Service-oriented Architecture: Describing Benefits from an Organisational and Enterprise Architecture Perspective

Hatitye Chindove, Lisa F. Seymour and Francois I. van der Merwe
Department of Information Systems, University of Cape Town, Rondebosch, Cape Town, South Africa

Keywords: Service-oriented Architecture, SOA Benefits, Enterprise Architecture, Service Orientation.

Abstract: Software architecture models describe the technical structure, constraints, and characteristics of software components and the interfaces between them. Service-Oriented Architecture (SOA) is a recent software architecture style with many benefits if used in the right context. Business agility, customer satisfaction, faster time to market, ease of partnering and lower business costs are some promised benefits. Yet SOA has not always benefitted organisations. One reason given is a misunderstanding of the relationship between SOA and enterprise architecture (EA). Therefore, this study in a large retail organisation in South Africa describes SOA benefits and classifies them into the various EA domains. SOA benefits are also classified into six broad categories namely: strategic, organisational, operational, managerial, maintenance and governance. The study comprises three cases from one organisation that deployed different architectures. SOA benefits are contrasted with benefits from other approaches. Organisational benefits not described before include greater collaboration amongst SOA participants enabling better learning opportunities. The results should assist IT management in preparing SOA business cases and in managing SOA deployments to ensure benefits are achieved.

1 INTRODUCTION

Organisations need to constantly change to keep up with market competition (MacLennan and van Belle, 2014). Yet change is greatly dependent on information technology (IT) and business infrastructure (Krafzig, Banke and Slama, 2005) which needs to be flexible enough to accommodate strategic changes and reflected then timeously and efficiently within the organisation. Enterprise IT is closely coupled with internal organisation, process and business models of the organisation. Therefore enterprise IT establishes system integration, agility and change which requires a focus on system design (Hoogervorst, 2004). System design in turn requires a set of statements and models that describe the solution component and the assigned functionality of those components (Krafzig et al., 2005; Valipour et al., 2009) which can be referred to as software architecture. Software architecture artefacts or models describes system components and includes the technical structure, constraints, and characteristics of the components and the interfaces between them (Reyes-Delgado et al., 2016; Valipour et al., 2009).

Service-Oriented Architecture (SOA) promises business agility and is defined as a software architecture style and a technical and organisational framework enabling the development of platform-independent business functionality through using services (Malekzadeh, 2010; Singh and Tyagi, 2015). A 'service' is described as a self-contained web based application capable of completing tasks on its own and able to discover and engage other services to complete higher level transactions (Stojanović and Dahanayake, 2005). Services are viewed as blocks and therefore SOA has component orientation that incorporates loose coupling and process control into its design (Hau et al., 2008).

Whilst SOA may seem to be the ideal architecture to tackle most business requirements other architectures such as Event-Driven Architecture or a combination of SOA and other architectures can be a better choice (Malekzadeh, 2010). Yet limited case studies of SOA in different industries exist (MacLennan and van Belle, 2014; Joachim, 2011). There are concerns that the risks and costs around the implementation and use of SOA design principles could make SOA projects run longer and become more expensive without yielding immediate benefits (Hau et al., 2008). Companies also struggle to

Chindove, H., Seymour, L. and Merwe, F.
Service-oriented Architecture: Describing Benefits from an Organisational and Enterprise Architecture Perspective.
DOI: 10.5220/0006383604830492
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 3*, pages 483-492
ISBN: 978-989-758-249-3

measure SOA return on investment (Malekzadeh, 2010). The SOA approach is described as less successful than anticipated with many challenges and SOA success stories are noted to be rare (Alghamdi, Potter and Drew, 2016). Therefore, there is a call to study these as case studies (Joachim, 2011; vom Brocke et al., 2010). Hence, the research question posed here is "What benefits are organisations achieving from adopting SOA?" Researchers have also noted the lack of a model for classifying SOA benefits (Viering, Legner and Ahlemann, 2009) and therefore we describe benefits using two frameworks, the first being Enterprise Architecture (EA).

2 ENTERPRISE ARCHITECTURE AND SOA

Services used in SOA are often described with characteristics and attributes that include; interface orientation, autonomy, loose coupling, standardization, reusability, business orientation and compassable (MacLennan and van Belle, 2014; Malekzadeh, 2010). Although SOA is grounded in IT, it does not mean it should be IT driven but it needs to also be driven by business (Alghamdi et al., 2016; MacLennan and van Belle, 2014). A view of how SOA is understood from an organisational perspective can be critical to its implementation. EA is an architecture that assists in understanding the business and IT perspective (Kistasamy et al., 2010).

