
Uncertainty of Enterprise Architect Expertise	• Uncertainty of the roles and responsibilities • Expectation of varied expertise from the Enterprise Architects	• At the same time, moving from a traditional requirements-gathering waterfall environment the way we always have been, and changing that to Agile, there's a ton of change management going on in this, helping folks not just come up to speed on, but truly understand what these are and how they function. Then marrying the two together when they are traditionally I think there's more of what you're looking for, these are ... traditionally you would see these at odds with each other. With enterprise architecture being very large, slow moving, want to understand, broad and deep across domains. How do you ... that's where we are right now is, how do we find that balance?
		• That's what we're really working hard on right now is what is ... what we have termed as just enough, just in time. How do we measure the ... or how do we balance the desire to get product out the door, from an architecture standpoint, taking that up to a high enough level and being able to look far enough across domains, and understand it across the different domains of architecture to where we don't end up in the same place that we are today.
	• Transition from traditional to agile methods • Reduced artifacts and “just in time” architecture	• What Vishal is referring to is that I came up with a concept of marking an item on the artifact as "just in time", which also meant we would really only create it if, and when, it was needed. Sometimes it may have been at the discretion of the architect themselves, as to whether they felt that it was needed as a work product or not. Other times it was, we may or may not need it at this point in time. We may or may not need it later. We will create that if, and when, we need it at the point in time it's needed.
		• Why I was such a big supporter of enterprise architecture is because I didn't like the way we were

		doing technology here at BNSF. We would have groups, we'd have infrastructure groups and they would work on their silos and then we would have tools architecture teams that would work on their silos. They didn't get their hands dirty, in my opinion, I'll just go out here for the record and say that I believe if you're an architect of any sorts, you need to be implementing technology. • One of the biggest challenges that made it. I think probably the biggest challenge from an enterprise architecture perspective has been getting people's heads wrapped around their role. For whatever reason, a lot of people put importance on titles, right? For instance, when we first defined the roles, we had enterprise architect, solution architect, data architect, security architect, application architect. • To me, the difference is that we're doing smaller increments of work. We're still growing into being able to do that. It is a mindset change for the people that have come into the role because typically, you would architect an entire application. You basically architect what I'm going to be doing for the next two years. Now we have transitioned into doing smaller pieces of what we're going to be architecting. Architecture work's going on all throughout the process. Feature teams are delivering smaller pieces of that work.
Bottleneck of long range planning	• Vision of long range planning. • Groomed Enterprise architects • Roadblock in agile EA.	• The groomed architectures, architects frequently go into other organizations ... Maybe it's done right. In our organization we take engineers and we walk them up through the ranks into architect without taking the engineer out of the engineer. In building hopes you have architects and engineers, and they're different for a reason. Architects think big. They think innovatively. In many cases, scientist and engineers have been groomed to pinpoint. Architects use deductive reasoning. Engineers use inductive reasoning. Scientists use deductive reasoning but they use it at the detail level. When you pick an architect you need someone who can use deductive reasoning very well. You don't generally ... It's hard to find in the ranks of the engineers. If you've got a program where you can groom someone into an architect and their effective.

- Again, I don't know if you were hear when I was talking about the way we create architects, but we promote out of the engineering crafts because they've done something great in their engineering craft and we get them up into the architect craft and they're still engineers but they have different title now.

- The part that really fascinates me is the planning piece, this just in time, just enough and how you are able to, or in the process of applying that and figuring that our for EA, it looks like you're doing that. How is the buy in form, have you gotten a lot of push back form, there must be some traditional EA people there. Have you gotten a lot of push back or are they coming a long?

- Especially as you add business architecture and enterprise architecture into the program, the team struggles a lot with understanding why it doesn't feel like we're making progress faster, and that's a hard message to say we need to slow down in order to speed up. We need to take the time to understand our current business capabilities, do a business discovery map, write out the future for both our solution and enterprise and data architecture in order to ensure we're building something that is scalable and supportable for the next 10 to 20 years.

