Content

Introduction ................................................................. 1
1.1 Background .............................................................. 1
1.2 Problem ................................................................. 1
1.3 Purpose ................................................................. 1
1.4 Research Questions .................................................. 2

Literature Review ........................................................... 2
2.1 Business Transformation and SOA ................................ 2
2.2 What is a SOA? .......................................................... 4
2.3 Service Orientation and SOA ........................................ 5
2.4 IT Flexibility ............................................................ 9
2.5 Business Agility ....................................................... 10
2.6 Competitive Advantage ............................................. 11
2.7 SOA Business Case .................................................. 12
2.8 Substantive Theory ................................................... 13

Methodology ..................................................................... 15
3.1 Research Method ..................................................... 15
3.2 Research Design ...................................................... 16
3.3 Sample ................................................................. 17
3.4 Data Collection ....................................................... 17
3.5 Data Analysis .......................................................... 18
3.6 Method Critique ..................................................... 19
4 Presentation and Analysis of Findings ................................................................. 20
   4.1 Open Coding ........................................................................................................ 20
   4.2 Axial Coding ......................................................................................................... 25
   4.3 Selective Coding ................................................................................................... 27
       4.3.1 Causal Conditions .......................................................................................... 30
       4.3.2 Core Category ............................................................................................... 31
       4.3.3 Context ......................................................................................................... 31
       4.3.4 Intervening Conditions .................................................................................. 32
       4.3.5 Action/Interaction Strategies ....................................................................... 32
       4.3.6 Consequences ............................................................................................... 34

5 Conclusion .............................................................................................................. 35
   5.1 Conclusive Discussion ......................................................................................... 35
   5.2 Future Research ................................................................................................... 37

Appendix I .................................................................................................................... I
Appendix II .................................................................................................................... XIII

References .................................................................................................................. XXI
List of Figures

Figure 1: IT & business force forces enterprises to become more dynamic ..................3
Figure 2: Three architectural perspectives ..................................................................8
Figure 3: A model of SOA effects on competitive advantage ....................................14
Figure 4: The research design: connecting the research question to data .................16
Figure 5: Open coding: Investment reasons .................................................................22
Figure 6: Open coding: IT-flexibility ...........................................................................23
Figure 7: Open coding: Business agility ......................................................................25
Figure 8: Axial coding: Relationships .........................................................................26
Figure 9: Axial coding: Relationships II ......................................................................27
Figure 10: Selective coding: Merged subcategories ...................................................28
Figure 11: Selective coding: Core category and aligned subcategories ......................29
Figure 12: Selective coding: Paradigm model with aligned core and subcategories .....30
1 Introduction

1.1 Background

Permanent changes in the business environment affect the value creation of organizations: Demands for more flexibility and agility impact on business design and execution (Cherbakov et al., 2005). General Managers, therefore, need to understand information and communication technology (IT) to its fullest potential in order to generate significant cost advantages and/or new value propositions, thus to gain competitive advantages (Lacity, n.d.). Service-oriented architectures (SOAs) provide an architectural innovative solution for these challenges: A new capability for organizations to respond quickly and effectively to business change and to leverage that change to gain a competitive advantage (Bieberstein et al., 2006, p.31).

1.2 Problem

However, many executives – responsible for setting strategic direction, business planning, and capital investment – have grown a negative attitude towards the power of IT to effect organizational performance (Murray, 2007): Few managers understand the extent to which IT plays a role in shaping their organizations’ strategies (Nolan & McFarlan, 2005, p.96): The continuous development of IT makes it difficult for non-experts to judge related benefits, resulting in a negative feeling of unpredictability (Sikora, 2005, p.210). While scientific understanding of SOAs is improving and technology is maturing, a more detailed analysis of the management-related questions associated with its impact on business is still lacking (Viering et al., 2009, p.45). Lacking such knowledge, general managers cannot understand the business potentials of SOAs, which in turn prevents them from aligning new IT infrastructures with competitive strategies, so that organizations fail to become more agile, innovative, and entrepreneurial, and subsequently to achieve competitive advantages.

1.3 Purpose

This research describes and analyses the interaction between SOAs and competitive advantage. The overall objective can be summarized as follows: describe and analyze the effects of SOAs on organizations’ competitive advantages.
1.4 Research Questions

The research question of this thesis read as follows:
How does a SOA effect on the creation of a competitive advantage?

2 Literature Review

2.1 Business Transformation and SOA

Technological discontinuities, plenty of regulatory and governmental compliance requirements, rapid shifts in customers’ needs and wants, and competition from nontraditional players, caused a turbulent faster business environment (Hamel & Valikangas, 2003), so that the future is hard to predict. Fundamental, therefore, is business agility to manage unforeseeable events as they occur: that is, to do business on demand (Hancock et al., 2003). An on demand business integrates end-to-end business processes and information – with key partners, suppliers, and customers – across an organization, enabling companies to respond with speed and flexibility to any customer demand, market opportunity, or external threat (Zisman, 2003). To gain business agility, “firms need to focus, become far more responsive, migrate more of their cost structure to variable models and develop resilient operations that can withstand a multitude of threats.” (Hancock et al., 2003):

- **Focus** – A focused organization concentrates on core business operations (customer-led competencies), which are strategically essential, and thus differentiate the business from competitors; Non-differentiating business operations are outsourced to highly qualified partners via tightly integrated value networks; Resources are allocated to core business operations, rather than concentrating investments on business operations that are performed better and more cost-effective by focused external partners (Hancock et al., 2003; Bieberstein et al., 2006).

- **Responsive** – A responsive organization senses changes in the business environment and responds dynamically to changing customer needs and further market conditions, thus is able to adapt rapidly; Furthermore, a responsive organization analyzes real-time information to make real-time decisions, and implement them correctly (Hancock et al., 2003).
- **Variable** – A variable organization adjusts cost structures and business operations flexibly, using external partners to transform fixed costs into variable costs, which is achieved through variable pricing and supply: that is, variable organizations pay only for actual use, not idle availability (Hancock et al., 2003).

- **Resilient** – A resilient organization continues to operate with consistent availability, security, and privacy despite unforeseen changes and threats – whether technological, economic, natural or political, by safeguarding human, physical and virtual assets, and constructing redundancy in key parts of the operational model (Hancock et al., 2003).

These business forces and the co-evolution of open standard, shareable, and flexible IT infrastructures (see Figure 1) have led to enterprise reconstruction and industry deconstruction, and transformed organizations into integrated industry networks, i.e. “ecosystems” (Bieberstein et al., 2006), characterized by a large number of loosely interconnected participants who depend on each other for their mutual effectiveness and survival (Iansiti & Levien, 2004, p.6).

![Figure 1: IT & business force forces enterprises to become more dynamic](source: Based on Bieberstein et al., 2005.)

To reconstruct and deconstruct (i.e. increase business agility), organizations need to concentrate on business operations as a set of interconnected business functions: that is,
discrete business processes and services, which are represented as business components, making up a business; SOAs offer a valuable response to the need for flexibility in business operations by providing the core structure of an on demand business (Bieberstein et al., 2006).

2.2 What is a SOA?

SOA has been defined in a number of different ways, exemplifying diverse understandings of what SOAs are: Definitions range from a solely technology driven approach to a business-focused approach (Georgakopoulos et al., 2007, p.192). For the purpose of this thesis, the following definitions are investigated: “A Service Oriented Architecture (SOA) is a form of distributed systems architecture that is typically characterized by the following properties: logical view […], message orientation […], description orientation […], granularity […], network orientation […], and platform neutral […]” (W3C, 2004). Additionally, the World Wide Web Consortium (W3C) details each of these properties in terms of services. That is, the W3C focuses on the technical aspects of architectures associated with services, while excluding business characteristics. Based on this, SOAs narrowest definition reads as follows: “SOA is a synonym for solution architectures making use of Web service technologies such as SOAP, WSDL, and UDDI, or – in other words – “SOA is defined as any product and project architecture conforming to the W3C Web services architecture (WSA).” (Bieberstein et al., 2006, p.5). However, Papazoglou (2008, p.23) states that: while the concept of Web Services and SOAs are often discussed in conjunction, these two are not synonymous: Web Servicces are not required to implement SOAs; rather they facilitate the SOA deployment, and are key in enabling interoperability. Interoperability, in turn, is a core concept of SOA (Josuttis, 2007, p.18), thus Web Services enable SOA (Söderström & Meier, 2007, p.389). McCoy and Natis (2003) extend this technical definition by integrating the relationship

1 The W3C is a consortium that drives the development of Web standards. W3C is supported by members of organizations, working full-time, and the public.
2 SOAP (Simple Object Access Protocol): A standard communication protocol, enabling the data exchange between Web Services.
3 WSDL (Web Service Description Language): A WSDL document describes the Web service interface, delivering a point of contact, more specifically; it is usable to determine what a Web service can perform, where it resides, and how to invoke it (Billy & Joseph, 2003, p.52).
4 UDDI (Universal Description, Discovery, and Integration): A mechanism for clients to dynamically find other Web services, that is, using a UDDI interface organizations can dynamically link to business services offered by external business partners (Billy & Joseph, 2003, p.52).
between architectures, stakeholders, services, business functions, and agility, thus establishing the interplay between SOAs and business: “SOA is a software architecture that builds a topology of interfaces, interface implementations and interface calls. SOA is a relationship of services and service consumers, both software modules large enough to represent a complete business function. So, SOA is about reuse, encapsulation, interfaces and, ultimately, agility.” This notion is further explicated and developed in the definition given in Bieberstein et al. (2006, p.5): “A service-oriented architecture is a framework for integrating business processes and supporting IT infrastructure as secure, standardized components—services—that can be reused and combined to address changing business priorities.” Relating to this definition, used in the present thesis, Granebring (2007, p.15) argues: SOA is a framework, not a technology, making process matters an essential consideration (Sprott & Wilkes, 2004). As the main aspect of SOAs, standardized components offer the opportunity to reuse and interoperate, enabling business processes to be responsive with a high degree of flexibility (Crawford et al., 2005). Separating the term “SOA”, the structure (architecture) differentiates from the power (service orientation) to provide business agility (Granebring, 2007). Both service orientation and business agility will be clarified in the following paragraphs.

2.3 Service Orientation and SOA

To achieve business agility (i.e. reconstruct and deconstruct) 5, organizations need to focus on componentization and service orientation. Componentization – realized through business components, representing atomic functional business elements – permit organizations to deconstruct, analyze, and then reconstruct into value nets, in which partnerships with customers and suppliers operate in a network; Business components, therefore, are connected both intra-firm and across organizational boundaries with components provided by external partners, enabling a flexible network: Organizations can in-source an outsourced component and vice versa, or – put another way – substitute current with different partners (Cherbakov et al., 2005, p.654). Cherbakov et al. (2005, p.656) describe a business component as follows:

- **Business Purpose** – defines the fundamental value offered by the business component, and the reason for its existence.

---

5 Cf. Chapter 2.1
• **Activities** – conducted by the business component, i.e. activities that are within its boundaries.

• **Resources** – tangible (e.g. people) or intangible (e.g. knowledge) – needed to operate.

• **Governance** – a management mechanism – required for autonomous operations, involving metrics and procedures for motivation, performance, and accountability.

• **Services**\(^6\) – offered and consumed, whereas service orientation enables the definition of components within exact boundaries.

A business component serves a unique purpose, offering a service for consumption to other business components (Flaxer & Nigam, 2004). Thus, “the key to seamless integration between business components is service orientation.” (Cherbakov et al., 2005, p.654). In this context: A service is a repeatable business task; thus, service orientation integrates business components as a linked chain of services and associated outcomes (Paoli et al., 2008, p.11).

