

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and
Information Systems

School of Computing and Information Systems

8-2008

Critical success factors in SOA implementation

J. ERICKSON

Keng SIAU

Singapore Management University, klsiau@smu.edu.sg

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research



Part of the [Databases and Information Systems Commons](#)

Citation

ERICKSON, J. and SIAU, Keng. Critical success factors in SOA implementation. (2008). *Proceedings of the Fourteenth Americas Conference on Information Systems (AMCIS 2008), Toronto, Ontario, Canada, August 14-17..* 1-9.

Available at: https://ink.library.smu.edu.sg/sis_research/9416

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylids@smu.edu.sg.

Critical Success Factors in SOA Implementation

John Erickson

University of Nebraska at Omaha
johnerickson@mail.unomaha.edu

Keng Siau

University of Nebraska Lincoln
ksiau@unl.edu

ABSTRACT

Service Oriented Architecture (SOA) has become flavor du jour for many businesses. Seemingly, almost every company has implemented, is in the midst of implementing or is seriously considering a SOA project. A critical question many organizations are facing now is – what are the critical success factors for SOA implementations? This research aims to identify a list of factors relating to SOA implementation success. A Delphi study forms the research method, and inputs regarding SOA critical success factors are requested from a panel of experts.

Keywords

Service-Oriented Architecture, Critical Success Factors, Delphi Study

INTRODUCTION

Businesses seek to improve productivity with the same or smaller operating budgets. Cost-overruns and time-overruns are common occurrences in systems development projects. For large systems development projects, inaccurate project planning and scheduling can leave businesses vulnerable. A number of relatively new means of developing and managing Information Technology (IT) includes Service Oriented Architecture (SOA), Web Services, Mashups, Ajax, and Web 2.0. Some of their underlying middleware realization schemas such as SOAP (Simple Object Access Protocol), UDDI (Universal Description, Definition, and Integration), XML Extensible Markup Language), and CORBA (Common Object Request Broker Architecture), among others, have burst onto the IT (Information Technology) scene. These approaches to information system architectures seem to have become the buzzwords of the day for many in the business world as well as in the Information Technology (IT) and Information Systems (IS) communities.

When it comes to adopting and implementing SOA, more and more businesses are concerned with the success of such implementations. For example, Dorman (2007), Havenstein, (2007), Ricadela (2006), and Trembly (2007) indicate that there is doubt emerging as to the real value of SOA to adopting businesses and organizations. In other words, organizations are asking themselves, “How can we ensure successful SOA implementations?” Are there a number of critical success factors that we should be aware of?

We begin with an understanding that the definition of SOA itself is somewhat unclear, and propose to use Erickson and Siau’s (2008) framework as a basis for this effort. We propose to execute a Delphi study to elicit expert ideas on identifying and ordering a set of Critical Success Factors (CSFs) for SOA. This effort should be considered as a research in progress, since the Delphi panel is now being recruited and the initial questionnaires are currently under development and will be piloted in late spring before the full study is implemented during the summer. Preliminary results should be available by August. The paper is organized as follows. Section 1 introduced the research. Section 2 will detail a number of the existing definitions of SOA, and briefly list the technologies typically included in SOA. Section 3 introduces the proposed SOA framework, and Section 4 reviews the current SOA literature. Section 5 states the research question and proposes the theoretical framework. Section 6 details the Delphi methodology proposed for the research methodology, and Section 7 discusses the expected results and contribution of the research.

SOA DEFINITIONS

Definitions of SOA

Even though organizations and businesses have thought in terms of Services for at least a decade (Web or IT Services), the name Service Oriented Architecture seems to have not been in general use for the most part before the late 1990s or early 2000s. However, SOA has subsequently burst onto the scene in almost all businesses and industries globally. SOA seems to have developed from an IT infrastructure area called Web Services. Web Services are often made up of technologies such as

SOAP, CORBA, EJB (Enterprise Java Beans), DCOM (Distributed Component Object Model), and even SIP (Session Initiated Protocol) among many others. Sulkin (2007), noted that Web Services often include middleware applications created with XML, and so XML is sometimes called an enabling technology for Web Services or SOA.