In as much as EA and SOA are different, they share a number of objectives and SOA without EA is compromised (Alghamdi et al., 2016). EA can be viewed as a structured and aligned set of plans which provide integrated modelling of the enterprise's business and technology landscape, in past, current and future states (Simon, Fischbach and Schoder, 2013). The models form the central deliverable of the EA practice. When EA evolution is not managed and aligned at the modelling level, misrepresentation and occasionally even failures result (Alwadain et al., 2016).

EA can be considered as a collection of the constituent Business, Information, Application and Technology architectures with inter-relationships among them, with their joint properties being essential to the entire architecture (Rohloff, 2011). Business systems should be agile to maintain business-IT alignment and EA has become an important enabler (Harishankar and Daley, 2011; Sasa and Krisper, 2011).

Business architecture artefacts can come from organisation initiatives such as strategic business planning and business process redesign (Harishankar and Daley, 2011). Therefore, business architecture should determine the technology architecture (Rohloff, 2011). Both SOA and EA share the objective of achieving business and IT alignment (Kistasamy et al., 2010). SOA can impact EA frameworks, methodologies, governance and tools (Alwadain, Fielt, Korthaus and Rosemann, 2016). To align EA models with the corresponding real world, enterprise architects need to be aware of changes in the enterprise (Alwadain et al., 2016).

However due to misunderstanding the relationship between SOA and EA, organisations have failed to benefit from their combined use (Alwadain et al., 2016; Kistasamy et al., 2010). There are no empirical studies describing how EA evolves due to SOA. Therefore, researchers have called for studying their integration (Alwadain et al., 2016).

EA and SOA both co-exist equally to ensure that technology solutions support business processes and SOA touches all EA domains. SOA brings more agility to EA practice, increases EA acceptance and usability (Zhao, 2013). EA provides SOA practice with enterprise views (Zhao, 2013). SOA and business architecture both enable business agility. SOA uses business architecture to develop artefacts such as the component business model as input to business services (Harishankar and Daley, 2011). SOA is also a methodology optimised to the application architecture domain (Kistasamy et al., 2010). Table 1 shows the mapping of EA domains with the relevant SOA solution stack.

Table 1: Mapping EA and SOA domains (Kistasamy et al., 2010).

SOA solution stack	EA Domain
Business Process	Business architecture
Services and components	Application architecture
Integration Architecture	Technology architecture
Data Architecture	Information architecture
Quality of Service, monitoring and infrastructure	Technology architecture

3 EVALUATION OF SOA

Evaluation of SOA should consider the costs, risks and benefits of its adoption. Power relations between firms in a specific industry play an influential role in SOA adoption and lead to variations of adoption in different industries (Ciganek, Haines and Haseman, 2009). Therefore, SOA needs to be evaluated and studied in different contexts. A driving factor for

adopting a methodology is cost. SOA adoption has significant hardware and network infrastructure costs, deployment costs (including application migration, integration, and database conversions), support and maintenance costs, especially in the early adopting stages, and performance costs (Rabhi et al., 2007). The most important project risks are reliability, security and performance (MacLennan and van Belle, 2014). Other risks include implementation challenges, SOA expertise, development governance, service design, technical systems health and project time constraints (Koch, 2007; Komoda, 2006; MacLennan and van Belle, 2014; Rabhi et al., 2007). Depending upon the project, SOA can introduce conflict such as when certain SOA design principles make the project longer and more expensive without yielding immediate benefits (Hau et al., 2008). The benefits of SOA also vary depending on the maturity levels of SOA adoption. SOA adoption can move from initial services, architected services, business and collaborative services, measured business services and finally optimized business services (Soni, 2005). Given that many organisations are failing to achieve return from SOA investments (Malekzadeh, 2010; vom Brocke et al., 2010), it is important to more clearly describe benefits of SOA. This paper aims to do this. SOA benefits identified in the literature are described in more detail in the findings section of this paper.

4 RESEARCH METHOD

The purpose of this study is to explore and describe the benefits derived by the use of the SOA architectural style versus those derived through other software architectural styles. This research is qualitative and interpretive in nature using a multiple case study (Yin, 2012) of three departments responsible for application development and application support (C1-C3) in a single large South African retail organisation where SOA and non-SOA architectural styles are used. Interpretive studies include second-order constructs of the researcher's interpretation of interviewees' first-order constructs (Walsham, 2006). The case study approach is useful for descriptive and explanatory approaches going well beyond exploratory research (Yin, 2012). Semi structured interviews were performed during 2015, supported by observation notes. Additional data sources include the extract of support calls and change requests from the service management system. The thirteen respondents were selected using judgement sampling across various roles in the software development process including:

- 2 project managers (C1 and C2);
- 3 enterprise architects (C1, C2 and C3);
- 3 solution developers / support personnel (C1, C2 and C3);
- 1 solution testers (C1); and
- 2 business analysts (C1 and C2).