Creating a flexible, innovative and fast-to-market operating organization does highly depend on enhanced integration between IT and business processes (Summit, 2004). Therefore, underlying IT needs to follow a transformation towards an improved state of integration, affecting processes, technologies and workforces (Summit, 2004). In order to benefit from the establishment of SOA the transformation process has to be handled carefully, since it contains of several economical, organizational and technological risks (Cherbakov et al., 2005). The most critical adoption factors of service-oriented systems according to Lue & Chang (2007, p.44) are described as: insufficient technology planning, lack of expertise, ineffective project governance, and organizational misalignment.

Thus CIOs and project managers should be aware of the necessity of comprehensive IT planning focused on the close integration of technology and business. Moreover not dealing with developer’s lack of expertise can for instance result in extended project times and costs. Ineffective project government can, for example, be overcome through better communication between IT workers, top management and end-users. A further condition of

---

\(^6\) Services are resources, or – in terms of IT – software, aiming at executing one or more tasks, realized by a provider and consumed by a requester (Granebring, 2007, p.16) – a mechanism, fulfilling requesters (B2C or B2B) needs or wants in accordance with a negotiated contract, which includes service agreement, function offered etc. (Sprott & Wilkes, 2004).
establishing a service oriented enterprise is the need of aligning diverse organizational departments in order to be inter alia more sensible regarding changing customer demands and thus react faster to market changes. (Lue & Chang, 2007)

When SOAs are implemented successfully, and thus enable service oriented organizations, then business value is created through services: First, business services are marketable to external business components respectively organizations, thus generating revenues (Cherbakov et al., 2005, p.655); and second, business services reduce the costs of change (Bieberstein et al., 2006, p.26): that is, costs associated with responses to change in the business environment.

To permit service-orientation, Fritz (Heutschli, 2007, p.194) accentuates 5 principles to which a service must be related to increase business value and adaptive business solutions:

- **Abstraction**: hides the underlying details of a service, and enables and preserves loose coupling (Erl, 2008, p.72).

- **Modularity** – permits for reusable pieces as a consequence of breaking down complexity.

- **Standardized connectivity** – facilitates flexible service composition to form larger processes and scenarios.

- **Loose coupling** – creates a specific type of relationship within and outside of service boundaries, emphasizing the reduction of dependencies (loosening) between the service contract, its implementation, and its service consumers (Erl, 2008, p.71).

- **Incremental design** – permits changes to compositions and configurations without influencing the insides of components.

SOAs concern the structure (architecture) and policy, required to enable the principles of service orientation (Allen, 2006). Sprott and Wilkes (2004) specified 3 architectural perspectives that are fundamental to SOA (see Figure 2):
• **Application architecture** – represents the business perspective, and consumes services offered by providers, while integrating them into business processes.

• **Service architecture** – operates as a link between databases, components and the consuming applications, and generates a logical view of available, usable services sets, which are invoked by a common interface.

• **Component architecture** – represents the underlying technical perspective, and determines diverse environments (e.g. databases), and fosters implemented applications, business objects and associated implementations.

A key element of the architecture is the Business Service Bus (BSB) respectively Enterprise Service Bus (ESB) – “a standards-based integration platform that combines messaging, web services, data transformation, and intelligent routing to reliably connect and coordinate the interaction of significant numbers of diverse applications across extended enterprises with transactional integrity” (Chappell, 2004, p.1), thus interconnects all services, enabling the (loose) coupling characteristics (Bieberstein et al., 2006, p.61), and interoperability of services (Josuttis, 2007, p.296): An ESB can connect applications generated with .NET, COM, C#, and legacy C/C++, can use J2EE components, and J2EE web services APIs., XML standards; the combined use of these standards deliver an open-ended, pluggable SOA, so that it supports industry-standard integration and proprietary elements through the utilization of
standardized interfaces; To build business processes across organizations boundaries, all applications and services are treated as abstract endpoints, which are connected into the bus (Chappell, 2004).

To sum up: At its core, an ESB permits the communication between diverse services across a range of different platforms; it decouples the consumer’s perspective of a service from the implementation and the technical aspects of service communication; Separating the consumer’s view of a service from the implementation massively increases IT flexibility: A service provider can be substituted with another without the need to change the IT infrastructure (Gao, 2006).

2.4 IT Flexibility

IT infrastructures are a collection of shared, tangible, IT resources (platform technology, network and telecommunication technology, key data, core data processing applications), which offer a basis to facilitate present and future business applications (Duncan, 1995). According to Byrd and Turner (2001), flexible IT infrastructures can support 1) diverse technologies that can be disseminated into the overall technological platform with ease in order to distribute information (data, text, voice, image, video) to everyplace within and across organizations boundaries, and 2) the design, deployment, and implementation of heterogenous business applications. Duncan’s (1995) research – conducted on IT infrastructures – has identified 3 attributes, constituting IT flexibility: connectivity, compatibility and modularity. Other studies have supported the core characteristics suggested by Duncan (Paschke et al., 2008). The characteristics of IT infrastructure flexibility can be described as follows (Byrd & Turner, 2000):

- **IT connectivity** – constitutes the capability of technological modules to connect to other modules within and beyond organizations boundaries, enabling

- **IT compatibility** – constitutes the capability to distribute information across any technological modules.

- **IT modularity** – constitutes the capability to easily complement, change, and eliminate software and/or hardware without major overall effects on the entire IT infrastructure.
IT flexibility enables business agility, because enhancement of business agility is not possible unless IT infrastructures are flexible (Furukawa, 2004, p.780). Flexible IT infrastructures deliver the technical platform, services and specialist resources required to deal quickly with unpredictable changes in the business environment (Bocij et al., 2008, p.557).

2.5 Business Agility

The concept of agility was first established within the business literature in 1991: Researchers at the Iacocca Institute of Lehigh University published an industry-based report, and coined the term “agile manufacturing” (Iacocca Institute, 1991). Since then, no consensus on a definition has been established (Oosterhout et al., 2007, p.53). Oosterhout et al. (2005, p.277) discuss common aspects among diverse definitions, and determined agility as “a way to cope with (to a large extent) unforeseen changes.” Thus, change is at the heart of agility, characterized by a high level of uncertainty. Milliken (1987, pp.136-38) distinguishes three different types of uncertainty:

- **State uncertainty** – involves a lack of understanding about how elements of the business environment might change;

- **Effect uncertainty** – is defined as the incapability to predict the effect of a change on an organization;

- **Response uncertainty** – relates to a lack of knowledge of how to respond to change and/or the incapability to prognosticate associated consequences of a response choice.

In this context, business agility emphasizes the capability to sense highly uncertain changes, forcing organizations to respond – either reactively or proactively – to these changes; While the term reactive signifies a fast and efficient response to change with the goal to maintain competitiveness, proactive means to lead or initiate change to develop a competitive advantage (Oosterhout et al., 2007, p.53) Bloomberg and Schmelzer’s (2006, p.12) definition of business agility is adopted in the present thesis: “[…] the ability to respond quickly and efficiently to changes in the business environment and to leverage those changes for
competitive advantage”; thus, change requires fast responsiveness, but also provides the possibility to gain a competitive advantage, whereas IT flexibility – the ability to quickly implement change (Evans, 2002, p.6) – is related to profitability: Byrd and Turner (2001) found that flexible IT infrastructures correlated positively with competitive advantage.

2.6 Competitive Advantage

The concept of competitive advantage has been established by Michael E. Porter (1985). Porter (1996) argues: Activities are at the heart of competitive advantage, possessing two dimensions: cost and differentiation: Costs are produced by executing activities: When an organization performs activities more efficiently than competitors, then a cost advantage emerges: that is, activities are performed at lower cost. Differentiation originates from how activities are executed, i.e. activities are performed better: When an organization performs activities uniquely, then customer value arises (e.g. product quality) (Porter, 1991). In this context: Competitive advantage emerges when an organization can provide products and services at lower cost than competitors, and/or when an organization can provide products and services at a higher price than competitors (Porter, 1985).

Activities produce value, which is determined by the market and customers; the value added to a product – above the costs necessary to execute the activity – is measured as the profit margin: that is, the overall performance of interrelated value activities is illustrated in the margin created, whereas profit margins are generated when an organization raises prices through differentiation or reduce the cost per unit (Hedman & Kalling, 2002).

The configuration of activities and how they interrelate is characterized by an organization's strategy (Porter, 1991). Porter (1980) developed three generic strategies to achieve competitive advantage:

- **Cost leadership** – an organization aims at being the lowest-cost producer within the industry, enabling organizations to outcompete rivals. Cost leadership allows commanding the lowest prices, whereas the low-cost basis still permits superior profits: that is, rivals cannot meet the price (Henry, 2008).

- **Differentiation** – an organization aims at generating products and services which customers perceive as unique or different, enabling organizations to meet customer needs more precisely. Differentiation permits charging premium prices, allowing
organizations to make superior profits: that is, high prices exceed the costs of differentiation (Henry, 2008).

- **Focus** – an organization aims at targeting a segment within the market, enabling organizations to serve customers within it particularly well. Focus is a mix of differentiation and low-cost strategy: that is, an organization has different strategies for different groups of customers (Hedman & Kalling, 2002).

### 2.7 SOA Business Case

Based on a review of 3 commonly used service definitions, Luthria and Rabhi (2009, p.96) classified 3 compliant, technical SOA principles which enable IT flexibility, and subsequently business agility:

- **Modularity**: SOAs offer a mechanism to implement business components in a modular manner; Modules can be reused and recombined across multiple business processes, enabling IT flexibility and business agility, thus innovative and competitive (Rosen, 2007).

- **Loose coupling**: Loose coupling permits to compose compatible business services (Mueller et al., 2007, p.1611), allowing organizations to replace or change business components, without changing other components in the IT architecture, which in turn leads to IT flexibility and agility compared to tightly coupled IT architectures (Linthicum, 2007): Organizations can adapt and respond to unanticipated events (Kaye, 2003, p.2).

- **Open standards**: Open standards breaking down proprietary barriers between software programs (Bieberstein et al., 2006, p.19), enabling heterogeneous IT infrastructures to interoperate, so that internal and external business services are easily connected via interfaces (Mueller et al., 2007, p.1615). Organizations need standards to become agile: The adoption of standards enhances business agility: When IT architectures are similar, it is much easier to make changes; Agility favors standardization, and standardization enables IT flexibility (Bloomberg & Schmelzer, 2006, pp.86-87).
To sum up: SOAs characteristics foster IT flexibility and, therefore, agile organizations, i.e. able “to cope with unexpected changes, to survive unprecedented threats of business environment, and to take advantage of changes as opportunities” (Sharifi & Zhang, 1999, p.9): Modularity, loose coupling, and open standards facilitate the reconstruction of a business process if business requirements change by changing only the interconnection of business services (Kano et al., 2005, p.680).

When SOAs address IT flexibility and business agility, then business benefits are concerned, because SOAs provide an instrument to be cheaper, better, and faster, thus more profitable in a dynamic sea of change (Goranson, 1999, p.69). Profitability is defined as the difference between what the organization earns in the form of revenue and what it pays out in the form of cost (Baumol & Blinder, 2008, p.158). Lower costs than rivals and/or the ability to differentiate and command premium revenues result in competitive advantage (Porter, 1991, p.101).

2.8 Substantive Theory

Based on the literature review, Figure 2 represents a substantive theory7 8 illustrating how a SOA affects the development of a competitive advantage for an organization. Effects of SOAs were specified through a simplified – causal conditions and consequences – paradigm model9. That is, the model provides a visual representation of the interactions between SOAs technical principles, IT flexibility, business agility, cost and revenue, and competitive advantage. The substantive theory can be summarized as follows: Costs and revenues are indirectly affected by SOA facilitated IT flexibility: Modularity, loose coupling, and open standards permit IT

---

7 Theory: Theories are generalizations - an attempt to draw generalizable findings from specific instances (Fischer, 2007, p.133).