- One of the weird things is due to my experience, I do like to get my head wrapped around exactly what our long term goal is. Obviously our long term goal is to replace all processes that currently flow within the mainframe. That is NSF business over the years has created a lot of mainframe processes and a lot of caveats within those. To me one of the struggles is trying to understand exactly what all those are. When I'm at a lower level doing some coding I understand the impact across the entire system. The idea is within an MVP you can list all the features that you want to get accomplished so those can get transferred into stories and they can use those as ... Basically you have a feature. Can have a high level epic, right? Each one of the epics can be composed of either sub-epics with stories.

- The biggest challenge to me right now is changing the mindset of everybody. When you're only really focused on 2 weeks' worth of work or planning shorter increments of work to develop a plan for a

		whole year is very challenging. Our executive leadership, they still expect to see what's your plan for a whole year. Or if it's a whole project, what's your plan for 5 years. It's to me very difficult to be able to produce those types of reports when we're really focused on planning small increments of work. That's been my biggest challenge.
Re-alignment of the roles and responsibilities of Enterprise Architects	• The roles of Project Manager • The solution Architects designing applications • Self organize and build the skills	• As far as assigning responsibilities and making sure we're progressing, that's really on the team themselves. The teams are empowered to make those decisions. What we've had to do is really rethink what the manager does, and so what we've done is in a lot of places where it's appropriate, we've had managers now playing the role of PMs but in the same point they're no longer a resource manager. They've become an individual contributor and now they're a PM. We've had some that retained the resource manager capability but they also take on the responsibility of an enterprise solution architect. • At the time there was a project manager opening at my level. Program manager opening at my level of the organization for TSM, Transportation Service Modernization, and I applied for it and several of the directors that I had worked for in the past came to me and said, you know what, you probably don't want a program manager job that's got a high level of detail orientation that you just don't have. We're going to create this job called business architect because we know we need it and we'll have you stand up that department. People that I had worked with in the past, once they understood that I was shopping, they actually found a product for me and build it to me. • The product owner and product manager will get together and get their business stories. Those people will generate those stories. If it's a technical story, we have some technical story of this needs to be validated this way or the test script needs to look like this, then the product owner, the product manager, and the solutions architect SA will get together, and the SA will generate those stories. Then the product owner and the SA will talk about what needs to be a higher priority. I think the technical story needs to go before the business story and why, and then they put all that in the back log of here's a priority of each story so that if there's

any work that needs to be done by the team, they can see what the highest level of priority and what they think they can do and what they think they can't.

- I am a solution architect/team leader. Mainly, I work with the solutioning team to make their actual solution of what we are building and at the application and also I lead the development efforts and work with their development team to make sure that deliver ... what we are delivering is matching with the architecture and solution that we are working with, that we designed with that solutioning team. I'm in the middle between did the architecture and the development.

- I think a lot of the terminology is some of the problems, too. They don't understand what they're supposed to do or how they're supposed to handle it. We're supposed to self organize and self decide how to do things as a team, but then we also have leadership and product owners that are trying to say, "This is how we have to do it," and, "This is how we want it done." That doesn't give the team the ability to decide some of those things.

- Yeah, I think there's been ... One of the biggest challenges that made it. I think probably the biggest challenge from an enterprise architecture perspective has been getting people's heads wrapped around their role. For whatever reason, a lot of people put importance on titles, right? For instance, when we first defined the roles, we had enterprise architect, solution architect, data architect, security architect, application architect.

- We came up with the concept of the enterprise solution architect. What we didn't think of at the time was that when the data architects sat down and looked, they were like, "How come we don't have an enterprise data architect?" Then the security architect said, "How come we don't have an enterprise security architect?" It was an unintended side effect of us creating a role that we felt solved the specific need, that now we had people looking at that, thinking to themselves, somehow or another, because the enterprise word was in front of the domain, that it carried some larger amount of importance.