A number of formal industry-based definitions of SOA exist from standards bodies such as the OASIS Group, the Open Group, XML.com, javaworld.com, the Object Management Group (OMG), W3C (World Wide Web Consortium) and IBM, as well as other sources such as Wikipedia, Webopedia, TechEncyclopedia, WhatIs.com, and Webopedia.org. The standards groups' definitions are particularly important because they quite often form the basis upon which the various SOA platforms are built. It is also critical that at some point in the future, the various definitions converge on (ideally) a single definition of SOA, and the underlying Services. Finally, the various definitions are included here to illustrate the lack of coherence in the entire area of SOA at the moment.

The Open Group defines SOA as “an architectural style that supports service orientation” (Open Group, 2007). The definition goes on to also include descriptions of architectural style, service orientation, service, and salient features of SOA. The OASIS Group defines SOA as “...a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains” (OASIS, 2006). The OASIS definition includes what they call a “reference model” in which the details of the definition are expanded and formalized.

The Object Management Group (OMG) defines SOA as “...an architectural style for a community of providers and consumers of services to achieve mutual value” (OMG, 2007). OMG adds that SOA allows technical independence among the community members, specifies the standards that the (community) members must agree to adhere to, provides business and process value to the (community) members, and “allows for a variety of technologies to facilitate (community) interactions” (OMG, 2007).

W3C defines SOA as “a form of distributed systems architecture that is typically characterized by...a logical view, a message orientation, a description orientation, granularity and platform neutrality.” W3C adds details describing what they mean by logical view, message and description orientations, granularity and platform neutrality.

XML.com defines SOA as follows: “SOA is an architectural style whose goal is to achieve loose coupling among interacting software agents. A service is a unit of work done by a service provider to achieve desired end results for a service consumer. Both provider and consumer are roles played by software agents on behalf of their owners (XML.com, 2007).

The javaworld.com SOA definition, composed by Kodali, is as follows, “Service-oriented architecture (SOA) is an evolution of distributed computing based on the request/reply design paradigm for synchronous and asynchronous applications”. Kodali goes on to describe four characteristics of SOA. First, the self-describing interfaces composed in XML, using WSDL (Web Services Description Language) for the self-description. Second, XML schema called XSD should be used for messaging. Third, a UDDI-based registry maintains a list of the services provided. Finally, each service must maintain a level of quality defined for it via a QoS (Quality of Service) security requirement.

Finally, IBM proposes that SOA “describes a style of architecture that treats software components as a set of services.” Further, IBM argues that business needs should “drive definition” of the services, and that the value proposition be centered around the re-usability and flexibility of the defined services.

Existing SOA Framework and Principal Components

Krafzig, Banke, and Slama (2005) proposed a three-level hierarchical perspective on SOA, in which Level 1 includes the Application Front End, the Service, the Service Repository, and the Service Bus. Accordingly, only the Service child (Level 2) has children, consisting of Contract, Implementation, and Interface. Finally, Level 3 of the proposed hierarchy is composed of Business Logic and Data, children of Implementation.

A NEW SOA FRAMEWORK

It became apparent to the investigators through the initial portion of this research that Service Oriented Architecture is commonly seen as an architecture, or way of assembling, building, or composing the information technology infrastructure of a business or organization (Erickson and Siau, 2008). We propose that the framework depicted in Figure 1 can be used to understand SOA.