8 Substantive theory: “[...] substantive theory is grounded in research in one particular substantive area. A theory at such a conceptual level, however may have important general implications and relevance, and become almost automatically a springboard or stepping stone to the development of a grounded formal theory.” (Glaser & Strauss, 1967, p.79). While substantive theory is developed for an empirical area; the formal theory is developed for a conceptual area (Glaser & Strauss, 1967, p.32).

9 Paradigm Model: A mechanism to propose relationships – representing causal conditions, phenomenon, context, intervening conditions, action/interaction, and consequences – among broader contextual issues and validate those by analyzing data, enabling the authors to think systematically about data (Strauss & Corbin, 1990, p.99).
flexibility, which in turn leads to business agility, impacting on costs and revenues, and ultimately competitive advantage. An example should illustrate the cause-and-effect relationships: Modularity (technical principle) permits the reusability and recombination of business components (IT flexibility), so that new business services can be constructed on demand at low costs (business agility), which in turn can be exchanged for money (increase in revenue), resulting in a competitive advantage.

Figure 3: A model of SOA effects on competitive advantage

Source: Own figure
3 Methodology

3.1 Research Method

To address the research question, grounded research methodology – developed by Barney G. Glaser and Anselm L. Strauss in the 1960s – was applied to data collected in the research process. Fischer (2007, p.123) defines grounded theory “is a process that allows theory to emerge out of the research material rather than being forced out of it by the use of a predetermined idea or theory.” Grounded theory is a research method that provides a systematic, comparative, and inductive approach for conducting research with the aim of generating theory (Bryant & Charmaz, 2010). Grounded theory methodology, therefore, connects broader, contextual aspects and their influences to the phenomenon under investigation through an abstraction of data and their reintegration into a theory (Holton, 2010). To derive a theory, an iterative process of moving back and forth between empirical data and emerging analysis takes place, i.e. data collection and analyses proceed simultaneously, whereas both data collection and analyses inform and streamline each other (Bryant & Charmaz, 2010).

However, researchers modified the grounded theory method, resulting in two approaches – the Glaser approach versus the Strauss and Corbin approach. More specifically: Glaser argues that researchers should start with no presuppositions, and should allow ideas to emerge from the data; whereas Strauss and Corbin advise to become familiar with prior research and using structured processes to make sense of data (Easterby-Smith et al., 2008, p.101). A grounded theory, according to Strauss and Corbin (1990, p.23), is one that is detected, educed, and provisory confirmed trough systematic data collection and analysis of data with respect to a specific issue.

Despite different perceptions, it is significant to note that both approaches possess common design principles: theoretical sensitivity, theoretical sampling, continuous comparative analysis, coding and categorizing data, literature as source of data, theory integration, and theoretical memos; the difference between these two approaches is the degree to which any element is adopted, instead of the substance of the element (McCann & Clark, 2003).
3.2 Research Design

![Diagram of research design]

**Research Question**

Grounded Theory Methodology

Stage 1: Preliminary Literature Review
- Theoretical Proposition

Verification and Validation

Stage 2: Open Coding
- Theoretical Sampling
- Constant Comparative Analysis

Stage 3: Axial Coding
- Theoretical Sampling
- Constant Comparative Analysis

Stage 4: Selective Coding

Grounded Theory

Comparison

Other Theories

How does a SOA effect on the creation of a competitive advantage?

Source: Own figure
3.3 Sample

The sample was limited to one IT architect – a full-time employee in non-profit institution familiar with SOA – and research documents. This IT architect represented the initial sample to collect the first body of data. The subsequent data gathering was directed by the theoretical sampling maxim pertaining to grounded theory: A process of gathering new data guided by emerging categories (Strauss & Corbin, 1990), requiring the authors to indentify new categories, while analyzing, examining, and filling out those; and, where necessary, recurring to data collection to broaden categories (Charmaz, 1990). For example, when the interviewee indicated that IT flexibility is an important causal factor to enable business agility, then subsequent data gathering was focused on this category, using research documents. Theoretical sampling persisted until theoretical saturation was achieved, that is, when no new data occurred relevant to categories and their subcategories, categories had conceptual density, and connections between categories were explicated and validated.

3.4 Data Collection

Data collecting methods included one semi-structured interview (Västerås City – a non-profit institution) and document analysis. Data were collected from the samples in a cyclical process. Before conducting the interview, the authors arranged a meeting to obtain general information about the key contact person and its institution. After the first-time meeting, the authors generated a rough interview guideline to carry out the interviews – the initial means of data gathering (first round of data collection). The interview started with open questions concerning the pre-developed paradigm model in order to obtain data about the widest possible range of aspects regarding inherent causal conditions respectively relationships and their associated consequences. When the respondents answer was insufficient, the authors asked new questions, which were based on topics thought about in advance. The interview were conducted in English and digital recorded. The later process of transcribing the interview was performed carefully with the aim to keep the conversation original. When transforming the audio recorded conversation into plain text, then certain dialogue elements – e.g. respond time, facial expressions, emphases or its flow – are no longer noticeable within the later process of analysis, thus might change the meaning or interpretations of entire phrases. Research documents were used to as secondary sources of data (second round of data collection). Research documents, surrounding SOAs business implications, offer ‘a rich
source of information, contextually relevant and grounded in the contexts they represent’ (Lincoln & Cuba, 1985, p.277). Documents were used to 1) gather data about already identified and developed categories and 2) gather new data about the paradigm model that were not gathered during the interview. Documents also ensured that the emerging theory was based on multiple sources of evidence.

3.5 Data Analysis

To generate a theory, the analyzing principle of grounded theory was applied: A process of reducing raw data into concepts that are determined to signify categories (Corbin, 1986). The categories were then educed and incorporated into a theory. Constant comparative analysis is the key strategy in developing a grounded theory (Corbin & Strauss, 1990), and signifies its flexibility and open-endedness (Charmaz, 1990): Data collection and data analysis occur simultaneously (Blaikie, 1993). In doing so, the authors were required to figure out categories and related properties. A constant comparative analysis consists of 4 stages (Glaser & Strauss, 1967):

- Contrasting occurrences applicable to each category;
- Incorporating categories and corresponding properties;
- Defining the theory;
- Writing down the theory.

This process continued until a detailed and abstracted substantive theory was generated. Coding commenced the process of theory evolution (Charmaz, 1990), and involved open coding, axial coding and selective coding (Corbin & Strauss, 1990). Fischer (2007, pp.258-61) states:

- **Open Coding** – is a process of identifying, naming, categorizing, and describing data related to the issue under investigation. For example, the authors may realized that SOA enabled modularity is discussed in the research material. The authors termed this as “modular SOA”. This category was then broken down into properties and related dimensions. Integration time might be a property of “modular SOA”, which can be dimensionalized as decreasing or increasing the time to integrate the architecture.
• **Axial Coding** – is a process of creating relationships between categories and between categories and associated subcategories, and examining categories with respect to their contexts, conditions, and consequences, enabling the authors to verify the relationships inherent in the theoretical proposition. For example, when the authors noted an indication of “modular SOA”, the data were analyzed in order to determine the context, in which it occurred, the conditions which caused it, and its consequences.

• **Selective Coding** – is a process of integrating the categories into a theory, related to the issue researched. This processed was achieved by choosing a core category to which the other categories – representing causal conditions – were linked, which subsequently created a theory, validated against data.

Coding procedures, memos, and diagrams were used to analysis the data. Data from interviews and documents were coded systematically. Memos are notes, which were made by the authors to record and explicate the theory. Diagrams were used as a visual representation of the conceptual relationships among categories.

### 3.6 Method Critique

Grounded theory methodology represents a qualitative research method. A key challenge within qualitative research remains reliability\(^{10}\), whereas replicating the research addresses the aspect of reliability; “The lack of replicability of grounded theory has been a criticism of the method” (Parry, 1998, p.97): The grounded theory process is dependent on the interplay between the researchers’ creative process and data, that is it is doubtful that two researchers produce an identical theory (Munhall, 2001): First, the researchers’ cognitive idiosyncrasies are inescapably implicated in the process of interpretation, thus will influence the interpretation (Berger & Kellner, 1981); and, second, no two situations are equal, because even within one research project, circumstances change constantly (Glaser & Strauss, 1967). In the context of the present thesis: Due to the character of semi-structured interviews, the interview schedule will be different for every interviewee – the same questions can’t be asked – and, furthermore, questionable is whether the same interpretations would emerge out of the research documents. Thus, full replication is difficult.

\(^{10}\) Reliability: “[…] demonstrating that the operations of a study–such as the data collection procedures–can be repeated, with the same result.” (Yin, 2008, p.40).
Since primary research data has been collected from a non-profit organization, the validity and thus analyses and conclusions of this thesis are primarily relevant for this specific organization, Västerås City. Adding secondary data, with the purpose of covering profit- and competition related information which could not sufficiently gained through the primary data source, represents general knowledge in these business related fields and could thus serve as suggestions for further improvements of the investigated institution’s SOA project.

4 Presentation and Analysis of Findings

4.1 Open Coding

The first part of the performed analysis is based on memo writing, which was done throughout the open-coding stage of the grounded theory approach. Information, mostly from the interview with Peter Mannerhagen (IT- Architect Västerås City), has been used to establish different heterogeneous categories. Moreover, subcategories consisting of properties and dimensions of those properties have been created, that are related to one category. As a result the categories of “Investment Reasons”, “IT-Flexibility” and “Business Agility” emerged during the open coding stage and are described in the following paragraphs. Since the primary source of the open coding memoing is a transformed interview, categories include few examples of property implications, which are discussed during the following stage of axial coding.

- **Category: Investment Reasons**

The first category consists of information and measurements related to general investment reasons of a SOA. “Empowering legacy systems”, “IT-cost reduction”, “Improved application development” and “Improved collaboration” have been identified as subcategories.

SOA leads to independence from legacy systems and their limited functionalities due to their silo based construction. One major goal of modern IT infrastructures is to combine systems through loose coupling, in order to gain a better user experience without losing any given functionality.

SOA enables the reduction of system complexities and thus their application density. A major tool to achieve this is the Enterprise Service Bus which connects different systems by providing a service layer among them. Improved application interfaces then allow that
functions from different systems can be joined together and therefore extend their functionality range. As a result IT-costs can be reduced through eliminating redundant IT functions, which do exist in several (now interacting) systems. On the other hand, IT complexity can still rise due to a continued development of systems based on more but inefficient integration with external systems.

Internal application development in a SOA environment is done with higher quality and in shorter cycle times. Developers have a better overview of the application inventory due to the usage of a registry that consists of all available applications, ordered in a specific way and thus helping them to create new applications more efficiently. Maintenance costs and administration costs related to a SOA infrastructure decrease mostly due to time savings as a result of better data quality and asset overview, granted through improved integration procedures and application registries.

As a result of loose coupling, reusability and integration tools like ESBs and registries, organizations are able to open their operating networks to partners and thus improving their collaboration regarding IT services. Accordingly reusability is used to enable the usage of the same applications in different locations. Most importantly the improved integration possibilities of a SOA allow opening up networks to external environments resulting in better collaboration.

Investing in SOA needs to be recognized as a long term investment, which concerns costs and benefits over time. However, investing in SOA is not avoidable if an organization owns a big and growing IT infrastructure, because otherwise IT governing costs will become enormous in the future. The payback time in the case of Västeras City is about 2-3 years (dealing with about 400 applications at the moment). The payback results from diverse cost savings, some of them mentioned above and additional creation through new e-services, respectively applications.