A combined deductive and inductive thematic analysis was performed (Fereday and Muir-Cochrane, 2006) by validating themes identified in the literature, uncovering new themes and describing the meanings the social actors attached to benefits of the architecture style used. Each text excerpt or empirical observation for each theme was counted and then totalled. Data validation was achieved through triangulation between multiple independent data sources.

C1 with 5 respondents had adopted SOA as its software architecture style across the entire department. C1 supported multiple systems to deliver merchandise to customers and subscribers of their systems and was responsible for managing customer information. The C1 application portfolio comprised 15 in-house applications and interactions with other third party applications. C1 had 10 employees, had existed for 2 years and had 9 other third party partners.

C2 with 6 respondents was not making use of SOA but followed an approach focusing on application design principles and using point to point integration and enterprise integration. C2 supported systems for inventory management, internal merchandise logistics and merchandise planning for the retailers of the organisation. Their application portfolio comprised 17 systems of mostly applications developed and licensed by third party vendors but also some custom built applications. C2 had 50 employees, had existed for over 10 years and had 20 other third party partners.

C3 with 2 respondents made partial use of the SOA architectural style in some aspects of its architecture but also applied other application design principles to achieve enterprise integration with different departments. C3 supported internal enterprise integration and integration with external partners. Their application portfolio of 60 systems comprised mostly applications developed in-house and other third party applications. C3 had 8 employees, had existed for 10 years and had 30 other third party partners.

Table 2: Summary of Benefits with counts of text excerpts per case.

Categorised Benefit	References	C1	C2	C3
Strategic Benefits		**26**	**6**	**6**
Improved Agility	(Joachim, 2011; Komoda, 2006; Malekzadeh, 2010)	6		1
Opportunity for agile development	(Malekzadeh, 2010)	6		
Improved Change Effect Size		3		
Domain Driven Development	(Bukhsh, Sinderen and Singh, 2015)			1
Leveraging Legacy IT Assets	(Lewis et al., 2005; Rabhi et al., 2007)	3		1
Reuse opportunity	(Bukhsh et al., 2015; Rabhi et al., 2007)	8	6	3
Operational and Maintenance Benefits		**28**	**5**	**14**
Reduced Maintenance Effort	(Carey, 2008)	7		2
Improved Operations Support	(Hau et al., 2008; Joachim, 2011)	4		
Reduced Development Time	(Rabhi et al., 2007; Yoon and Carter, 2007)	4		1
Improved Flexibility	(Bukhsh et al., 2015; MacLennan and van Belle, 2014)	5	1	4
Reduced Testing effort		1		
Increased Issue Isolation		1		
Improved Refactoring Opportunity		1	2	1
Improved Standardization	(MacLennan and van Belle, 2014)	3		2
Improved Interoperability	(Lewis et al., 2005)	2	1	1
Improved Self Documentation				1
Reduced Maintenance Costs	(Hau et al., 2008; Joachim, 2011)			1
Ability to handle large data volumes	(Yoon and Carter, 2007)			1
Incremental Development Opportunity			1	
Organisational Benefits		**17**	**1**	**4**
Improved Collaboration		3		
Improved Learning Opportunity		6		2
Increased Opportunity to Innovate		4		2
Improved System Understanding		4		
Improved Team Dependencies			1	
Managerial Benefits		**4**		**4**
Ease of Integration	(Joachim, 2011; Yoon and Carter, 2007)	1		1
Improved Data Visibility	(The Open Group, 2009)	1		
Opportunity to measure cost savings		1		
Separation of Concerns		1		3
Governance Benefits		**6**		
Improved Application Control	(Hau et al., 2008)	1		
Improved Change Control		4		
Improved Compliance Management		1		
Key: Unique to SOA	Not Unique to SOA	Not in SOA		

5 FINDINGS AND DISCUSSION

In this section, we will for brevity purposes refer to the SOA architectural style simply as SOA. In this case study respondents described some of the benefits as being either unique to SOA, achieved through SOA but not uniquely and not achieved through SOA. The findings are presented in Table 2 with the count of empirical observations or text excerpts from all interviews. Note that in some cases a respondent could have mentioned one benefit more than once. A benefit framework for SOA covering strategic, operational and technical dimensions has been called for (Viering et al., 2009) and therefore these categories were used and extended. Due to space limitations, only the benefits seen as unique to SOA, are described and contrasted with the relevant literature.