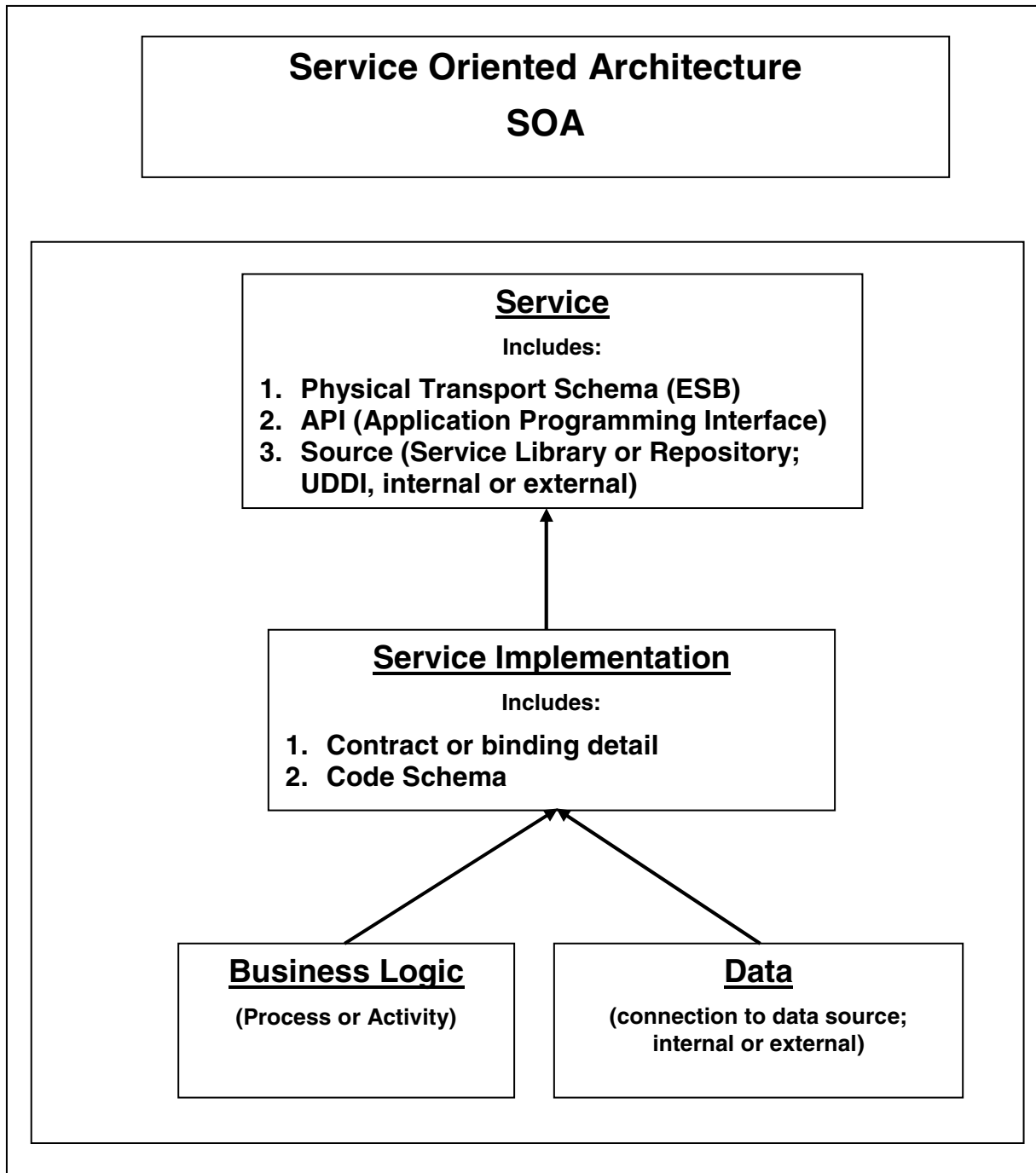


Figure 1: SOA Framework

LITERATURE REVIEW

SOA research is quite limited currently. The existing body of work can be classified into several distinct categories. The first type of research examines and recommends implementation strategies related to SOA. These efforts typically include exploratory or recommendation-type research that proposes various means to approach SOA implementation, and may or may not include patterns or blueprints. A second type of research into SOA involves proprietary or non-proprietary tool development. These investigations may or may not include proprietary industry software, but most of these research efforts also propose use of patterns or blueprints, and may overlap somewhat with the first research category. A third type of research examines the existing SOA standards, or proposes new standards designed to solve a problem or shortcoming identified in the existing standards. Also included in this category are proposals for middleware development, support or modification in support of SOA efforts. Finally, the last research category represents a higher level approach to SOA research, and proposes new meta-models or changes to existing meta-models. Most importantly for this research, no research effort that we are aware of has attempted to assess SOA critical success factors, at least in metrical or financial terms. Table 1 depicts a classification of past SOA research into the above categories. The year of the research can be found in the References.