When SOA is fully implemented and therefore deployed as an enterprise infrastructure, then performing fast changes of business processes or immediate creation of new ones due to business environment changes are possible.
Figure 5: Open coding: Investment reasons
Source: Own figure

- **Category: IT flexibility**

The second category deals with IT-flexibility and contains the subcategories of “Loose coupling”, “Open standards”, “Modularity” and “Necessity to change”.

IT-flexibility through SOA is possible but demands changes on different IT related levels, such as architectural, physical and logical, as well as changing the way you work with IT.

Moreover loose coupling is highly important for IT-flexibility, since it allows system changes or modifications without many difficulties. This is enabled through the usage of Web services, acting as connectors between applications and their demanded data sources. Components like data sources can be exchanged or modified quit easily in a SOA, because only directly affected Web services needs to be adjusted accordingly. The rest of the application is still working without performing any change.

The same logic applies for the fact that modularity allows IT-flexibility. Web services are one example of modules or components. Rewriting or changing them due to system modifications or upgrades does imply that any application using them will get the changes
automatically, as well. Thus modularity enables fast changes within the IT infrastructure. In addition reusability of SOA components is mostly enabled through the feature of modularity. Moreover open standards are vital for IT flexibility. As a matter of fact vendor dependencies do not exist since components can be bought from any component manufacturer. Also open standards allow interoperability, which means that different systems can be connected to each other. This also includes silo systems, which are usually not designed to be open and share information. Accordingly open standards are used to enable the integration of legacy systems with the intention of exploiting joint functionalities. From a technical standpoint, integration is supported by Integration Engines and/or ESBs to assure a high quality data structure, and thus are important components to enable interoperability based on open standards.

Figure 6: Open coding: IT-flexibility

Source: Own figure

- **Category: Business agility**

The last category is named business agility and consists of the subcategories of “React towards change”, “Collaboration”, “Better system integration” and “Application development”. Some of the information used to establish those categories has been extended through literature since the interview partner could not offer sufficient data.
One form of business agility is the reaction to legal changes and thus regulatory requirements that need to be fulfilled by an organization and do affect its IT infrastructure. Other events that trigger the demand of business agility are reaction to business changes or business opportunities. Hence faster adoption to market change is also important regarding agility and is related to the strategic level of an organization. Linked to the technical features of SOA, loose coupling enables the IT infrastructure to react accordingly and in a fast manner, since changes do only have to be performed in few places and not for whole systems.

Open standards related to agility can be exploited through making use of enabled collaboration with partners and thus reduce IT costs for instance through shared development costs, -time, -risks, as well as increasing the overall organizational productivity. Therefore collaboration is accomplished through shared or exchanged e-services respectively applications with partners.

A further benefit of improved integration within a SOA is the decrease of system integration times. However, a conflict between vendors of IS, not developing according to open standards, and better integration exists nowadays. SOA is interested in the creation of interfaces to enable system integration and joint functionalities, but vendors often prefer to sell their systems as closed entities, hindering easy information sharing with other systems.

Application development cycles are generally improved through SOA. This included shorter development times, e.g. through asset reuse and better application- and interface transparency for the developers. Thus the time-to-market aspect for applications is reduced. Moreover, communication between the business- and IT representatives of an organization is supported through the service aspect of SOA, also resulting in better application development. SOA furthermore is able to foster innovation within an organization, leading to possible cost savings on the one hand, and revenue creation on the other. (Legner & Heutschi, 2007; van den Berg et al., 2007; Chen, 2008; Woolf, 2008)
4.2 Axial Coding

Throughout the stage of axial coding further information regarding the identified categories has been analyzed. Information emerged from intensive literature reviews regarding SOA with a focus on the linkage between IT infrastructure and business agility serving as a useful secondary data source (McCann & Clark, 2003). This procedure on the one hand was used to increase the description of subcategories and thus their internal properties and dimensions. On the other hand it aided the researchers to identify internal and external links between subcategories. Internal links are described as relationships or dependencies among subcategories of the same category, whereas external links are associated to dependencies among subcategories of different categories.

Figure 8 shows the identified external links among the three established categories, as well as the internal links of subcategories, which have been discovered during axial coding. As the figure illustrates, interrelationships between all categories exist.
A more detailed view on the recognized relationships is demonstrated in figure 9. All linkages between the mentioned subcategories emerged during the memoing procedures of the open coding and axial coding stages. This implies the usage of primary data, represented by the interview, and secondary data sources in order to generate the illustrated outcome.

The table consists of the three categories of Investment Reasons, IT-flexibility and Business Agility and their connected subcategories. A subcategory from the left side of the table does have an identified relationship to another subcategory if their matching cell is labeled: supports. The accentuated rectangles are visualizing the areas of possible internal links. Accordingly, any matching cells labeled: supports, outside of those areas are indicating external links.

For an even more detailed view on the established relationships, appendix II shows how the actual subcategory associations got created through various literature resources.
4.3 Selective Coding

As an important step towards the emerged final theory further category modifications and – refinements are performed during the beginning of the selective coding stage. Accordingly some subcategories have been recognized as containing homogenous contents to a big extent. Those subcategories are merged in order to reduce redundancies among the whole set of subcategories. The merging procedures results in the creation of new subcategories, which are still representing their intended meaning, but enriched with further details.

Thus, the subcategories of *Improved application development* (Investment Reasons) and *Application development* (Business Agility) are combined and result in the new subcategory of *Improved application development*. Also the subcategories of *Improved collaboration* (Investment Reasons) and *Better system integration* (Business Agility) have been merged under the new name of *Improved interoperability* (Figure 10).
After establishing the final set of subcategories a core category, respectively central phenomenon of the investigation was detected. According to McCann & Clark (2003; p.14), important features a core category needs to fullfil among others are its abilities to appear frequently in the data, helps to explain most of the variation in the data, can be linked easily with other categories and progresses the emergence of a theory forward. Also a core category indicates a problem, issue, an event or a happening that is defined as being significant to respondents (Strauss & Corbin, 1990).

With respect to this conditions the researchers decided to elect Improved interoperability as core category. As a result of critical thinking based on the information gathered during the open- and axial coding stages, the SOA characteristic of improving
system integration was identified as the key concept which enables further benefits. These benefits are needed to emerge the final theory and thus are closely related to the initial research question. Moreover recognizing improved system integration as a core concept of SOA, it can be linked to all other subcategories without difficulties (Figure 11). Consequently the subcategory of Improved interoperability fulfills the mentioned conditions of a core category.

![Figure 11: Selective coding: Core category and aligned subcategories](source: Own figure)

In order to investigate the relationships between subcategories and core category the paradigm model has been used. The paradigm model puts the core category into context with the components of causal conditions, context, intervening conditions, action/interaction strategies and consequences (McCann & Clark, 2003). Those components, aside from the context component, are represented by subcategories different from the core category, which fit the component’s characteristics. Figure 12 visualizes the applied paradigm model, regarding its components and assigned subcategories.
4.3.1 Causal Conditions

Causal conditions are recognized as events that lead to the establishment of the core category (Pandit, 1996). Throughout the analysis of the primary and secondary data it became clear that the technological concepts of SOA enable a new level of system integration. Thus causal conditions are referring to the subcategories of Modularity, Loose coupling and Open standards.

Open standards do enable more convenient system integration by being independent from any specific vendor. Thus, various systems can be interconnected as long as they are supporting open standards. The technical main themes of a SOA (modularity, loose coupling, open standards) enable the usage of ESBs and/or IEs, which are technical components usually existing as servers and granting the interoperability of diverse systems and applications. (Mannerhagen, 2010)

Having an IT infrastructure based on loosely coupled and modular components results in further benefits. Both SOA characteristics foster system integration since modifications or substitutions of parts or complete systems are easy realizable by not upgrading or modifying the underlying infrastructure (Mac Vittie, 2005). Accordingly changes of partial system components are applied to the rest of the system without further adjustments. Moreover the use of reusable modular components reduces the overall infrastructure complexity, since for example redundant system functionalities can be eliminated (Mannerhagen, 2010).
4.3.2 Core Category

*Improved interoperability* in the context of this study primarily refers to enhanced integration of information systems. As already mentioned before, loose coupling, open standards and modularity enable the usage of ESB or IE as fundamental parts of system interoperability and thus improved system integration. A further elementary SOA infrastructure component that is closely linked to the enhanced system interoperability is registries. A registry in the sense of SOA is a centralized repository comprising a directory of all implemented services of an organization. Usually it follows a specific grouping scheme, for instance listing services according to business units or business functions. Taking advantage of both concepts interoperability is based on a “find, bind, invoke” approach, where a service provider publishes its service description in a service registry which then can be identified by a service consumer. Binding is achieved by accepting the service’s interface contract. If accepted, different services can interact with each other through usual request and reply mechanisms of applications. (Mac Vittie, 2005; Minguez et al., 2009)

An Enterprise Service Bus is used in-between different services, serving as a standard-based integration platform. Therefore ESBs are responsible for interconnecting applications to each other and thus enabling data sharing. Accordingly the main task of an ESB is to deal with data transformation, respectively ensuring high data quality, as well as intelligent and reliable routing among organizational wide stored applications. (Mac Vittie, 2005; Minguez et al., 2009; Mannerhagen, 2010)

4.3.3 Context

Context in this case refers to “the particular set of conditions and intervening conditions, the broader set of conditions in which the phenomenon is couched” (Pandit, 1996).

As discovered throughout the analysis of primary and secondary research data, it is very common that organizations have difficulties regarding system integration because of the increasing heterogeneity of their ISs, which leads to the establishment of various information silos. Silos hinder information sharing among internal organizational applications and also external ones, which could be used to interact with customers or suppliers in a more efficient and effective way. Integration among silos and other applications is often problematic, since
they differ in technical specifications such as their use of operating systems, programming languages or database approaches. (Minguez et al., 2009; Mannerhagen, 2010)

In particular legacy systems are often considered as silo systems by consisting of fundamental organizational information which is not intended to be shared. However, in order to incorporate and thus exploit the information flow of all business processes within or beyond organizations it is essential that information usage and exchange is enabled regardless of any technological specification or dependencies of underlying systems or applications. Since SOA deals with the transformation of business processes into reusable, loosely-coupled services, it enables easy adjustments or creation of business processes and thus system and application integration. (Minguez et al., 2009; Mannerhagen, 2010)

4.3.4 Intervening Conditions

“Intervening conditions can act as facilitator or constrainer regarding the action/interaction strategies in a particular context” (Strauss & Corbin, 1990).

As a result of the discussed causal conditions and the core category of Improved interoperability the subcategory Empower legacy systems has been selected as the primary intervening condition. Based on the previous components of the paradigm model, silo information systems are enabled to act more open regarding their information flow. Thus silo systems can be connected with other systems or applications, for instance performing joint functionalities. As a result their range of functionalities is enhanced. Moreover system information can be shared and exchanged within and beyond the organization. (van den Berg et al., 2007; Minguez et al., 2009; Mannerhagen, 2010)

The second subcategory assigned to the stage of intervening conditions is Necessity to change. In order to exploit the benefits of SOA, organizations need to realize that they have to change their usual IT-related operating and working routines. For instance SOA affects the architectural, physical and logical levels of IT. (Mannerhagen, 2010)

4.3.5 Action/Interaction Strategies

Action, respectively interaction strategies of the paradigm model refer to the actions and responses that occur as the result of the phenomenon (Pandit, 1996). Moreover they are strategic or routine responses connected to the core category (Strauss & Corbin, 1990). Thus
the subcategories of *React towards change* and *Collaboration* have found to be matching those requirements.