5.1 Strategic Benefits

SOA has the capacity to assist with developing a sustainable, IT-based plan to achieve strategic goals. Six themes emerged, namely: improved agility, improved opportunity for agile software development, improved change effect size, domain driven development opportunity, reuse opportunity, and leveraging legacy IT assets.

Agility refers to the ability of the organisation to sense and respond to market opportunities and changes in its local environment. Krafzig et al. (2005) state that the main motivation around SOA is to deliver agility and enable rapid development and modification of software through service composition. Within the SOA case, new system development time was reduced by means of reuse as demonstrated in the following quote: *"so the time and effort to fulfil all those capabilities now when you plug in a new sales system is far easier and quicker. So that gives the business the agility"* (R4-C1).

With agile development, software evolves through collaboration between self-organizing, cross-functional teams with adaptive planning, evolutionary development and rapid and flexible responses to change. A software architecture design artefact is expected to be expanded during the design phase of a software engineering process which is represented by a software development methodology (SDM) and can thus be considered as a well-structured process for elaborating a software system (Reyes-Delgado et al., 2016). The separation of concerns is used as a principle in both agile software development and SOA which can be considered in terms of component orientation thereby supporting loose coupling and ultimately flexibility and agility of the solution. Separation of concerns positively influenced the development process and delivery time allowing for agile development practices. Agile software development was a major theme that was only evident in C1. *"In terms of delivery… no matter where you have to work, you kind of know where to find everything... You will know exactly what piece of code to look at when maintaining"* (R3-C1).

Improved change effect size measures the benefit of a single change on one component across applications. This emerged as a minor theme in C1 where SOA was adopted and changes to a service resulted in a cascading effect on all other components making use of the updated component. "*so it might not be quicker to implement the change but when you implement the change everyone benefits immediately from that change so we have seen that going quite well"* (R1-C1).

Domain driven development is a software development approach where the implementation is tied to an evolving model with the main focus being on a core domain and domain logic. SOA provides an opportunity to categorize service provisions at a domain level within an organisation and provides the organisation more opportunities to reuse their assets and develop new functionality where ownership and standards reside with the owners of the services for a domain. The ownership derived from the domains will also improve collaboration in the cases where reuse is required. This was evident in C1 and C3. *"Also the nice thing we are seeing coming from SOA is you know the term called domain driven development. Each company or division owns that domain of the business. …so with our approach and moving towards SOA, basically it comes down to the people who own that domain they will provide services around those domains"* (R8-C3).

5.2 Operational and Maintenance Benefits

Operational benefits in this study refer to operational activities performed by the relevant case departments enabled by SOA as well as the actions taken to prevent, diagnose, update, replace and repair IT infrastructure. We describe five benefits unique to SOA, namely, reduced maintenance effort; improved operations support; reduced development time; flexibility; and reduced testing effort.

Maintenance effort was reduced through the reuse of existing capabilities to develop applications in response to support calls. This confirms the literature (Carey, 2008). Figure 1 shows that over a period of a year when support calls for the SOA infrastructure increased there was not a significant increase in changes implemented to resolve the calls.

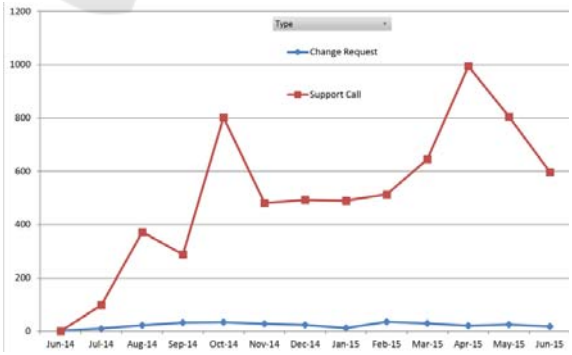

Figure 1: Support calls and resultant change requests.

Improved operations support only appeared as a theme in C1. SOA made operation work more manageable and simpler in the cases where investigations have to be conducted due to the use of

asset wrapping for vendor applications. Figure 2 shows that in C1 the ability to resolve support calls within agreed service delivery agreement parameters improved with SOA when introduced in 2015 and even when there was a spike in calls. *"whereas with this one now you can kind of say, if something is going wrong with somebody's order when it is not being delivered they you know 'okay it is in this area' so you can go and look exactly in that area"* (R11-C1). Literature has noted that SOA improves the automation and management of business processes (Joachim, 2011).