Decoupling	• I actually started out as an App Dev person many years ago with languages I won't admit to knowing. I actually started out in App Dev and so I had that particular skillset. That is why I said that in the security space being able to fill this role as a security solution architect and provide this consulting is a bit of a rare skillset. The mix of skills that are required becomes critical. The good news is that's why the way we're adapting and the way we are adopting and being able ... I don't require a lot of people with those skillset because they're difficult to find. • Before even I joined, about 2 years ago, people didn't have the title as an enterprise architect. The titles were like senior consulting systems engineer. That's what the titles were. They worked as enterprise architect, the team's name was enterprise architect, the group was names as enterprise architect but they didn't have the title. Everything started evangelising from that basic crux of it. That's how it was earlier. It was mainly involved in innovation and emerging technologies. That's where EA got involved. • They are defined so the deliverables Architects have only one deliverable working code that adds business value. Anything else that you say is necessary and should be done at a level it adds value I completely disagree with the notion that where only theoretical document is considered a deliverable for the architect. that’s where lot of problem with EA starts are with documents. • I will define the word architect in this context. To have a meaningful discussion the discussion has to happen only between the architects who have responsibility for the project. In other words architect should be a working architects. You have the responsibility to deliver project A I have the responsibility to deliver project B the interaction are that technical. • Our biggest challenges so far have been having folks understand that how you function differently in this model, instead of having multiple ... each team being its own basically micro-silo, and not really working together cross-team as I was saying earlier, and being able to move that up and understand how you work in an Agile fashion, and

Enterprise Architects empowering agile teams	• Concerns over people acceptance of agile • Motivating people to start with incomplete requirements • Focus on enterprise Architecture to support agile delivery	then how those tie together cross-program, across our program. The education side of that, and again it all gets back to OCM, the change management piece of that, it is a completely different way of working for our team and as I said, we're only in our second PI, from a scaled Agile standpoint, but it was fascinating to see how much better our second PI went, our PI planning event went, than the first one. • The biggest change is, and I have to tie this back again to our OCM piece, the biggest change that I have seen is, we moved from very much a ... we're moving from very much a command and control structure to, if you're going to do Agile and introduce the concept of dev ops at the same time, we're ... we've been given much more freedom to make decisions at our team level. At the same time, I also have much more open-door policy of what I'm seeing that, yeah, that network is opening up across the organization, not so much just going, in your own silo, going up the chain, to get access across. • My role in the agile environment ... well, the primary role has really been really defining kind of our overarching principles for enterprise architecture. That the agile teams will then be responsible for implementing and or complying with. • The projects in agile have typically ended up with higher amounts of customer satisfaction and development satisfaction. The developers have been happier about it. • Right, we're working thought that and when I say deliverables, what we define in Transformit, is really, we had no concept of minimal liable product for architecture. That's something that I've been pushing on people, just enough just to find the emergent architecture, let's not sit and try to architect this thing form end to end. It's way too big. We had to break it down and say we will do EA for this little piece of what we call shipping process management and then we'll take it all the way down into the implementation level and then we'll implement that piece of technology and then we'll go back up and look what's next, what's on the boundaries. Then we'll start architecting those things. As you can imagine, that's a very difficult