As a matter of fact, collaboration is depending on the access availability of information. The traditional silo-based form of organizational applications avoids the desired collaboration among employees, as well as with partners and customers. Since SOA overcomes this drawback of traditional IT infrastructures, it fosters internal and external collaboration. A result of internal collaboration for instance is enhanced organizational productivity enabled through better decision making as a consequence of improved information availability. External collaboration in a SOA environment can be achieved in cooperation with partners such as suppliers. (van den Berg et al., 2007)

Development- and/or final products of e-services could be shared or exchanged resulting in decreased development efforts such as -costs, -times or –risks (Mannerhagen, 2010). Moreover collaboration is eased through the abstract service principle, which clearly points out business functionalities of applications. Since service registries follow a certain standard (e.g. UDDI), their listed information is not only helpful inside an enterprise, but also with respect to business partners, such as suppliers, by allowing them to better integrate their own applications (Mac Vittie, 2005). Also relying on the interface standards of Web services, integration among collaborating parties can be performed more easily and flexible (Homann et al., 2004).

Thus collaboration in general allows faster adaption to altering market conditions and therefore enables organizations to act more agile, especially from an IT infrastructure point of view. Well working system integration is one key element of achieving business agility and an interoperable, adaptable IT infrastructure. Therefore SOA enables enterprise applications to correspond to dynamic market and its altering demands, constraints or opportunities. In this sense referring to the loose coupling ability of IT infrastructure components, existing services can be upgraded or replaced very efficient when required, avoiding negative impacts on business operations to a big extent. (Chen, 2008; Minguez et al., 2009)

Accordingly, reusable IT assets can be exploited towards an improved reaction towards change. Within a SOA infrastructure only one single instance of a certain business function exists, which then can be reused in different applications. Consequently, any change concerning the unique business function will be present in the linked applications as well. Thus market condition changes, for instance resulting in different pricing or discounting rules, can be managed easily by applying altered business logics to multiple applications at the same
time. Therefore less time is demanded to implement changes in general and a fast reaction to dynamic market conditions is granted. (Mac Vittie, 2005)

4.3.6 Consequences

“The consequences of the paradigm model refer to outcomes, both intended and unintended, of actions and responses” (Pandit, 1996). Thus the subcategories of Improved application development and IT-cost reduction seemed to be most suitable for describing the overall outcomes of previous paradigm model stages. Any outcome should be related to business benefits of SOA in order to serve as valuable arguments towards answering the initial research question.

One aspect of gaining business benefits through SOA usage is based on how well the lower level of IT infrastructure service can provide the intended business level customer service and thus increases an organization’s performance (Brodkin, 2007; van den Berg et al., 2007; Chen, 2008; Woolf, 2008). Even though, definitions of a service varies from an IT-respectively business standpoint, which has caused many misunderstandings regarding past IT projects, service orientation offers new opportunities for companies to accomplish better outcomes of their designated IT- and business alignment. As a consequence of improved collaboration SOA facilitates easier communication between IT and business representatives, since a general terminology is shared by both parties and potential application demands of service owners or users can be linked to underlying technical processes, in terms of services instead of IT projects, without much complication. Users know what services they desire and developers can offer a quick solution. (Legner & Heutschi, 2007; Chen, 2008; Haines & Haseman, 2009)

Throughout this study it was revealed that SOA can offer a wide range of cost saving possibly to organizations. One of the most important ones is the ability of reusing already coded services in combination with standardization, leads to a reduction of general development costs and operating costs, for example through a reduction in application redundancy and reduced development cycles. Because once a SOA infrastructure is in place, developers are able to reduce project durations and the time-to-market aspect for new applications by taking advantage of the loose coupling and reuse abilities of infrastructure components. Those time savings accordingly lower product- and service development costs. (Legner & Heutschi, 2007; van den Berg et al., 2007; Chen, 2008; Feig, 2008; Woolf, 2008; Mannerhagen, 2010)
Secondly, transforming the IT infrastructure of an organization towards a services-oriented solution decreases the existence of large, complex and monolithic systems (van den Berg et al., 2007). Thus system changes can be done without time consuming analysis-, planning-, and testing phases. Adjusting single services of a SOA therefore increases the efficiency of system changes regarding the required time and effort, since the IT/IS landscape as a whole is not affected by it (van den Berg et al., 2007; Brodkin, 2007; Mannerhagen, 2010). Hence SOA supports the organizational ability of decreasing operating costs as a result of less complex IT maintenance (Legner & Heutschi, 2007; van den Berg et al., 2007; Woolf, 2008; Mannerhagen, 2010). Furthermore, standardized interfaces among applications result in cost savings due to more efficient system integration (Legner & Heutschi, 2007).

In addition reusability is connected to the possibility of more effective monitoring and management of IS, for instance due to less software support and maintenance demands (van den Berg et al., 2007; Woolf, 2008). This is because redundant service capabilities can be eliminated and less code is needed to execute the same application portfolio as in a comparable non-SOA-based environment (Brodkin, 2007; van den Berg et al., 2007; Mannerhagen, 2010). Moreover IT management is eased through higher transparency regarding application dependencies and application interfaces. Therefore the decreases of IT complexity leads to various cost savings (Legner & Heutschi, 2007).

But not only does SOA reduce costs, it also enables the creation of additional profits. The composition of once-implemented, reusable, individual services can be used to generate additional services, providing new functionalities for service users without much effort and possibly resulting in new revenue streams due to changing business environments or upcoming opportunities (Brodkin, 2007; Chen, 2008; Woolf, 2008). This is furthermore related to act more innovative as an organization. Therefore SOA adoption does not only allow diverse IT benefits, it also serves as a business innovation enabler for instance by adding new business processes to support shifting customer demands or enhancing organizational business processes (Haines & Haseman, 2009).

5 Conclusion

5.1 Conclusive Discussion

The focus of the thesis lays on how SOA effects on the creation of competitive advantage. The underlying theoretical assumption that was supposed to be validated throughout the
research process is defined as followed: Costs and revenues are indirectly affected by SOA facilitated IT flexibility. Moreover modularity, loose coupling, and open standards permit IT flexibility, which in turn leads to business agility, impacting on costs and revenues, and ultimately competitive advantage. Therefore SOA has been investigated from different angles using memoing and the paradigm model as a powerful tools of the grounded theory approach.

One major finding of this research deals with SOA as an enabler of improved interoperability, respectively enhanced integration of information systems. Its importance derives from the common situational issue organizations have to cope with: a complex legacy system infrastructure, which lacks IT flexibility and thus hinders business agility. SOA offers a flexible IT infrastructure, which consists of improved interoperability as a key feature, based on exploiting its technological characteristics of modularity, loose coupling and open standards.

Throughout the analysis part of this thesis it has been confirmed that SOA enables organizations to respond rapidly and effectively to business change from an IT point of view. In case SOA is implemented as a complete enterprise IT infrastructure solution, it allows on-demand IT environmental changes through its flexibility. Furthermore it has been found that a main aspect towards business agility deals with the transformation of silo system regarding their service application. Within an ideal SOA, silo systems basically don’t act as isolated systems anymore since their information is made accessible to other systems. Thus their containing service functionalities are able to be used for collaboration purposes by linking them to other services within or beyond the organization. Another important part of SOA based business agility has been identified as the ability to perform quick modifications of existing business functions based on modularity and reused IT components.

As described earlier organizational competitive advantage can result from 1) business agility in terms of reacting fast and efficient towards environmental changes trying to develop a competitive advantage, e.g. through differentiation or 2) providing products and services at lower cost than competitors. Both approaches can be linked to the identified business benefits of SOA, resulting from achieved IT flexibility and business agility:

- Improved IT- and business alignment
- Improved creation of new application
- Reduced application development costs
- Reduced application development time
- Reduced IT operating and maintenance costs
• Reduced system integration costs

In order to act agile and take advantage of market changes in an efficient and effective way, a well-established alignment of IT and business goals can be supportive. This can include the ability to act more innovative based on better communication between business and development representatives or simply reduce time-to-market of new projects, which can result in either cost savings or additional revenue creation. Moreover application creation in general is eased within a SOA environment due to modular, reusable assets that are able to be interconnected in different ways without much effort. This could result in the creation of new services that fulfill the demands of additional customers or new market niches and thus lead to increased revenue and competitive advantage.

Although SOA enables revenue creation, it emerged throughout this study that most business benefits are linked to cost savings. Concerning application development, cost savings are mostly achieved by shorter development cycles. Therefore application development can be performed more cost efficient, creating more services in a specific timeframe than a non-SOA based infrastructure would accomplish. The second big cost saving potential derives from the creation of a less complex IT infrastructure. A service-oriented architecture does decrease the complexity and of large, monolithic systems. The flexibility of SOA infrastructures can be linked to fewer costs of IT support and -maintenance, -operating and -system integration. Nevertheless it should be noticed that gaining all those benefits are developed over time, since establishing a SOA is about performing expensive changes to organizational IT on different level, which makes SOA a long-term investment.

5.2 Future Research

Since this research follows a grounded theory approach which is initially based on information from a government institution, it seems appropriate to extend the primary data collection regarding several profit oriented, respectively private organizations. By doing so, secondary research data usage could shift towards a more supportive role since more primary data would be available as a strong basis for performing the different coding procedures. Thus the emerging theory might be developed more independently and in a less literature-biased way than presented in this thesis. Moreover, categories and subcategories might be refined or altered due to further available information, resulting in better research outcomes.
Future research based on the findings of this thesis could also be done by performing accordant quantitative researching in specific or different industry areas. Thus it would be interesting to carry out a validation of the emerged theory by testing it against empirical data. Including a large sample size of investigated organizations on quantitative business benefits and competitive advantages of SOA would be supportive in order to attempt generalization of the findings of this research project.
1. General Questions:

1.1) What are the main reasons for investing in SOA?

- Reduce complexity
- Reduce IT costs
- Innovation
- Improve business processes

The main reason is to establish an IT environment that is independent of the old systems that we have today. Our different businesses are today really dependent on a specific system. Instead of being dependent on the function that the systems deliver (is right?!), we want to establish a SOA infrastructure to get the loose coupling from the user experience to what the systems deliver. If we can generate this loose coupling with an integration interface to one or several underlying systems, then we can generate better use to our users and we can also establish independent because of the loose coupling to the underlying system. We can replace every system individually without losing functionality, the user and...

MEMO NOTE: SOA leads to independence from legacy systems and their limited functionalities due to its silo construction. Combining systems is the goal, through loose coupling, in order to gain a better user experience and not losing functionalities.

To reduce complexity let me draw a picture... If you have a lot of users out in the environment using different kind of data, usually they use one system each, like a silo. If you don’t have an ESB, then you have a silo... let me show you a power point. One application is connected to one DB and that’s the big problem.

PPT: You have a system A, it has got its own data and you have a user connected to a network and this in an e-service scenario where you have citizen using any kind of web application connected to an e-service on system A. so you got a silo, right?! And of course you have lot of more systems and certainly a lot of more users with different needs and that needs are laying in one or more of these systems. So, you establish this service layer. Part of the service layer is the ESB. This is the solution we want to establish. Get a lot of better functions because of this of course, reducing costs and we get a user interface that you can control and develop in a good matter. And in the long run you can disconnect some of the systems, because you will find out that system A and system B are in fact the same system, using and delivering the same type of data like planning systems or project planning systems. If these are two existing project planning systems than you will find that the underlying system here could very well serve the user of this system as well. So you can reduce the amount of systems you have in your environment. But this is not related to reduce complexity, but reducing IT costs in the long run, though. [Showing next slide]This is the worst scenario if we continue using the silo infrastructure where every e-service got his own system, own database than we would have an environment that is growing like a virus. [Next slide]That picture shows how we want to establish our future IT environment.
MEMO NOTE: Complexity issue – Systems (and thus their applications) can be connected through a service layer (ESBs). Better functions derive from this, further application development is improved. Costs can be reduced through eliminating redundant IT functions.

1.2) Why is system complexity not a factor?