Author(s)	Primary Focus of SOA Research			
	Implementation Strategies; Patterns and Blueprints	Tool Development	Standards Proposals or Modifications (including Middleware)	Ontological or Meta-Model Development or Modification
Stahl	Y		Y	Y
Kim and Lim	Y		Y	
Shan and Hua	Y		Y	
Schmidt, et al.	Y		Y	
Crawford, et al.	Y		Y	
Brown, et al.	Y	Y		
Ferguson and Stockton	Y	Y		
De Paw, et al.	Y	Y		
Jones		Y	Y	
Chen, et al.				Y
Borker, et al.			Y	
Duke, et al.				Y
Zhang		Y	Y	
Malloy, et al.			Y	
Verheeke, et al.			Y	
Hutchinson, et al.	Y			
Li, et al.	Y			

Table 1: SOA Research Classification

SOA Implementation Strategies; Patterns and Blueprints

Stal (2006) advocated using architectural patterns and blueprints (software engineering patterns) as a means to enable or foster efficient deployment of SOA. He supported loose coupling of services in a registry or library to the extent that he thought that removing the services' dependency on the registry's or provider's distinct location would benefit the deployment of SOA. Stal maintained that this would eliminate, or at least minimize a layer in the SOA framework. He also proposed a more tightly defined and controlled integration of middleware using XML or similar tools. Basically, Stal suggested a metamodel and pattern approach to defining SOA, but did not suggest what the research might accomplish, or how the research into SOA would be framed. Additionally, since Stahl suggested changes to the SOA meta-model and also proposed changes to SOA standards, his research fits into at least 3 of the arbitrary classifications identified previously. Kim and Lim (2007) also proposed a distinct means to implementing SOA, using in this instance, Business Process Management, in addition to a variant of the SOA framework specifically dealing with the telecommunications industry. Similar to Stal (2006), Kim and Lim (2007) did not propose empirical research into SOA, but rather focused on implementation and standards in a specific industry.

Shan and Hua (2006) followed a now common form in that they proposed an SOA approach for the Internet banking industry. They also compiled a list of patterns that have been proven to be successful for other online service industries. However, the models they used and end with are very detailed regarding how SOA should be implemented for first online companies in general, and then Internet banking specifically. This again does not propose or frame specific research but rather suggest an implementation approach and a structure for SOA.

The Enterprise Service Bus (ESB) is explained in detail, but from a general perspective, rather than a company specific approach in Schmidt, Hutchison, Lambros and Phippen's (2005) expository. The article is informative regarding ESB implementation and design patterns, but it is not research oriented.

Crawford *et al.* (2005) proposed a different way to structure SOA and they called it on-demand SOA. They essentially proposed an even greater loose coupling of services and their connecting elements, than other perspective of SOA, and that would allow much more flexibility for the adopting organizations and the end-users.

Hutchinson, Henzel, and Thwaites (2006) described a case in which an SOA based system was deployed for a library extension collaboration project. Much of the case details the SOA approach itself, and explains the experiences of the project developers and implementers. They noted that while the SOA architecture could be expected to reduce the operational maintenance costs overall, the way the system was specified and delivered in this particular case might require more work from IT to keep some services, such as flash players, etc. up to date. While it does not specifically mention it in the article, perhaps a more loosely coupled architecture might alleviate some of those operational maintenance costs.

Li *et al.* (2007), proposed a methodology to migrate the functionality of legacy systems to Web Services or SOA architecture. They used a case study to investigate the efficacy of their proposed methodology, finding that while it was possible to make such a migration from legacy to SOA (or Web Services), the changes that it required from the organization were considerable, and that some process re-engineering would likely be necessary.

Tool Development

Brown *et al.* (2005) presented an industry oriented perspective on the SOA puzzle. They suggested an approach to service orientation using the proprietary IBM Rational platform. Their recommendations follow similar paths as previous researches, but are also filtered through the IBM Rational lens. The article is primarily illustrative in nature, suggesting how to best implement SOA using IBM Rational tools. In a similar vein, Ferguson and Stockton (2005), also detail IBM's programming model and product architecture.