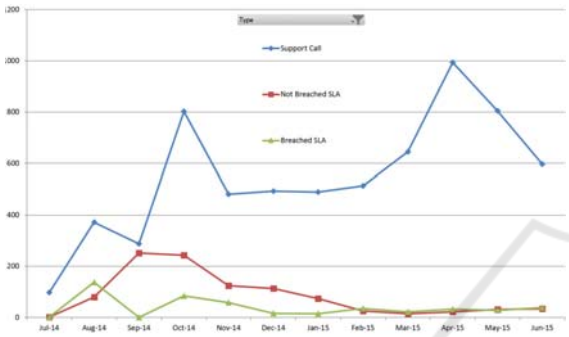

Figure 2: C1 support calls and service level agreements.

Reduced development time (Yoon and Carter, 2007) is enabled by reduced component dependencies. SOA is component based (The Open Group, 2009) and also due to the composability of services component dependencies are minimal. The reduced component dependency influences flexibility and testing effort whilst it is dependent on composability of the service. Four respondents mentioned this as a SOA benefit: *"with the services we were able to agree on what data we were going to transmit between each other through a service layer first and each one of us could write dummy endpoints that we could then code against and test against. When we were both ready we simply took off the dummy points and pointed at each other and it would work. That saved us a lot of time as that meant we could develop our work in parallel... we could keep it quite segregated"* (R1-C1). Development effort for new services replacing legacy applications also reduced with SOA. The creation of asset wrappers for legacy applications enabled tests to compare legacy applications with new services.

Flexibility in system designs allows high design adaption with external changes and is a valued SOA benefit (MacLennan and van Belle, 2014). Flexibility appeared across all cases but was more evident in C1 and C3. Composability and interoperability improve flexibility. SOA increased flexibility allowing a number of application frontends to be developed based on one backend. The reuse of the backend services and SOA design principles improved flexibility. Robust architecture design principles of layered architectures and reduced component dependencies also drive flexibility in the application landscape. Loose coupling of services with SOA requires good design principles to achieve flexibility (MacLennan and van Belle, 2014). This is evident from a quote: *"so if you haven't followed these design principles, basically you are going to rewrite the application. I mean you are duplicating all the logic, all the business rules and duplicating effort. Whereas if you had the layered architecture with the OO principles and the SOA principles, you will be in a position where all your business rules and all your business logic is encapsulated into your service layer and then you really get a light-weight front end. So your front-end really does nothing except interact with the user"* (R4-C1).

Reduced testing effort emerged as a minor theme appearing once in C1. Through service composition, testing in single functional areas was much simpler. There was a better view of dependencies in the landscape enabling directed testing and less testing effort. *"I think it has made it a lot much quicker because things are so much compartmentalised, …so you really won't need to do the whole regression testing and all that kind of stuff because it hasn't really touched those areas"* (R11-C1).

5.3 Organisational Benefits

Organisational SOA benefits refer to benefits in the IT employee's view of the organisation and its environment. These included improved collaboration, improved learning opportunity, increased opportunity to innovate, and improved system understanding.

Collaboration is the action of working together on a joint intellectual effort. This theme appeared in two C1 interviews and showed that SOA enabled collaboration in solving problems. Different teams would work together to deliver solutions and assist each other in problem solving. It is argued that with SOA using layered service components, people with different skills can work in different layers providing flexibility in skills management (Zhao, 2013).

Participants using SOA were seeing more learning opportunities coming from using this architectural style as it enabled quick initial contributions without having a full understanding of the business processes: *"no matter where you have to work, you kind of know where to find everything… which brings down the learning curve"* (R3-C1).

Innovation is the process of translating an idea into a product or service that creates value to its consumers. This was a minor theme with respondents

mentioning that there were more opportunities to innovate using SOA. Given the opportunity to reuse existing services, there are opportunities to innovate how a service is delivered to consumers such as through mobile or web based solutions.

Improved system understanding refers to the ability to comprehend the linked tasks of delivering a service. C1 respondents described how the architectural view allowed a better understanding of the application portfolio from a development perspective. At a glance one could better understand how the application portfolio is arranged and also how the low level parts of the landscape are organized. *"it is a lot simpler because you … see at a glance what is happening. It is not like everything is this huge complicated and everything is thrown together like the old legacy systems"* (R11-C1).

5.4 Managerial Benefits

Managerial benefits incorporate those related to the control and monitoring of organisational resources. Three themes unique to SOA in the cases studied are: ease of integration; improved data visibility, and opportunity to measure cost savings.