[Drawing picture] This is a user (could be a citizen), this user has any kind of application connected to some database. If we want to put in some extra functions in this app (for example establish a secure ID information from the tax government), then we would had to complete it with some integration with the tax government. We have to make that connection here, probably go outside our net to the tax government. And then we would find that we would establish another function, could be some kind of integration with another system, containing relevant information of course. So we make that connection. So far the user is very happy because we are matching up to his new demands all the time. If we continue this development which every system has (continuous change all the time)... if take into our aspect here that we have around 400 systems and this upscale with new integrations all the time makes every system very hard and very expensive to govern (have a government for), so what we can do to reduce complexity is for instance this function of the tax government, instead of letting every system that needs that input go to a central source that we share as a web service, that web service has a function to the DB, that connection makes sure that this information that all the systems require is stored locally, and there is one function that is collecting this data. And this function (web service) is connected to the ESB. And this application, same as every other application that will need this service, is connected by a web service to the ESB. So every application who needs this type of information, does it with this web service, instead of every application remembers up to 100 with this same demands should have their own coupling to it. Because it is reusable, it is a coupling to the tax government, the other one is not. And that’s reducing complexity.

MEMO NOTE: Complexity can still rise due to continued development of systems and more integration with outside world system which is not done optimally.

1.3) The core of reducing the complexity is the ESB then? Actually the connection between the application and one kind of DB, that every application that needs this information can connect to.

Yes. But not surely DBs, it is a type of data. If we have the tax government as an example they have a very large DB containing every citizen in Sweden. With birth dates, marriage status, age... So this data is named truth... so it would be very poor solution if we have applications or systems in our environment where we type this data in manually. Instead of having all this systems talk to the text government, we can make the systems with this web service... and this web services knows how to contact this web service (which is connected to the data).

1.4) So every other data can connect to the same web service in order to get the same data when needed?!

Yes.
1.5) Web services could be connections to the outside but also could be connection to the provider of the data for example.

ESB and web services (for connection purposes) must exist in a SOA environment. [...] If we think this through in user scenario, this user want to fill in this form in this web application. Request for child care, to do so he must type in his personal number, when doing so we have this function in the application to check if this is the right person. To do so, the application goes with this web service over the ESB to this web service to get information from the tax government and displays information from there in the application.

Web services are necessary parts of SOA. Its not necessary regarding a SOA definition. But it helps a lot when you should implement it. Most SOA implementations are with web services. As I mentioned last time, we haven’t establish the ESB yet, so far we use web service to web service connections. This is more flexible than a direct coupling.[...] If we have a web service than we can change this web service regarding the application need, you can change this web service if this systems has an upgrade. That is a kind of loose coupling. Having an ESB you have a full loose coupling. That will make this WS unaware of this WS. The only thing this WS knows is where to find this WS by going to the ESB with a flag or an address, bringing its own data with it, putting it in here as a request, this WS isn’t aware of any other systems, it just waits for someone making a request.

1.6) Is the WS able to find the right DB or is it some kind of predetermined, that this WS is connected to this other WS?

No it is not connected. How this WS would find this one? Ok, this WS is connected to the ESB. Every WS has its own address. That would be in any kind of register connected to the ESB. Every WS would have an awareness of all the addresses that are published on the ESB. So when you have a request coming in this way, this WS will have a check in the registry and go the way the registry tells him. So that way you have for every instance of action you have direct coupling. But as soon as this coupling is made then you have none, right?! So, that’s loose coupling.

MEMO NOTE: The registry in the ESB allows loose coupling

1.7) What do you think about the statement that reduction in IT costs is a reason to implement SOA?

Yes on of the reasons, but the reduction of IT costs lies ahead. To enable reduce of IT costs you have to invest in the infrastructure. So you have that kind of curve (pointing out S curve).

MEMO NOTE: long term cost benefits through SOA

1.8) What do you think about the statement that Innovation/ creating new services or apps are reasons to implement SOA?

SOA is an enabler to make new e-services to our citizens in a faster way. In a more secure way. So it is definitely working as an innovation more to our different departments in Västeras City. They are enabled to control more of the development of their e-services. There are different departments responsible for delivering their responsibility against the citizens. And they can more easily see how the e-service, or any change of already established e-services, can make better use of them. On the one hand for the citizens of course, on the other hand can the departments see more easily and quickly how the e-service can establish savings in the ordinary administration. There are saving maintenance costs but also the day-to-day administration work decreases. If we establish an e-service that increases the data quality regarding a request from a citizen and make a tight connection to their system that they work on daily, then they can spend less time for checking the data and instead focus on the request of the citizen. That could lead to, that handling the request could be done with fewer people. Time is available for other things.
MEMO NOTE: Application development is done with higher quality and faster. Developers have a better overview of the application inventory to create new ones. Maintenance costs and administration costs decrease mostly due to time savings as a result of better quality and overview.

1.9) [30:30] How would you refer to Business processes?
We have done quite a lot of work with defining our.. we call it business processes or work process. And a work process could be for example handling applications for child care for instance. From the point that the requests come to the government to deliver response to the citizen, …there we have a work process right?! We have made a lot of WP handling. There are a lot of WPs that are made and designed, we have a system where we put all those processes into. This system supports reusable parts of processes as well. So if we have a segment of a process that is for instance buying a PC, if this process is already designed and used by one of the departments other departments design their own process of buying a PC they could use that part of the process.

MEMO NOTE: Reusability used for same applications in different locations.

1.10) Please explain if this does refer to some kind of reusability in your opinion?
Yes, absolutely. The system also supports the ability to, out of the business processes that are in the system, there is the possibility to make a connection to some part of the infrastructure, for instance different parts of the SOA infrastructure, the WS are reusable and you can make connection to for instance this WS that is recoupling to the tax government… If you have a WS that needs personal information you can have a coupling to that WS in this system. That’s just a logical coupling not a physical. And if you make any changes in the work process than the changes you do to the work process regarding to that part that has the coupling to this WS could also make the change in WS.

1.11) [35:30] Business process explanation, as a sequence of producing something. Consumers need the change immediately. Can you fulfill this customer need, by performing fast changes of a business process or even to create new ones immediately?
Fully implemented it would be possible to do that. What you are getting close to is what you can achieve when having a complete enterprise IT architecture. You know what that is?

[John Zachman framework of enterprise IT explanation follows, useful for handling changes, interactions within the organization are made visible.]

MEMO NOTE: Fast changes can be implemented if the SOA is fully implemented as a basic infrastructure of an enterprise.

2) 50:30 Profitability is the difference between revenues and costs. Lower costs than rivals and/or the ability to command premium revenues result in competitive advantage. From the information given:
2.1) What do you think is the cost factors impact of SOAs? And what do you think is the revenue impact of SOAs?
We have a situation where you have to invest to get SOA, after that you start to earn money. On the other hand if you don’t implement SOA, well you will end up with very large costs in governing the e-services and the connections that would be in very large numbers, so that’s the big problem, to make those who handle the money up here, they must understand why they should give us money to establish SOA in order to save money in the long run. I haven’t got any figures that would point out revenue of establishing SOA, I’m sure though that we in let’s say 2-3 years after establishing an ESB connection we would start earning money with every new e-service that we connect to the ESB. The economic foundation; you can see in an government like VC, you have taxes as income, this money comes into the financial department where they split up the money for different parts of our business. […] For IT its different, we don’t get any money from the central financial department. We deliver IT
services to VC and they pay us for the services. And the prices that we set for the services we deliver are in fact around 0 in profit. We can’t use market prices for services to sell them to our colleagues. The citizens don’t have to pay for the services. […]

MEMO NOTE: SOA is a long-term investment, and not avoidable if you have a big IT infrastructure. Payback time in this case is achieved in 2,3 years (400 applications?). Payback results from diverse cost savings and revenue creation through new e-services.

2.2) Could you explain the situation you are dealing with in terms of any competition with or from other regions?

We sell some services to Hasterhama which is a region nearby. Those are not connected to directly to our SOA. They can use the same applications but not the same data, that’s a legal aspect. You can’t use the same DB because we don’t serve the same citizens. But we can deliver the same IT services. It’s not SOA related at all. We only deliver server management; their server is in our care. Their network ends up in a special part in our server hall. They access their servers from Hasterhama. […] mail clusters run in different instances of same server(s)

What’s possible if we do a connection to SOA, if we have SOA fully implemented we could use this functionality that’s connects to an external data source also to Hasterhama, also by defining a specific WS that is talking to their application. So this part and all on this side of the ESB is reusable for any other customer.

MEMO NOTE: Open up your network through SOA is possible, better integration, better collaboration.

2.2) Could you explain if there are any plans regarding the future for doing so?

Not panned, by recognized the possibility to do it. What is planned is, to establish regional (federal state) federation between all the different municipals and establish central functions as for instance SOA infrastructure and centrals function, for instance, personal information. So that’s something that is happening this year.

MEMO NOTE: see above

Technical characteristics and IT flexibility:

3) 1:03:10 Flexibility emphasizes the ability to quickly implement change. Flexible IT infrastructures deliver the technical platform, services and specialist resources required to deal quickly with unpredictable changes in the business environment. From the information given:

3.1) Which technical SOA characteristics affect IT flexibility and how do they achieve it?

- modularity affects IT flexibility?
- loose coupling affects IT flexibility?
- open standards affect IT Flexibility?
  - Interface- & application standardization
  - Reusability
  - Improved integration
• Improved transparency
• Decreased complexity

It is possible but to do so you need to have some kind of framework to handle change in a lot of different levels.
To handle change, for instance if you have change implemented in how you do integrations between 2 systems
you have to perform change in the architectural level, in the physical level, logical level and this must
correspond, otherwise you will fail. We use a framework called ITIL (IT infrastructure library) and its specific
about how you handle an IT environment in a secure way.

MEMO NOTE: IT-flexibility through SOA is possible but demands change on different levels

3.2) [1:05:48] Is that some kind of IT government?
Yes, that framework is vital to handle any larger IT environments and definitely it affects how you use changes
and can establish the flexibility when you have SOA. SOA enables flexibility, but you can’t have it if you don’t
use any kind of support of how you should implement it.

3.3) [1:06:50] But which characteristics of SOA do really lead to IT flexibility?
The most important one is the loose coupling that leads to flexibility. Why it’s done is just because, this WS
(referring to drawing on board) is aware of this system.

And its connected here, and (as pointed out before) there is kind of repository that knows that this WS is
available. This WS is connected to an application and the app knows that it needs data from somewhere. This
WS knows through this repository where to get it. The flexibility part is when you have to implement some kind
of change, let’s say this systems runs out of time and it’s not available for any upgrades anymore […], then you
have to buy another system to support the government (tax). What you know is what this application needs to
continue with delivering information to the user. You know from this WS where to get it. We also know what
kind of information the application would need. When we buy another application we could buy an application
that it should correspond to the needs of the system. Most of the time requirements are not met 100%. So we
make a small change where the demanded data should be collected. Because there is most likely another kind of
DB with another kind of structure of the information. You would need to do some changes here, but you could
do the changes to almost any system / you could buy almost any system to make the connection here.

MEMO NOTE: loose coupling is highly important for IT-flexibility, allows system changes without much
difficulties
3.4) [1:10:32] So, the most important is loose coupling? Yes

3.5) What about modularity? Change and rebuild as a response and as contribution to IT flexibility? Yes, absolutely. With the same example...

But let’s say we have the need to make a connection to another DB, containing some kind of specific information, so we have a WS with that information. These 2 WS could be a module of… let’s say… this application has a need of a total amount of data to deliver the information to the consumer. To deliver that total amount of data you have the segment of data which is generic. One kind of data that is generic and very frequently used is user information. Most of the e-services use some kind of user information. So that is this WS that supplies the user information is a component, a generic component for every e-service to use. If we had the federation (like talked about before) established already, we have a local copy there of all the citizens in the whole federation than we should make this connection to that DB instead of going to the tax government and the federated DB should have this connection instead. So, we rewrite this one module, and all the e-services that use this module will have the changes done instantly. No other changes are needed. That’s one of the big benefits of using modules or components. And the components physically are WS or small applications.