De Pauw *et al.* (2005) described the benefits of Web Services Navigator, a proprietary tool created to provide a better visualization of SOA and Web Services in a loosely coupled architecture. The tool can help with design pattern, business logic, and business process analysis, and thus, help with SOA architecture design and implementation.

Jones (2005) suggested that SOA, Service, and Web Service standards were "on the way" and provided a list of existing tools, such as UML and/or the Rational Unified Process that could aid the SOA (or service) design process. However, he also advocated the push toward formal definitions of such SOA basics as services, to the end of providing a more coherent and cohesive structure that he thought would enhance the ability of developers and adopters to understand and deploy SOA.

Standards Proposals or modifications (including middleware proposals)

Borker *et al.* (2006) suggested a way of handling XML-based data in an SOA or service environment. Their idea involved the use of queryable and unqueryable data and would necessarily also involve XML formatted data. This represents empirically based research into a part of SOA, namely the underlying services.

Zhang (2004) explored the connection between web services and business process management, and described the modular nature of the service (and Web Service) perspective. He detailed the software industry's approach to Web Services, and provided evidence that standards development would quickly mature, beginning in 2005. He maintained that once standards were agreed upon, a connection to Business Process Management would be easier to sell to business. Zhang's efforts go a bit further than others, because he did not stop at this point, but developed a prototype e-procurement system that composed external service to operate.

Malloy *et al.* (2006) developed an extension to WSDL (Web Services Description Language). They insisted that Web Services' specifications were "...typically informal and not well-defined," and proposed what they called an intermediate step between requiring more formal and rigorous service specifications and the informal nature of the existing service specifications. They accomplish this balance by extending the WSDL to include support for application argument and return that would help automate and expand the ability of services to operate in multiple environments. They provided an example of how their WSDL extension could allow a single service to function successfully in different applications using multiple zip code formats (5 versus 9 digits, and hyphens versus no hyphens).

Verheeke, Vanderperren, and Jonckers (2006) proposed and developed a middleware level that they called Web Services Management Layer (WSML). They saw the primary advantage of their approach in that it provided a reusable framework. They further believed that the use of their framework would enable "...dynamic integration, selection, composition, and client-side management of Web Services in client applications" (p. 49). They were aware that their approach could cause some problems in a distributed system, since implementation of it resulted in a centralized architecture.

Ontological or Meta-Model Development or Modification

Chen, Zhou and Zhang proposed an ontologically based perspective on SOA, Web Services and Knowledge Management (2006). They attempted, with some success, to integrate two separate research streams into one. They presented a solution to show that semantic and syntactic based knowledge representations could both be depicted with a comprehensive ontology that also described web service composition. While their framework represents a step toward automated (web) service composition, more research is still needed.

Duke, Davies and Richardson (2005) recommended and provided details on using the Semantic Web to organize an organization's approach to SOA and Web Service orientation. They suggested that combining the Semantic Web and SOA into what they called Semantic SOA would provide benefits to adopting organizations. Then, they further proposed an ontological model of the semantic SOA, attempting essentially to create a meta-meta-model of SOA, using their experience with the telecommunications industry as a case example.

RESEARCH QUESTION AND THEORETICAL FOUNDATION

Research Question

What are the critical success factors involved with SOA implementations? Traditionally, software (or system) successes and failures have been estimated by the "usual suspects" such as return on investment (ROI), net present value (NPV), breakeven, internal rate of return (IRR), or other similar financially based approaches. These approaches are all based firmly on the idea that if we can assign some numbers to a system, then we can compare them across projects, systems or organizations. But numbers are not always the best measure. In this research, we aim to identify a list of critical success factors for SOA implementations. A Critical Success Factors (CSF) analysis tries to identify a small number of goals or factors that, if they are reached or successfully completed, will tend to predict or indicate success (Rockart, 1979). While CSF analyses have been often used to elicit information requirements in the systems analysis and design phase of systems development, they could also be used to identify project success factors as well.

