

# Service Oriented Requirements Engineering: Practitioner's Perspective

Muneera Bano Sahibzada and Didar Zowghi

Research Center for Human Centered Technology Design  
Faculty of Engineering and Information Technology  
University of Technology Sydney, Australia  
{Muneera.Bano,Didar.Zowghi}@uts.edu.au

**Abstract.** Over a decade ago Service Oriented Architecture (SOA) was introduced to provide better alignment between business requirements and IT solutions. During this period a great deal of research interest has emerged from academia and industry alike, to promote this new style of software development. The promise was that SOA based development will improve reusability, agility, platform independence and dynamic discovery, reconfiguration and change management. In spite of all the promises and enhancement in tools and technologies, the service oriented software development continues to face various challenges especially in Requirements Engineering. In this paper we present a qualitative study of Service Oriented Requirements Engineering. Data was collected by conducting interviews with practitioners from IT companies in Sydney, who are experienced in working on SOA based projects. The objective was to explore the issues and challenges faced during requirements analysis in service oriented software development. The results show that Service-Oriented software development has not only inherited existing issues of traditional Requirements Engineering but has also introduced new challenges. The technology has become advanced in SOA but the issues related to the organizational and business aspect of service oriented development need more attention for achieving true benefits of this technology.

**Keywords:** Service Oriented Software Engineering, Requirements Engineering, issues and challenges.

## 1 Introduction

The Service Oriented Architecture (SOA) based software development was introduced with promises to bring into reality the true form of reusability along with agility, platform independence and dynamic discovery, reconfiguration and change management. Service Orientation is referred in IT paradigm by various terms such as Service Oriented Architecture (SOA), Software as a Service (SaaS), Service Oriented Computing (SOC), and Service Oriented Software Engineering (SOSE). In current market conditions the development teams are under pressure to meet deadlines while producing quality software within approved budget that would at the same time satisfy the customers. Reusability of

software was considered to reduce cost and time for development and various solutions emerged as a result of research on development methodologies. Service-Oriented software development is an evolutionary form of Component based Software Development (CBSD) and Object Oriented (OO) development with the difference that it uses web services instead of objects or packaged components [14]. The use of service requires a contractual agreement between service provider and service consumer. Reusing existing components in form of services in collaborative and distributed environment presented a very interesting perspective for saving time and cost of development. But at the same time it introduces a number of new software engineering challenges [1]. A great deal of research efforts were put into the directions of achieving the true spirit of SOA from academia and industry alike [10] [11].

According to Zave [20], Requirements Engineering (RE) can be defined as: *“the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families.”*

RE is considered the most crucial part of software development that establishes the goals and objectives of the system by consulting with all the relevant stakeholders. If these goal and objectives are not articulated correctly at the outset, the end product can not fully satisfy the intended users, no matter how much time and resources are spent for the development. According to Karl Weigner [21], *“If you don't get the requirements right, it doesn't matter how well you do anything else.”*

The term Service Oriented Requirements Engineering (SORE) was first used in 2003 by Eck and Wieringa [2], to show that it is different from traditional RE, with two different dimensions as a result of separation of service development and Service-Oriented software development. Service-Oriented solutions have to consider both service provider and service consumer. The service provider needs to understand the functional and non-functional aspects of the service being offered, which has to compete in marketplace with other services. This requires understanding of potential consumers of that service. For service consumers, the challenge is to find the correct solution to the business needs of the organization.

In spite of all the improvements in tools and technologies, the service oriented software development is facing various challenges especially in requirements engineering. Technology is an essential part of providing solutions in form of services, but ignoring the importance of enterprise's business and IT requirements would result in project failure. In SOSE, the developers have software components ready for them in form of services, available over internet or intranet. After requirements elicitation, the ideal situation is to align service to meet business requirements. The organizations require quick solutions and the developers are usually under pressure of tight deadlines. Instead they try to provide solution where requirements are compromised based on available services. This would result in unsatisfied customers and failure of project even though the technology is highly promising.