Integration refers to joining together subsystems into a single system. Ease of integration appeared as a minor theme but it has a significant influence on the other benefits of SOA. Ease of integration enabled development without bottlenecks arising at integration stages as standards of interoperability had been agreed to. Due to ease of integration, a range of potential solutions existed for new requirements. A literature review identified multiple papers supporting integration as a SOA benefit (Joachim, 2011).

Data visibility through messaging and the ability to measure cost savings through metrics appeared as minor themes in the data, and were only evident in C1. Measurement metrics were applied to costs in the SOA application landscape as demonstrated by the following quote: "*if you take each capability … determine effort in providing a solution … and look at how many times you reuse that capability. You can determine the cost savings"* (R4-C1).

5.5 Governance Benefits

Governance benefits relate to the processes that help to ensure efficient and effective use of IT in enabling organisational goals. Three themes emerged in this investigation namely: Improved application control, improved change control and improved compliance management in software development.

Literature refers to improved application control as a new SOA world that can be easily controlled and adapted (Hau et al. 2008). The presence of domain boundaries improves application control improving the governance of processes and compliance. *"Where services help is for example if your services is returning a clear text ID number… you can mask it at your service level so that every system that is subscribing gets a masked ID number… makes every (system) POPI compliant in one go"* (R1-C1). In this case POPI refers to the Protection of Personal Information Act in South Africa.

Improved change control refers to managing software changes, ensuring that unnecessary changes aren't made, that services are not unnecessarily disrupted and that resources are used efficiently. Responsive change management can be considered a technical architecture principle. This theme was only evident in C1 and was derived from SOA through flexibility, composability and separation of concerns governed by agile software development principles. "*In terms of the web service and the windows services using the same block of code, refactoring is quite simple. Like most of the code sits in one place and you know where to go find it and is relatively simple*" (R3-C1). "*This model gives you the capabilities to adapt and control new changes"* (R5-C3).

Compliance management in software development refers to complying with rules, which may be organisational standards or legislation. This theme appeared as a minor theme in case C1 where SOA had improved compliance management. Through the increased change effect size, changing selective components of services to be compliant resulted in all systems making use of the relevant service becoming compliant. With SOA compliance management can be at a domain level of services. This eases domain application and ownership of compliance.

5.6 Summary of SOA Evaluation

The relationship between SOA and EA has been misunderstood resulting in organisations failing to benefit from their combined use (Alwadain et al., 2016; Kistasamy et al., 2010). Researchers have therefore called for studying their integration (Alwadain et al., 2016).

To assist in the understanding, the benefits identified in this study are summarised from an EA perspective in Figure 3 and show how SOA is contributing towards all four EA domains. The explanation of the categorisation and the definition of

Categorised Benefits	Business Architecture	Info / Data Architecture	Technology Architecture	Application Architecture
Strategic Benefits				
Improved Agility	x			
Opportunity for agile development				x
Improved Change Effect Size		x		
Domain Driven Development	x			
Operational and Maintenance Benefits				
Reduced Maintenance Effort				x
Improved Operations Support				x
Reduced Development Time				x
Improved Flexibility			x	
Reduced Testing effort				x
Organisational Benefits				
Improved Collaboration				
Improved Learning Opportunity				
Increased Opportunity to Innovate				x
Improved System Understanding				x
Managerial Benefits				
Ease of Integration			x	
Improved Data Visibility		x	x	
Opportunity to measure cost savings		x		
Governance Benefits				
Improved Application Control				x
Improved Change Control			x	
Improved Compliance Management		x		

Figure 3: Benefits of SOA classified from an EA Perspective.

the domains from TOGAF (The Open Group, 2011) follows.

Application Architecture describes the structure of software applications, their inter-relations and relationships with key business processes. As expected, the majority of SOA benefits are within the application architecture domain.

Business Architecture describes the structure and relationships between business strategy, governance, functions, and key business processes. The benefits improved agility and domain driven development fall under this domain. Improved agility is a business strategy and supports key business processes so falls in the Business Architecture domain. The concept of a business domain can be viewed as how an organisation draws its competitive boundaries or functions.

Data Architecture describes the structure and relationships between the organization's data sources, logical and physical data assets and data management resources. Four benefits are considered data architecture benefits Improved change effect size has an impact on data and cascading changes to all existing applications as the data is tightly coupled with the service that manages it. The measurement of cost saving is possible through the data architecture which can expose and enumerate data that was not previously evident. Compliance is not possible unless the data / information architecture allows for its specific measurement, evaluation and management. Data visibility can be considered in terms of data/information architecture in that you need to define the messages, their attributes etc. which benefits the information architecture.