The Input Process Output (IPO) framework (Hackman, 1987) is useful in this research for understanding what the developers' or proponents' thoughts are on SOA success factors during Input, Process, and Output. Hackman (1987) developed the IPO model or framework to explain the effectiveness of teams in organizations. The model proposes that a variety of factors form the inputs, the group then engages in its processes, and forms the outputs. The inputs were typically characterized as performance goals, the processes as task cohesion, and the outputs as the actual performance. The

beginnings of the IPO framework extend back to at least the 1960s and have been used in various forms almost continuously until the present, according to Yu (2005).

RESEARCH METHODOLOGY

We will use the Delphi research method to solicit the inputs from a panel of experts. The Delphi method is a systematic and interactive research method to obtain the “correct” answer from a panel of independent experts. The carefully selected experts answer questionnaires in two or more rounds. After each round, a facilitator provides an anonymous summary of the experts’ answers from the previous round as well as the reasons they provided for their judgments. Thus, participants are encouraged to revise their earlier answers in light of the replies from other members of the group. It is believed that during this process the diversity of the answers will decrease and the group will converge towards the “correct” answer. Finally, the process is stopped after achieving a pre-defined stop criterion (e.g., number of rounds, achievement of consensus, and stability of results). In our case, the stopping criterion will be the achievement of consensus. The final results from the experts will consist of a list of critical success factors that the expert panelists agree upon as being critical for successful SOA project implementation.

Usually, the group of experts is geographically dispersed, as will be the case for many of the subjects participating in this research (i.e., different companies and institutions). Generally, Delphi studies attempt to form a reliable consensus of a group of experts in a specialized area (Ludwig, 1997). The approach is a process that focuses on collecting information from the expert group through a series of questionnaires, and providing feedback to the group between questionnaires. Linstone and Turoff (2002) categorized Delphi studies into 2 different types. The first is what they call a conventional Delphi, in which the study is executed via paper and pencil; this form of Delphi can consume much time. The second form a Delphi study can take is termed “real-lucré” by Linstone and Turoff (2002). This form of Delphi is computer administered, in what amounts to real-time. This research will be a combination of the two approaches in that it will likely be administered via computer, but will not be completed in real time or even in one sitting.

Many past and present researchers have used the Delphi approach to study various topics. According to Lawrence Day (2002), Delphi studies have been used to examine and predict future advances in computers and technology, cosmetics, insurance, and recreation. AT&T studied the future of the telecommunications industry, MacMillan focused on the future of newsprint, and Skandia Insurance tried to identify and rank economic losses related to computer systems (Day, 2002).

Kunne (2002) indicated that more than 463 research efforts used the Delphi method between 1975 and 1994. The ENRiP (Exploring New Roles in Practice) project sponsored by the University of Sheffield School of Nursing and Midwifery used a Delphi study to assess new roles in the practice of nursing (2001). The use of Delphi studies appears to spread across industries and time, while the methodologies tend to split into two camps; developing a consensus on the future, or developing a consensus on such areas as regional planning (Kunne, 2002).

Delphi Steps

Delphi studies typically include ten steps (Nehiley, 2003):

- i. Formation of the team (the researchers in this case)
- ii. Selection of subjects
- iii. Development of the first round questionnaire
- iv. Pilot test the questionnaire for wording, etc. (this will be accomplished via a small group of “local experts”)
- v. Send first round to participants
- vi. Analyze first round responses
- vii. Construct (and test if necessary) the second round questionnaire
- viii. Send the second round to participants
- ix. Analyze second round responses (repeat steps 7, 8, and 9 until the desired consensus level is reached; this will likely mean two or three iterations for this research)
- x. Prepare the final analysis and report.

EXPECTED CONTRIBUTION

SOA has tremendous potential and possibilities. Practitioners have started to explore and implement SOA. Nevertheless, ensuring SOA implementation success has always been a concern. As one of the first research efforts in the area, the set of Critical Success Factors resulting from this study will be of tremendous values to practitioners who are thinking, exploring, or contemplating SOA implementation as well as practitioners who are already in the process of implementing SOA in their organizations. Academicians will benefit from the research too as this study will inform literature, answer some existing questions on SOA, highlight new research questions, and move the SOA field forward.