		balance because you have the purest that want to say, "Oh well, if we're going to do EA, we need to do it, we need to sit and we need to figure this out."
Resource allocation of the EA man-hours		• Safari means basically you are dedicated for a sprint not for that entire program increment and if you are a dedicated resource means like you are dedicated for the entire program increment. You will not change your alliance to the team for that entire 10 sprint time. Okay. Me being a safari resource, my alliance or my allegiance to the team is only for one sprint not for the entire program increment. Infrastructure architects are also like that, solution architects are always dedicated to the team. They are always available. The enterprise architect, security architect or infrastructure architects we are safari. You don't need the infrastructure all the time right. You need it only when we are procuring we are putting the concept. Those are Safari. They will come in only for the time when you need to get new hardware, set up new hardware, those infrastructure architects will be committed for that one sprint or two sprints or three sprints and once it is available, they will move on to something else. • Safari means basically you are dedicated for a sprint not for that entire program increment and if you are a dedicated resource means like you are dedicated for the entire program increment. You will not change your alliance to the team for that entire 10 sprint time. Okay. Me being a safari resource, my alliance or my allegiance to the team is only for one sprint not for the entire program increment. Infrastructure architects are also like that, solution architects are always dedicated to the team. They are always available. The enterprise architect, security architect or infrastructure architects we are safari. You don't need the infrastructure all the time right. You need it only when we are procuring we are putting the concept. • The people who are committed to the sprints are called pigs and the people who are not are called chickens. We as enterprise architects we are chickens for most of the times for the team but if there is a specific work is needed for the team, we are pigs for that one sprint. We are not pigs for the entire program increment. For an example a team needs my dedicated attention, mu dedicated time

	for that one sprint or there is a one story that they are playing, I can be a pig for that sprint for the team but I would not be there for all their entire 10 sprints. I am a chicken. They can always call me, consult me, get my time on ad-hoc basis but I am not committed to them. I am not required to be with them all the time.

Table 3 Quotes table

4.4 Discussion

The research method I followed is the grounded theory approach in which we attempted to analyze the interview data to arrive at the codes and relate the meaningful codes to emerge with the constructs. The first step in this process is line by line coding, where I started with many thousands of codes. The next step was to prune the codes and group meaningful codes into categories. With some categories grouped together, the next round of data analysis was a selective coding approach. Here, the idea was to narrow down the focus onto the research question and obtain more meaningful categories. Next I found relationships among the categories and arrived at an emerging theory. The detailed research steps, tasks and outcomes are displayed in the appendix table.

To support the emergent theory, I conducted a review on the existing literature in the related field of study. The findings and the theory of Enterprise Architects, which emerged from the data analysis, supported the existing complex adaptive system theory. In this section I will compare the Enterprise Architect theory to the adaptive system theory and show the relevance. The purpose of complex adaptive system theory is to support the emergent theory from the findings about the role of the evolution of Enterprise Architect.

The first emergent aggregate dimension is the **chaos of the Enterprise Architect,** which discusses the uncertainty the Enterprise Architects went through when the company was initially transitioning from waterfall to agile Enterprise Architecture. There was confusion about the roles and the responsibilities of Enterprise Architects when the organization adopted agile Enterprise Architecture. Slowly, the Enterprise Architects in the firm self-organized to adapt to the changes in the environment. There was coupling of Enterprise Architect roles and responsibilities where they were shouldered multniple responsibilities. The **self-organizing aspect of the Enterprise Architects** helped them **re-align to the roles and responsibilities** that were needed for the organization to adopt agile Enterprise Architecture and succeed, as which shown in figure-5.

The final aggregate dimension is the **new order of the Enterprise Architect** who develop innovative ideas and empower the agile teams in an organization to fully function with the adoption of agile Enterprise Architecture. The new role of the Enterprise Architect is to help agile teams understand agile ceremonies and to nurture a collaborative work culture. The Enterprise Architects also took up the challenge of being a safari resource to help the teams in an ad hoc manner. The key for the Enterprise Architect is to handle dependencies across the upper and lower levels of the hierarchy in an organization.