Technology Architecture describes the the technology building blocks from which that IT system will be constructed using the logical software and hardware capabilities, such as networks and standards that support the other architectures. Four benefits are included under technology architecture because of their impact on the underlying infrastructure. Improved flexibility and ease of integration is achieved through services which reduce component dependencies. When considered from an ITIL perspective, change control is the governance of the technical process of implementing and managing the software change. Finally data visibility is also considered from its technical implementation through messaging and hence is also within the technical architecture domain.

Some of the benefits also had associated cost and risk implications that were not be described here and are required for a complete evaluation. The practitioners noted that improved reuse opportunities resulted in implications on backward compatibility, maintenance effort, testing challenges and increased testing costs. The more a single service is reused across a number of different applications, the greater the effort required to modify the service. This in turn influenced service composition and reuse opportunity where greater focus on the design was required to meet the disparate change requests.

5.7 Limitations

This study has limitations in that some respondents failed to fully comprehend several benefits and in C2 there were very few respondents. The study also limited its scope to benefits even though the respondents referred to the risks and costs of SOA. The categorisations used for grouping the benefits were derived from interpretive thematic analysis and to an extent are exploratory. Further studies could validate the classification.

6 CONCLUSION

Researchers have called for more studies of successful SOA cases and a better understanding of the SOA business case. In response, this study describes SOA benefits within a large successful retail organisation in South Africa. The benefits of the SOA architectural style are classified into five broad categories in an organisation and in terms of the EA domains. These benefits will be useful for practitioners when making SOA business cases and when trying to ensure that benefits are obtained. The study also explores the connections between EA and SOA from a practioner perspective, an area that requires further research. Benefits impacted all EA domains which to some extent supports the perception of SOA as an EA style (Zhao, 2013).

The respondents described how the SOA approach enabled them to improve themselves and the organisation as a whole. Organisational benefits arising from the use of SOA have not been described before and include greater collaboration across teams and improved learning opportunities. This resulted in better engagement of the staff and better involvement in their workspace.

During the study it became apparent that there is still a gap in knowledge in terms of the levels of stakeholder involvement in SOA projects that needs study. Future work can also focus on gaining more understanding of the causal costs and risks around adopting SOA. Future work could therefore apply a different research approach such as grounded theory to provide more information around the phenomena of SOA. This can assist in having a more defined business case for SOA in organisations. A future longitudinal study could also identify additional SOA benefits and understand which benefits are achieved as an organisation goes through different levels of SOA maturity.

REFERENCES

Alghamdi, B., Potter, L. E., Drew, S. 2016. Identifying Best Practices in Organisational SOA Governance Adoption: Case Study of Saudi Arabia's E-Government Programme. PACIS 2016 Proceedings, 365.

Alwadain, A., Fielt, E., Korthaus, A., Rosemann, M. 2016. Empirical insights into the development of a service-oriented enterprise architecture. *Data & Knowledge Engineering*, 105, pp. 39-52.

Bukhsh, Z. A., van Sinderen, M., Singh, P. M. 2015. SOA and EDA: A comparative study: Similarities, differences and conceptual guidelines on their usage. In e-Business and Telecommunications (ICETE), 2015 12th IEEE International Joint Conference on, (2), pp. 213-220.

Ciganek, A., Haines, M., Haseman, W. 2009. Service Oriented Architecture Adoption: Key Factors and Approaches, *Journal of Information Technology Management* (20:3), pp. 42-54.

Carey, M. J. 2008. SOA what?, *IEEE Computer* (3:41), pp. 92-94.

Fereday, J., Muir-Cochrane, E. 2006. Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development, *International Journal of Qualitative Method*s, pp. 80-92.

Harishankar, R., Daley, S.K. 2011. Actionable Business Architecture, in *IEEE 13th Conference on Commerce and Enterprise Computing (CEC)*, pp. 318-324.

Hau, T., Ebert, N., Hochstein, A., Brenner, W. 2008. Where to Start with SOA: Criteria for Selecting SOA Projects, in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, (314), pp. 1-9.

Hoogervorst, J. 2004. Enterprise architecture: Enabling integration, agility and change, *International Journal of Cooperative Information Systems*, (13:03) pp. 213-233.

Joachim N. 2011. A literature review of research on service-oriented architectures (SOA): characteristics, adoption determinants, governance mechanisms, and business impact. AMCIS 2011 Proceedings, 339.