REFERENCES

1. Borker, V., Carey, M., Mangtani, N., McKinney, D., Patel, R., and Thatte, S. 2006. "XML Data Services." *International Journal of Web Services Research*. Vol. 3. No. 1. pp. 85-95.
2. Brown, A., Delbaere, M., Eeles, P., Johnston, S., Weaver, R. 2005. "Realizing Service Oriented Solutions with the IBM Rational Software Development Platform." *IBM Systems Journal*. Vol. 44. No. 4. pp. 727-752.
3. Chen, Y., Zhou, L., and Zhang, D. 2006. "Ontology-Supported Web Service Composition: An Approach to Service-Oriented Knowledge Management in Corporate Financial Services." *Journal of Database Management*. Vol. 17. No. 1. pp. 67-84.
4. Crawford, C., Bate, G., Cherbakov, L., Holly, K., and Tsocanos, C. "Toward an on Demand Service Architecture." *IBM Systems Journal*. Vol. 44. No. 1. pp. 81-107.
5. Day, L., 2002, "Delphi Research in the Corporate Environment", In H. Linstone and M. Turoff (eds.), *The Delphi Method Techniques and Applications*, <http://www.is.njit.edu/pubs/delphibook/index.html>.
6. De Pauw, Lei, M., Pring, E., and Villard, L. 2005. "Web Services Navigator: Visualizing the Execution of Web Services." *IBM Systems Journal*. Vol. 44. No. 4. pp. 821-845.
7. Dorman, A. 2007. "FrankenSOA." *Network Computing*. Vol. 18. No. 12. pp. 41-51.
8. Duke, A., Davies, J., and Richardson, M. 2005. "Enabling a Scalable Service oriented Architecture with Semantic Web Services." *BT Technology Journal*. Vol. 23. No. 3. pp. 191-201.
9. ENRiP web site, 2001, University of Sheffield, <http://www.snm.shef.ac.uk/research/enrip/refs.htm>.
10. Erickson, J., Lyytinen, K., and Siau, K. 2005. "Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research." *Journal of Database Management*. Vol. 16. No. 4. pp. 80-89.
11. Erickson, J. and Siau, K. 2008. "Web Services, Service Oriented Computing, and Service Oriented Architecture: Separating Hype from Reality." *Journal of Database Management*. Vol. 19. No. 3. pp. 42-54.
12. Ferguson, D. and Stockton, M. 2005. "Service oriented Architecture: Programming Model and Product Architecture." *IBM Systems Journal*. Vol. 44. No. 4. pp. 753-780.
13. Hackman J. 1987. "The design of work teams. In *Handbook of Organizational Behavior*, ed. JW Lorsch, pp. 315-42. Prentice-Hall.
14. Havenstein, H. 2006. "Measuring SOA Performance is a Complex Art." *Computer World*. Vol. 40. No. 2. pp. 6.
15. Hicks, B. "Oracle Enterprise Service Bus: The Foundation for Service Oriented Architecture." Retrieved from: http://www.oracle.com/global/ap/openworld/ppt_download/middleware_oracle%20enterprise%20service%20bus%20foundation_250.pdf on 11-18-2007.
16. Hutchinson, B., Henzel, J., and Thwaites, A. 2006. "Using Web Services to Promote Library-Extension Collaboration." *Library Hi Tech*. Vol. 24. No. 1. pp. 126-141.
17. Jones, S. 2005. "Toward an Acceptable Definition of Service." *IEEE Software*. May/June. pp. 87-93.
18. Kim, J. and Lim, K. 2007. "An Approach to Service Oriented Architecture Using Web Service and BPM in the Telcom OSS domain." *Internet Research*. Vol. 17. No. 1. pp. 99-107.
19. Krafzig, D., Banke, K., and Slama, D. 2005. "SOA Elements." Prentice Hall. Retrieved from: http://en.wikipedia.org/wiki/Image:SOA_Elements.png on 12-02-2007.