MEMO NOTE: Modularity allows IT-flexibility, fast changes possible

3.6 [1:14:27] What do you think about open-standards affecting IT flexibility? OS is vital. If we didn’t use OS we would be in big trouble very soon. The big benefit of using OS is that you could use any kind of partner or manufacturer of components. The only thing that we set the demands for is that they should follow an OS. Then we know that when we get a component deliver, it will work.

MEMO NOTE: No vendor dependencies through OS, thus IT-flexibility

3.7) So that means OS enabling the interoperability? Could you explain this further? Absolutely. Between different systems and between different governments. We have to interact with different governments, for instance some of the responsibility we have in VC is we have to deliver health care for senior citizens. In this responsibility we have a certain kinds of people involved. Doctors, that prescribes medicine for elder people. To do so they need to have a connection to the drug stores to deliver the recipe for the medicine and this connection must be as a defined standard. We have to be very fine that the doctor is the real doctor to prescribe the medicine.
MEMO NOTE: interoperability is a part of IT-flexibility, enabled through OS. Allows to connect different systems (from different locations). Also silo systems can work together using OS.

3.8) [1:17:00] Does that also mean that heterogeneous IT systems can work together using OS, also related to different programming languages?  
Yes. There are 2 big programming standards, Java and Dot Net. If you use OS you can make those two talk to each other in a very good manner. We have this far only used Dot Net, which makes its easier if we use Dot Net standards. […]

3.9) [1:18:40] Could you explain if and how this would result in improved integration?  
Yes. Let me draw again something. If we have a lot of legacies here and we have a need from any application mainly getting their information from this system, make a connection to this system to get some of the information here to make this system work better. And we will need some kind of integration environment. So again I I will draw the ESB, but the ESB and the WS who is the simple use in a SOA environment for making requests and replies… a part that we haven’t touch yet is an integration engine. An IE we use to make more complex data handling. It is a server, according to an applications need, it could get data from another system, it could restructure the data in another structure to deliver it in manner that another system could read it.

MEMO NOTE: Integration of legacy systems in order to exploit joint functions. Integration Engines /ESB are used.

3.10 [1:22:29] Isn’t data transformation related to the ESB?  
Doesn’t need to be, it could be. We have set it up as a service besides the ESB. We can draw it as a star, and this one (ESB) is a pipe. If you have a web service and we have end-user talking to an app, which has a WS.

So the scenario could be that the user is in the app and the app should deliver quite a complex data to the user. Some of the information is strictly read directly from another system by the WS and ESB but part of the data is information from this system and this system (both lower right ones) and the workflow before you could deliver this data is to check the data from system against this system to get a new data structure.

MEMO NOTE: ESB/IE checks data structure, assures quality. Important for advanced integration.
3.11) [1:24:55] *What do you mean by data structure?*

In a simple way it could be, lets say you have a field about containing an address. An address could be Storagatan, and the short form for this could be Storag. This system could contain street information which is a combination of both possibilities. Before we publish information up here for the user each street address from the system comes up the IE and compares to if we have the spelling Storag. We should replace it with Storagatan. When you are done so, you can deliver that information string. So what you are doing is you improving the quality of the data for the user […] by dealing with inconsistency of the data due to errors or laziness. […]That’s what’s the IE does.

The information will come from this application to the WS but it would connect to the WS on the IE and the IE will orchestrate the whole scenario. The whole scenario will be reading information from here, reading information from here and collating the data before delivering it to the application. This is an important part for advanced integration. Some of the IE contain their own ESB. Both would be a server, and there would be a logical connection between these servers. An ESB could also act as an IE, you don’t have to have them separated.

3.12) [1:31:50] *The IE would be an important component to enable interoperability?*

Yes.

3.13) *And that’s connected to open-standards?*

Yes.

3.14) *What’s the underlying technical structure?*

An ESB could be one or several servers. We are using several servers. Why? Well, we have an IT environment that is divided into several active directories with secure boundaries or in MS-language it’s called forests (maybe fortress?). We have one forest that contains our server resources. We have one forest that contains all the users and if our users should be enabled to use ESB they have to connect on the ESB on their own forests and make a connection between the forests on the service bus level. And that way we only have to make one opening in the firewall, because the forests are divided by firewalls.

3.15) [1:34:16] *And that makes the whole thing more secure?*

Yes. We also have a forest called DM center (DMC). This is part of the data environment that connects the internet, that’s where we have our public servers and so on. We also have an ESB in that forest. So in that way we make a secure connection from where our resource net where our server resource are to the external forest.

3.16) *Which makes it much harder to attack?*

Yes.

3.17) *Ok, so otherwise you would have several openings instead of one?*

Yes. Because every WS in an ESB has an address where it is about to get his data. Every address, on a low level communication data, must go through parts of specific parts of the firewall and if you have a lot of WSs that would be a lot of holes in the firewall. So we made the logical connection between the ESBs instead, letting of the WS talk through firewalls to one physical ESB. And that makes it more secure.
**IT flexibility and business agility:**

4) Business agility emphasizes the ability to respond quickly and efficiently to changes in the business environment and to leverage those changes for competitive advantage; thus, change requires fast responsiveness, but also provides the possibility to gain a competitive advantage. From the information given:

4.1) [1:37:20] How do you think SOA enabled IT flexibility affects business agility?
- Improved collaboration
- Decreased development time & costs
- Act more innovative
- Better adoption to market
- Better IT/business alignment

For VC, business agility could be for instance responding to a new legal demand and make a change in our IT environment that corresponds to that demand. That is easier to do with SOA, depending on the benefits of the loose coupling. Because you don’t have to make the change all the way. You make the change at one place and then all these services that uses that system can benefit from that.

MEMO NOTE: agility as a reaction to legal changes. Loose coupling is helping.

4.2) [1:38:50] What do you think are the kinds of business agility resulting from SOA enabled IT flexibility?

Another aspect as well is, that if we use OS and other municipals in Sweden use OS as well we could develop e-services and exchange e-services with each other. In that way really reduce costs over all (over the whole country), instead of letting every city buying or establishing their own e-services.

MEMO NOTE: OS foster collaboration and cost reduction.

4.3) [1:43:42] Is that related to improved collaboration through SOA?
Yes.

4.4) So that can result in decreased costs because you can share development time and –risk, etc. and also you can come up with new things faster and create new revenue streams.
Yes.

4.5) What do you think about the following keywords in connection with SOA and business agility:
- Act more innovative
- Better collaboration
- Better adoption to market
- Better IT/business alignment
- Decreased system integration time

Some parts are not applicable to a municipal. To be innovative, we try to do that. We are somehow restrained to be innovative. Because we have to follow the law, for instance and thus cannot be innovative in some areas. We
couldn’t change the manner of how we conduct health care for senior citizens for instance. There are legal
demands, about how to do that. So we cannot be as innovative as a private company can be. But we could be
more collaborative like the private sector because we don’t have any competitors. Instead we have collaboration
between us and Eskilstuna for instance sharing an e-service if we have produced it ourselves. That’s of course
not possible if we bought a product (legal issues).

Better adoption to market… In this part the municipals of Sweden really compete. For instance, there is a price
delivered every year to the municipal made the most improvement to the e-government. For us this is the thing
compete in.

4.6) [1:47:37] Does everybody care about this?

No, it’s only the larger city who care about it. It is more important to the strategic level of the city and its image.
No extra funds, but it is easier to get extra funds if you have proven yourself to be in the frontline of the
development. I think on this aspect we are as competitive as the private sector, on a different lever, though.

Business alignment; There are some who wishes that business alignment would be a greater issue in a municipal
but it isn’t today. Hopefully it will be in the future. Today, this has a low priority.

Decreased system integration time; of course with SOA it’s much easier but it also sets different kind of level of
demands to those who deliver the system. The most of the companies who delivers and IT system, deliver a
complete system, they deliver their own silo. What we are interested in is the user interface and WS to connect to
our SOA. And that’s a complete different kind of demand, and unusual today. Companies who deliver today are
not used to this kind of demands; they rather not want do deliver that way. That’s their business model. There is
a Swedish network… a government called SQL (series…..) that is Swedish municipals and health care. One
should think it is quite a big organization, but it isn’t. There are several networks that are held by SQL and in
those networks, we try to make this SOA infrastructure become more … it should have the same fundamental
rules in all municipals installed. So we will make one implementation structure that should be in every municipal
in Sweden, just to make the costs lower in the future and the possibility to exchange applications among each
other. So that’s the big work which is conducted outside of Västeras. […] Establishing these federations is the
focus right now. It is more easy to establish SOA infrastructure if you could divide costs among several
municipals and you could that way combine the forces of the …?… (members) in each group. Besides this
federation establishing here in Västmanland (state area of Västeras) we are planning to take this SOA
fundamental use for SOA establishing to be ready in somewhere in 2011. […] we are one of the top 10 in
Sweden…]

Well if you are planning to establish SOA you definitely have to change and this is not just apply on IT, its
applying on every part of the company. You have the change the way our work, and SOA will change the way
you work if you establish SOA correctly. SOA is one of the most important changes in the IT environment you
will see.

MEMO NOTE: Collaboration through shared e-services. Fast adoption to market is important for the strategic
level, also image creation. Conflict between vendors of IS and better integration. SOA is interested in interface
creation; vendors want to sell whole systems. SOA establishing changes not only the IT infrastructure, it changes
every part of the company.

4.7) [1:58:10] What is the next big thing?
Cloud computing is developing in parallel. [...] There is a contradiction between cloud computing and SOA, caused by the existence of latency time which SOA cannot use to work well. [...]
## Appendix II

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>System integration for organizations operating in always changing business environments is one key element of achieving business agility and an interoperable and adaptable IT infrastructure. It thus enables enterprise applications to correspond to dynamic market changes in a flexible way. (Minguez et al., 2009)</td>
<td>Better system integration (Business Agility)</td>
<td>Reaction towards change</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>In general, today’s organizations have difficulties regarding system integration because of the increasing heterogeneity of ISs, resulting in the existence of information silos. Those silos derogate the information sharing across diverse applications, which is needed to establish an information flow within the organization, as well as among itself and the outside world, for instance with customers and suppliers. Integration among silos and other applications is often problematic, since they differ in technical specifications such as their use of operating systems, programming languages or database approaches. In addition legacy systems do often fall into the category of silo systems, often containing fundamental organizational information but are not designed towards information sharing. However, in order to incorporate and thus exploit the information flow of all business processes within or beyond organizations it is essential that information usage and exchange is enabled regardless of any technological specification or dependencies of underlying systems or applications. Since SOA deals with the transformation of business processes into reusable, loosely-coupled services, it enables easy adjustments or creation of business processes and thus application integration. (Minguez et al., 2009)</td>
<td>Improved collaboration (Investment Reasons)</td>
<td>Empower legacy systems</td>
</tr>
</tbody>
</table>
Another basic SOA infrastructure element is a registry, which is a centralized repository comprising a directory of all implemented services of the organization (transparency). Usually the directory follows a specific grouping scheme, for instance listing services according to business units or business functions. It supports the application developers in order to find a demanded service without much effort. Besides, registries follow a certain standard (e.g. UDDI), allowing benefiting from the listed information not only inside an enterprise, but also with respect to business partners, such as suppliers, by allowing them to better integrate their own applications. (Mac Vittie, 2005)