Kistasamy, C., van Der Merwe, A., De La Harpe, A. 2010. The relationship between service oriented architecture and enterprise architecture, in *2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops*, pp. 129-137.

Koch, C. 2007. How to get the most from SOA, *CIO*, (20:22), pp. 53-60.

Komoda, N. 2006. Service Oriented Architecture (SOA) in Industrial Systems, IEEE International Conference on *Industrial Informatics*, 2006, pp. 1-5.

Krafzig, D., Banke, K., Slama, D. 2005. *Enterprise SOA: Service Oriented Architecture Best Practice.* New Jersey: Pearson Education Inc.

Lewis, G., Morris, E. J., Smith, D. B., Wrage, L. 2005. Service-Oriented Architectures as an Interoperability Mechanism, News at SEI, Carnegie Mellon University. Retrieved http://www.sei.cmu.edu/library/abstracts/news-at-sei/eyeonintegration20052.cfm

MacLennan, E., van Belle, JP. 2014. Factors Affecting the Organisational Adoption of Service-Oriented Architecture (SOA), *Information Systems and e-Business Management*, (12:1), pp. 71-100.

Malekzadeh, B. 2010. Event-Driven Architecture and SOA in collaboration, Masters Thesis. University of Gothenburg.

Rabhi, F.A., Yu, H., Dabous, F.T. Wu, S.Y. 2007. A service-oriented architecture for financial business processes, *Information Systems & e-Business Management,* (5:2), pp. 185-200.

Reyes-Delgado, P.Y., Mora, M., Duran-Limon, H.A., Rodríguez-Martínez, L.C., O'Connor, R.V. and Mendoza-Gonzalez, R. 2016. The strengths and weaknesses of software architecture design in the RUP, MSF, MBASE and RUP-SOA methodologies: A conceptual review, *Computer Standards & Interfaces*, (47), pp. 24-41.

Rohloff, M. 2011. Integrating Innovation into Enterprise Architecture, in *Wirtschaftinformatik Proceedings*, pp. 776-785. Association for Information Systems.

Sasa, A., Krisper, M. 2011. Enterprise architecture patterns for business process support analysis, *Journal of Systems and Software*, (84:9), pp. 1480-1506.

Simon, D., Fischbach, K., Schoder, D. 2013. An Exploration of Enterprise Architecture Research, *Communications of the Association for Information Systems* (32), pp. 1-72.

Singh, N., Tyagi, K. 2015. A Literature Review of the Reliability of Composite Web Service in Service-Orientated Architecture, *ACM SIGSOFT Software Engineering Notes,* (40:1), pp. 1-8.

Soni, B. 2005. A New Service-Oriented Architecture (SOA) Maturity Model, White Paper, Sonic Software Corporation, Amber Point Inc. Retrieved http://soa.omg.org/Uploaded%20Docs/SOA/SOA_Maturity.pdf.

Stojanović, Z., Dahanayake, A. (Eds.). 2005. *Service-oriented software system engineering: challenges and practices*. IDEA Group, Hershey.

The Open Group. 2009. Service Oriented Architecture. Retrieved https://www.opengroup.org/soa/source-book/soa/index.htm

The Open Group. 2011. TOGAF® Version 9.1 Chapter 2 Core Concepts. Retrieved http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap02.html

Vom Brocke, J., Becker, J., Braccini, A.M., Butleris, R., Hofreiter, B., Kapočius, K., De Marco, M., Schmidt, G., Seidel, S., Simons, A. Skopal, T., 2011. Current and future issues in BPM research: a European perspective from the ERCIS meeting 2010, *Communications of the Association for Information Systems* (28), pp. 393-414.

Valipour, H., Amirzafari, B., Maleki, K., Daneshpour, N. 2009. A brief survey of Software Architecture and Service Oriented Architecture, in *2nd IEEE International Conference on Computer Science and Information Technology*, pp. 34-38.

Viering G, Legner C, Ahlemann F. 2009. The (lacking) business perspective on SOA-critical themes in SOA research. *Wirtschaftsinformatik.*

Walsham, G. 2006. Doing interpretive research. European Journal of Information Systems, 15, pp. 320- 330.

Yin, R. K. 2012. A (very) brief refresher on the case study method. Applications of case study research, pp. 3-20. Sage.

Yoon T, Carter P. 2007. Investigating the antecedents and benefits of SOA implementation: a multi-case study approach. AMCIS 2007 Proceedings. 195.

Zhao Y. 2013. Enterprise Architecture and SOA. ArchiTech Consulting LLC. Retrieved http://www.architechllc.com/uploads/Publications/EA-SOA.pdf