20. Kunne web site, 2002, accessed 2003 <http://www.kunne.no/meritum>.
21. Li, S., Huang, S., Yen, D., and Chang, C. 2007. "Migrating Legacy Information Systems to Web Services Architecture." *Journal of Database Management*. Vol. 18. No 4. pp. 1-25.
22. Linstone, H., and Turoff M., 2002, "Introduction", in: Linstone, H., and Turoff, M., eds. *The Delphi Method Techniques and Applications*, pp 4-6.
23. Ludwig, B., 1997, "Predicting the Future: Have you considered using the Delphi Methodology?" *Extension Journal*, October, Vol. 35, No. 5.
24. Malloy, B., Kraft, N., Hallstrom, J., and Voas, J. 2006. "Improving the Predictable Assembly of Service Oriented Architectures." *IEEE Software*. March/April. pp. 12-15.
25. Nehiley, J., 2003, *How to Conduct a Delphi Study*, <http://extmarket.ifas.ufl.edu/FOCUS/html>, accessed 1/10, 2008.
26. OMG (Object Management Group). 2007. Retrieved from: <http://colab.cim3.net/cgi-bin/wiki.pl?OMGSoaGlossary#nid34QI> on 11-25-2007.
27. OASIS (Organization for the Advancement of Structured Information Standards). 2006. Retrieved from: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm on 11-25-2007.
28. Open Group. 2007. Retrieved from <http://opengroup.org/projects/soa/doc.tpl?gdid=10632> on 11-25-2007.
29. Ricadela, A. 2006. "The Dark Side of SOA." *Information Week*. September 4th. pp. 54-58.
30. Rockart, J. 1979. "Chief Executives Define Their Own Data Needs." *Harvard Business Review*. March-April.
31. Schmidt, M., Hutchison, B., Lambros, P., and Phippen, R. 2005. "Enterprise Service Bus: Making Service Oriented Architecture Real." *IBM Systems Journal*. Vol. 44. No. 4. pp. 781-797.
32. Shan, T. and Hua, W. 2006. "Service Oriented solution Framework for Internet Banking." *Internet Journal of Web Services Research*. Vol. 3. No. 1. pp. 29-48.
33. Siau, K. 2003. "Interorganizational Systems and Competitive Advantages – Lessons from History," *Journal of Computer Information Systems*, Vol. 44, No. 1, pp. 33-39.
34. Stal, M. 2006. "Using Architectural Patterns and Blueprints for Service oriented Architecture." *IEE Software*. March/April. pp. 54-61.
35. Sulkin, A. 2007. "SOA and Enterprise Voice Communications." *Business Communications Review*. Vol. 37. No. 8. pp. 32-34.
36. Trembly, A. 2007. SOA: Savior or Snake Oil?" *National Underwriter Life & Health*. Vol. 111. No. 27. pp. 50.
37. Verheeke, B., Vanderperren, W., and Jonckers, V. 2006. "Unraveling Crosscutting Concerns in Web Services Middleware." *IEEE Software*. January/February. pp. 42-50.
38. W3C (World Wide Web Consortium). 2007. Retrieved from: <http://colab.cim3.net/cgi-bin/wiki.pl?WwwCSoaGlossary#nid34R0> on 11-25-2007.
39. Wikipedia (SOA). 2007. Retrieved from: http://en.wikipedia.org/wiki/Service-oriented_architecture#SOA_definitions on 9-25-2007.
40. XML.com. 2007. Retrieved from: <http://www.xml.com/pub/a/ws/2003/09/30/soa.html> on 11-25-2007.
41. Yu, C. 2005. "An I-P-O Model of Team Goal, Leader Goal Orientation, Team Cohesiveness, and Team Effectiveness." Dissertation. Texas A&M University. Retrieved from: <http://txspace.tamu.edu/bitstream/handle/1969.1/3222/etd-tamu-2005C-MGMT-Yu.pdf?sequence=1> on 3-03-2008.
42. Zhang, D. 2004. "Web Services Composition for Process Management in e-Business." *Journal of Computer Information Systems*. Vol. 45. No. 2. pp. 83-91.