Figure 3 provides an overview of the evolution of the Enterprise Architect role during the agile EA adoption phase in the company “Alpha.” Enterprise Architects play a key role in the agile adoption of an organization, and I focus on the Enterprise Architect’s maturity and how they assist the organization to succeed in agile EA adoption. Complex adaptive systems theory supports the Enterprise Architect’s initial chaos, when the organization transitions from traditional to agile Enterprise Architecture. The focus is then on the self-organizing aspect of the Enterprise Architect to re-align their group and adapt to the new roles and responsibilities expected of them, as depicted in Figure 3. The Enterprise Architect must also innovate in an organization, adopting agile EA by tailoring the agile Enterprise Architecture to suit the organization. The results showcase the self-organizing and creative aspect of Enterprise Architects in this complex adaptive system scenario, as discussed in the above case study.

5. CONCLUSION

In this paper, I analyze the factors that influence agile Enterprise Architecture adoption and focus on the Enterprise Architect aspect. It is a novel idea to dig deep into the Enterprise Architect perspective and identify the phases that Enterprise Architects undergo. The findings were interesting and show the ability of Enterprise Architects to self-organize and adapt to a complex system, and to help an organization adopt agile Enterprise Architecture. Overall, Enterprise Architects evolve and innovate to help an organization achieve better agile EA framework. However, there are few limitations to this study because the results are presented based on data gathered from one firm in the transportation sector. The general applicability of the Enterprise Architects' perspectives during agile Enterprise Architecture must be considered with further case studies. Future work targets similar case studies on organizations in different sectors to identify the role of Enterprise Architects during agile EA adoption.

References

1. Abrahamsson, P., Conboy, K., & Wang, X. (2009). 'Lots done, more to do': the current state of agile systems development research. *European Journal of Information Systems, 18*(4), 281.
2. Ambler, S. W. (2009). The agile scaling model (ASM): adapting agile methods for complex environments. *Environments*.
3. Ambler, S. W., & Lines, M. (2013). Going Beyond Scrum: Disciplined Agile Delivery. *Disciplined Agile Consortium. White Paper Series*.
4. Cameron, B. H., & McMillan, E. (2013). Analyzing the current trends in enterprise architecture frameworks. *Journal of Enterprise Architecture*, *9*(1), 60-71.
5. Cao, L., Mohan, K., Peng, X., & Ramesh, B. (2009). A framework for adapting agile development methodologies.*European Journal of Information Systems*, 18, 4, 332-343
6. Chan, F., & Thong, J. (2009). Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems* , 46, 803-814.
7. Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, *34*(11), 131-133.
8. Conboy, K., & Wang, X. (2007). Agile practices in use from an innovation assimilation perspective: a multiple case study.
9. Dingsøyr, T., & Moe, N. B. (2014, May). Towards principles of large-scale agile development. In *International Conference on Agile Software Development* (pp. 1-8). Springer International Publishing.
10. Dyba, T., & Dingsoyr, T. (2009). What do we know about agile software development?. *IEEE software, 26*(5), 6-9.

11. Fallmyr, T., & Bygstad, B. (2014, January). Enterprise Architecture Practice and Organizational Agility: An Exploratory Study. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on* (pp. 3788-3797). IEEE.

12. Fichman, R. G., & Kemerer, C. F. (1997). The assimilation of software process innovations: An organizational learning perspective. *Management science*, *43*(10), 1345-1363.

13. Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems*, *15*(2), 200-213.

14. Isham, M. (2008, August). Agile architecture is possible you first have to believe!. In *Agile, 2008. AGILE'08. Conference* (pp. 484-489). IEEE.

15. Jonkers, H., Lankhorst, M. M., ter Doest, H. W., Arbab, F., Bosma, H., & Wieringa, R. J. (2006). Enterprise architecture: Management tool and blueprint for the organisation. *Information Systems Frontiers*, *8*(2), 63-66.

16. Laanti, M. (2014, May). Characteristics and principles of scaled agile. In *International Conference on Agile Software Development* (pp. 9-20). Springer International Publishing.

17. Leffingwell, D. (2010). *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. Addison-Wesley Professional.

18. Mangalaraj, G., Mahapatra, R., & Nerur, S. (2009). Acceptance of software process innovations -- the case of Extreme Programming. *Empirical Software Engineering* , 18, 344-354.

19. Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72-78.

20. Pereira, C. M., & Sousa, P. (2004, March). A method to define an Enterprise Architecture using the Zachman Framework. In *Proceedings of the 2004 ACM symposium on Applied computing* (pp. 1366-1371). ACM.
21. Poston, R., & Patel, J. (2016). Making Sense of Resistance to Agile Adoption in Waterfall Organizations: Social Intelligence and Leadership.
22. Reifer, D. J., Maurer, F., & Erdogmus, H. (2003). Scaling agile methods. *IEEE software*, *20*(4), 12-14.
23. Rogers, E. M. (2003). Diffusion of innovations, edn. *Free Pres., New York*.
24. Ross, J. W., Weill, P., & Robertson, D. C. (2006). *Enterprise architecture as strategy: Creating a foundation for business execution*. Harvard Business Press.
25. Russo, N. L., Fitzgerald, G., & Shams, S. (2013). Exploring adoption and use of agile methods: A comparative case study.
26. Sambamurthy, V., Bharadwaj, A., & Grover, V. (2003). Shaping agility through digital options: Reconceptualizing the role of information technology in contemporary firms. *MIS quarterly*, 237-263.
27. Senapathi, M., & Srinivasan, A. (2012). Understanding post-adoptive agile usage: An exploratory cross-case analysis. *The Journal of Systems and Software, 85,* 1255-1268.
28. Sidky, A., Arthur, J., & Bohner, S. (2007). A disciplined approach to adopting agile practices: the agile adoption framework. *Innovations in systems and software engineering*, *3*(3), 203-216.
29. Tallon, P. P., & Pinsonneault, A. (2011). Competing perspectives on the link between strategic information technology alignment and organizational agility: insights from a mediation model. *Mis Quarterly*, 463-486.

30. Tang, A., Han, J., & Chen, P. (2004, November). A comparative analysis of architecture frameworks. In *Software Engineering Conference, 2004. 11th Asia-Pacific* (pp. 640-647). IEEE.
31. Urbaczewski, L., & Mrdalj, S. (2006). A comparison of enterprise architecture frameworks. *Issues in Information Systems*, *7*(2), 18-23.
32. Vaidya, A. (2014). Does dad know best, is it better to do less or just be safe? adapting scaling agile practices into the enterprise. *PNSQC. ORG*, 1-18.
33. Vijayasarathy, L. E. O. R., & Turk, D. (2008). Agile software development: A survey of early adopters. *Journal of Information Technology Management*, *19*(2), 1-8.
34. Zarvić, N., & Wieringa, R. (2014). An integrated enterprise architecture framework for business-IT alignment. *Designing Enterprise Architecture Frameworks: Integrating Business Processes with IT Infrastructure*, *63*.
35. Vidgen, R., & Wang, X. (2006). Organizing for agility: A complex adaptive systems perspective on agile software development process.
36. Stacey, R. D. (1996). *Complexity and creativity in organizations*. Berrett-Koehler Publishers.
37. Anderson, R. A., Issel, L. M., & McDaniel Jr, R. R. (2003). Nursing homes as complex adaptive systems: relationship between management practice and resident outcomes. *Nursing research*, *52*(1), 12.
38. Choi, T. Y., Dooley, K. J., & Rungtusanatham, M. (2001). Supply networks and complex adaptive systems: control versus emergence. *Journal of operations management*, *19*(3), 351-366.

39. Janssen, M., & Kuk, G. (2006, January). A complex adaptive system perspective of enterprise architecture in electronic government. In *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on* (Vol. 4, pp. 71b-71b). IEEE.

40. Janssen, M. (1998). Use of complex adaptive systems for modeling global change. *Ecosystems*, *1*(5), 457-463.

Appendix

General interview questions were developed as probes to illicit information. A sample of these questions is contained below.