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Another basic SOA infrastructure element is a registry, which is a centralized repository comprising a directory of all implemented services of the organization (transparency). Usually the directory follows a specific grouping scheme, for instance listing services according to business units or business functions. It supports the application developers in order to find a demanded service without much effort. Besides, registries follow a certain standard (e.g. UDDI), allowing benefiting from the listed information not only inside an enterprise, but also with respect to business partners, such as suppliers, by allowing them to better integrate their own applications. (Mac Vittie, 2005)</td>
<td>Improved collaboration (Investment Reasons)</td>
<td>Improved application development</td>
</tr>
</tbody>
</table>

According to Minguez et al. (2009; p. 49), “the basic communication within SOAs follow the “find, bind, invoke” paradigm, where a service provider publishes its service description in a service registry in order for a service consumer to find it, accept the interface contract and then invoke the service through the known request/reply mechanism.” Often an Enterprise Service Bus (ESB) in-between is used, serving as a standard-based integration platform handling web services, data transformation and intelligent routing and thus establishing reliable connections among organizational wide stored applications. An ESB provides a solution for connecting any application to it and sharing its data with other service applications. (Minguez et al., 2009; Mac Vittie, 2005)

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>According to Minguez et al. (2009; p. 49), “the basic communication within SOAs follow the “find, bind, invoke” paradigm, where a service provider publishes its service description in a service registry in order for a service consumer to find it, accept the interface contract and then invoke the service through the known request/reply mechanism.” Often an Enterprise Service Bus (ESB) in-between is used, serving as a standard-based integration platform handling web services, data transformation and intelligent routing and thus establishing reliable connections among organizational wide stored applications. An ESB provides a solution for connecting any application to it and sharing its data with other service applications. (Minguez et al., 2009; Mac Vittie, 2005)</td>
<td>Improved collaboration (Investment Reasons)</td>
<td>Empower legacy systems</td>
</tr>
</tbody>
</table>
Another main characteristic of SOA infrastructure is the reusability of assets. From a technological point of view, building reusable components means creating a specific set of functions and supplying the standard interfaces that allow them to be used over and over again (van den Berg et al., 2007). Within a SOA infrastructure only one single instance of a business function exists, which then can be reused in different applications. Accordingly, any change concerning the unique business function will be present in the linked applications as well. Thus market condition changes, for instance resulting in different pricing or discounting rules, can be managed easily by applying altered business logics to multiple applications at the same time. Therefore less time is demanded to implement changes in general and a fast reaction to dynamic market conditions is granted. (Mac Vittie, 2005)

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Improved application development (Investment Reasons)</td>
<td>Improved collaboration</td>
</tr>
<tr>
<td>Likewise, upgrades or modification of the underlying infrastructure of services do not affect the applications interacting with those services. Because applications are only linked to the services and not to the infrastructure itself, they don’t need to be adapted to any change of the infrastructure. Compared to past infrastructures, where modifications of such kind did resulted in interruptions throughout various levels of an enterprise system, with SOA only the implemented services need to be altered instead. (Mac Vittie, 2005)</td>
<td>Loose coupling; Modularity; Open standards (IT flexibility)</td>
<td></td>
</tr>
</tbody>
</table>
Furthermore, the complexity of technical implementations is visually reduced by the abstract service principle of SOA, representing business functionalities with clear business values (Herr et al., 2004; Homann et al., 2004). Thus the underlying service specifications like platform dependencies and programming languages are not necessarily need to be understood in order to exploit its benefits. In particular the importance of Web services within SOA is able to decrease the details of interface implementations, since it is an accepted standard of distributed computing. Thus interface integrations among collaborating parties can be performed more easily and flexible, resulting in lower integration costs. (Homann et al., 2004)

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved application development</td>
<td>Internal</td>
<td>External</td>
</tr>
<tr>
<td>Improved collaboration</td>
<td></td>
<td>Collaboration (Business Agility)</td>
</tr>
</tbody>
</table>

The implications of the reuse and standardization abilities of SOA technology can result in business benefits. The composition of once-implemented individual services can be used to generate additional services, providing new functionalities for service users without much effort and possibly resulting in new revenue streams due to changing business environments or upcoming opportunities (Brodkin, 2007; Chen, 2008; Woolf, 2008).

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved application development</td>
<td>Internal</td>
<td>Application development (Business Agility)</td>
</tr>
<tr>
<td>Improved collaboration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open standards, Modularity (IT-Flexibility)</td>
<td>External</td>
<td></td>
</tr>
</tbody>
</table>

Moreover standardization is also important from a business innovation point of view. For instance, standardized interfaces and messaging protocols are supporting innovation processes regarding applications, business processes and software development. (Haines & Haseman, 2009) According to Haines & Haseman (2009, p.2), innovation theory has already been used to explain the link between

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open standards (IT-Flexibility)</td>
<td>Internal</td>
<td>Application development (Business Agility)</td>
</tr>
<tr>
<td>Improved collaboration</td>
<td></td>
<td>Improved collaboration (Investment)</td>
</tr>
</tbody>
</table>
organizations and web services usage. Four different levels of innovation have thus been identified: technical solution level, IT solution level, internal business solution level and external business solution level. The authors then related those levels to the context of SOA, basing their decision on the fact that web services are a key component of SOA (see Table 1). Furthermore a multiple case study was conducted, leading to the conclusion that all investigated organization achieved at least the second level, and the majority even reached the third level of innovation through SOA adoption. Thus SOA adoption does not only allow diverse IT benefits, it also serves as a business innovation enabler.

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>The ability of reusing already coded services in combination with standardization, also leads to a reduction of general development costs and operating costs, for example through a reduction in application redundancy reduced development cycles (Legner &amp; Heutschi, 2007; van den Berg et al., 2007; Chen, 2008; Woolf, 2008). Because once a SOA infrastructure is in place, developers are able to reduce project durations and the time-to-market aspect for new applications by taking advantage of the loose coupling and reuse abilities of infrastructure components (Legner &amp; Heutschi, 2007; Woolf, 2008). Those time savings accordingly lower product and service development costs (Feig, 2008). Moreover, by establishing a Service-Oriented-Architecture an organization is able to react and adapt quicker concerning business changes and opportunities (Brodkin, 2007; van den Berg et al., 2007; Woolf, 2008).</td>
<td>Modularity; Open standards; Loose coupling (IT-Flexibility)</td>
<td>Application development; Collaboration (Business Agility)</td>
</tr>
</tbody>
</table>
Achieving alignment between IS capabilities and business objectives has been one of the primary concerns for CIOs for more than twenty years now (Chen, 2008). Accordingly business benefits or competitive advantage through SOA usage is based on how well the lower level of IT infrastructure service can provide the intended business level customer service and thus increases an organization’s performance (Brodkin, 2007; van den Berg et al., 2007; Chen, 2008; Woolf, 2008). Even though, definitions of a service varies from an IT- respectively business standpoint, which has caused many misunderstandings regarding past IT projects, service orientation offers new opportunities for companies to accomplish better outcomes of their designated IT- and business alignment. SOA facilitates easier communication between IT and business representatives, since a general terminology is shared by both parties and potential application demands of service owners or users can be linked to underlying technical processes, in terms of services instead of IT projects, without much complication (Legner & Heutschi, 2007; Chen, 2008; Haines & Haseman, 2009). Users know what services they desire and developers can offer a quick solution (Chen, 2008). However, applying SOA to an organization cannot be done overnight. Thus, managers should focus on the technology’s long term rather than its short term benefits (Brodkin, 2007).

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Achieving alignment between IS capabilities and business objectives has been one of the primary concerns for CIOs for more than twenty years now (Chen, 2008). Accordingly business benefits or competitive advantage through SOA usage is based on how well the lower level of IT infrastructure service can provide the intended business level customer service and thus increases an organization’s performance (Brodkin, 2007; van den Berg et al., 2007; Chen, 2008; Woolf, 2008). Even though, definitions of a service varies from an IT- respectively business standpoint, which has caused many misunderstandings regarding past IT projects, service orientation offers new opportunities for companies to accomplish better outcomes of their designated IT- and business alignment. SOA facilitates easier communication between IT and business representatives, since a general terminology is shared by both parties and potential application demands of service owners or users can be linked to underlying technical processes, in terms of services instead of IT projects, without much complication (Legner &amp; Heutschi, 2007; Chen, 2008; Haines &amp; Haseman, 2009). Users know what services they desire and developers can offer a quick solution (Chen, 2008). However, applying SOA to an organization cannot be done overnight. Thus, managers should focus on the technology’s long term rather than its short term benefits (Brodkin, 2007).</td>
<td>Application development (Business Agility)</td>
<td>React towards change</td>
</tr>
</tbody>
</table>

Service-oriented-architecture enables improved collaboration and thus productivity. Collaboration is depending on the access availability of information. The traditional silo-based form of organizational applications and information avoids the desired collaboration among employees, but also with partners.

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service-oriented-architecture enables improved collaboration and thus productivity. Collaboration is depending on the access availability of information. The traditional silo-based form of organizational applications and information avoids the desired collaboration among employees, but also with partners</td>
<td>Empowered legacy systems (Investment Reasons)</td>
<td>Improved collaboration</td>
</tr>
</tbody>
</table>
and customers. Productivity in a SOA environment can be enhanced for instance through better decision making due to information availability. Moreover increased collaboration with partners and suppliers allows faster adaption to altering market conditions and thus enables organizations to act more agile. (van den Berg et al., 2007)

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOA also reduces the IT/IS complexity of an organization by breaking down large applications into smaller pieces. The IT landscape of organizations usually follows a large, complex, monolithic approach, featuring a wide range of functionalities and strict rules for business process executions (van den Berg et al., 2007). Any change within those systems leads to very time-consuming analysis-, planning- and testing phases. Adjusting single services of a SOA is less inefficient regarding the required time and effort, since the IT/IS landscape as a whole is not affected by it. Also having the reuse ability in mind, SOA leads to a decrease of IT/IS complexity. Redundant capabilities can be eliminated and less code is needed to execute the same application portfolio as in non-SOA-based environments, resulting in cost reduction (van den Berg et al., 2007; Brodkin, 2007). In addition reusable services support the reduction of operating silos (van den Berg et al., 2007; Feig, 2008).</td>
<td>Modularity (IT flexibility)</td>
<td>Application development; React towards change (Business Agility) IT- cost reduction; Improved collaboration (Investment Reasons)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>More benefits of standardized interfaces among applications are cost savings due to more efficient system integration, as well as the ability of decreasing operating costs as a result of less complex maintenance demands for the IT department (Legner &amp; Heutschi, 2007).</td>
<td>Open standards (IT-Flexibility)</td>
<td>IT-cost reduction; Improved collaboration (Investment Reasons)</td>
</tr>
</tbody>
</table>
Reusability is furthermore connected to the possibility of more effective monitoring and management of IS, for instance due to less software support and maintenance demands (van den Berg et al., 2007; Woolf, 2008) (Woolf, 2008). Moreover IT management is eased through higher transparency regarding application dependencies and application interfaces (Legner & Heutschi, 2007). Therefore SOA is able to decrease the IT/IS complexity of organizations which leads to savings of certain cost- and time factors.

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Modularity (IT-Flexibility)</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>External</td>
</tr>
</tbody>
</table>

Furthermore SOA enables new scopes of business agility by upgrading, replacing or substituting existent services or even switching service providers without causing negative impacts on business operations (Chen, 2008).

<table>
<thead>
<tr>
<th>Text Passage or Interview Respond</th>
<th>Sub-Category</th>
<th>Relationship to Sub-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Modularity; Loose coupling; Standardization (IT-Flexibility)</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>External</td>
</tr>
</tbody>
</table>
References


Charmaz, K., 1990. 'Discovering' chronic illness: Using grounded theory. Social Science and Medicine, 30(11), pp.1161-72.