Phase 1 interview questions

Interviewee roles : Top management, Enterprise Architects, Program managers

History and legacy in BNSF

Please begin by giving me a short history of your own career and how you came to be working with your present organization.

Please give me a brief overview of the company and its history.

Traditional EA

How would you typically describe the work process in traditional EA?

How many artifacts you used in traditional EA projects?

What were the major failures/obstacles you faced in traditional EA implementation?

Were there misalignments between different EA groups?

What were the key challenges in implementing traditional EA practices?

How long it took to implement changes with traditional EA?

Move towards scaled agile EA

What are the key drivers behind the move to transfer to agile methods?

What was the role of top management in agile transformation?

How do you plan to implement scaled agile?

What is the project structure? What is the timing for each phase?

What was done with respect to product, processes, tools, and facilities to realize there was smooth transition between traditional EA and scaled EA practices?

What are the success criteria for the scaled EA program, for each phase?

How far should it be alignment with company's vision? How important is it to be aligned? How feasible?

Which are the processes to identify and deal with differences and exceptions?

What are the potential failures/obstacles you foresee (people, processes, product, tools, infrastructures)?

What are the potential benefits (financial, learning, etc.) of scaled agile?

Which (management technical) tools, methods, and resources are being put in place to facilitate the scaled agile?

After implementing EA

How far are you aligned with company's vision? And how far are you aligned with scaled agile frameworks?

What are the weak areas or obstacles? What is being done to remedy this?

What in your view has succeeded; what has failed?

Which tools, methods, and resources are being put into place for this stage?

Have relationships/structures/cultures/motivations changed (across sites, within sites, between site and global company)? If yes, how have they changed?

In the future should it be alignment or drift?

How important is it to stay aligned? How feasible?

Phase 2 Interview questions

Interviewee roles : Developers, senior developers, project managers

Introduction

Please begin by giving me a short history of your own career and how you came to be working with your present organization.

Please give me a brief overview of the company and its history.

Scaled agile process level questions

Please describe a "typical" agile enterprise architecture design project or technology change situation that your organization is involved in terms of size, duration, the number of participants (and their locations), stakeholders, and deliverables.

Describe in chronological sequence how a project was initiated since its start (can you check that from your calendar, e-mail, activity log etc)?

What are the tasks you performed describe them in sequential order and tell us about the people you interacted, what tools you used and where was it done?

Also tell us the duration of each activity and lag between activities?

What was your role in the task?

What were the deliverables in each task?

Were people open to new ideas?

How was the experience working with people outside your team?

How do you manage friction between different groups or inside the group?

Few snippets of code groupings to generate themes

Uncertainty of EA role

1. Traditional vs agile process dilemma
2. EA role Ambiguity
3. Change in existing EA process
4. Confusion over EA responsibilities
5. Lack of clarity in EA role

Bottleneck of long range planning

1. Groomed Architects
2. Reasoning approach

3. Lack of quick and innovative thinking
4. Lack of EA vision
5. Less clarity in EA planning

Coupling

1. Added EA responsibilities
2. Hiring new Enterprise Architects
3. Multiples role handling
4. More scope for EA
5. Self-organizing aspect of EA

De-Coupling

1. Emergence of security consultant role
2. Creating new specialized roles
3. Wrapping roles and responsibilities in people's head
4. Self-organizing and EA alignment
5. Acquiring clarity

Empowering agile development teams

1. Enterprise Architects help in agile transition
2. EA offer agile education
3. Assist agile teams in learning phase
4. Encouraging teams to work in agile style

Managing dependencies across upper and lower hierarchy

1. Enterprise Architect assisting top management

2. EA guys empowering agile teams
3. Handling dependency issues
4. TransformIT

Resource allocation of Enterprise Architect manhours

1. Chicken and pig
2. Acting as safari resource
3. Juggling across multiple teams
4. EA support on adhoc basis