In this paper, we present findings of interviews conducted from 13 practitioners from different companies in Sydney, who have experience in SOA based software development. The interviews were part of our larger exploratory research project. The objective was to explore the issues and challenges faced by the practitioners in industry in RE phase during Service-Oriented software development.

The rest of the paper is organized as follows; Section 2 outlines our research strategy that we undertook for our investigation. Section 3 gives details of interview design. Section 4 presents analyzed results from interviews. In Section 5, we discuss the various implications of the results. Section 6 presents conclusion and future work.

## 2 Research Strategy

Our interviews were one part of a larger research plan. The overall aim of our research project is to improve RE in Service-Oriented software development. We divided the task in two phases. The first phase is to understand the problem space and the second is to propose a solution that would bring improvements to the problems of RE in SOSE. Figure 1 depicts all the steps involved in both phases along with their objectives. We have completed Phase 1 and in current study we are presenting the results of industrial interviews, which is the last step of phase 1.

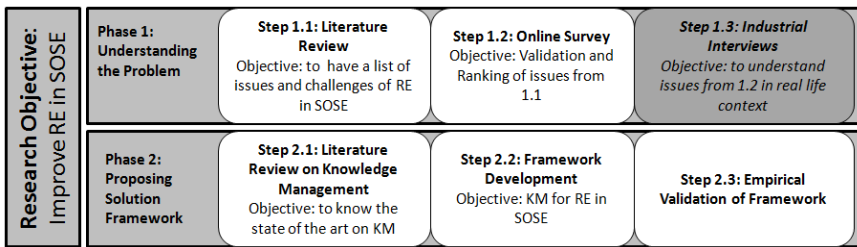


Fig. 1. Complete Research Plan

Our exploration of understanding the problem phase was carried out in three steps. First we performed a comprehensive review of research literature [14], in order to develop a list of issues and challenges reported. As a second step we conducted an online survey [15] [16], and asked the practitioners to confirm and rank the existence of the issues and challenges that we had identified in step 1. Based on the results from survey, we developed a series of questions and approached 13 practitioners working in different IT companies in Sydney for face-to-face interviews. Following is a brief description of the first two steps.

### 2.1 Literature Review

SOSE had been the subject of research community from different perspectives. Many methods, techniques and tools have been proposed by different mega projects and research teams. These efforts include “Service Centric System Engineering (SeCSE [3])”, “Service Oriented Development In a Unified fraMework (SODIUM [4])”, “Software Engineering for Service Oriented Overlay Computers (SENSORIA [5])”, “Service Oriented Modeling and Architecture (SOMA) [6][7]”, “Service Oriented Architecture Framework (SOAF [8])”, “Service Development Lifecycle Methodology (SDLM [9])”. The real benefits of any research efforts can only be verified when it is

applied in a real setting. With the exception of few (e.g. IBM), most of the methods proposed to date are not adapted by industry [11], and hence lack empirical validation of their true value. Only 8 challenges of RE in SOSE have been investigated empirically till 2008 [12]. Failure of the Universal Description Discovery and Integration (UDDI) [13] and the incident of Amazon web service failure last year [21] has put a question mark on the whole idea of service oriented solutions. There is a need for further empirical work in this area with real life projects to provide feedback for improvement in current methods and practices and also to enrich the knowledge in SOA domain and open further research directions [14].

The aim of our literature review was to extract the list of issues and challenges of RE in SOSE that has been reported in published research. We conducted an exhaustive search for publications that were available on online electronic databases. Their inclusion in our review was based on the criteria that they should be pointing out any problem or challenge that is faced during RE for Service-Oriented software development. The challenges identified from the literature were published in [14].

## 2.2 Online Survey

The purpose of conducting the online survey was to validate the list of issues identified in the research literature, from practitioners working on Service-Oriented system development. The population comprised of the practitioners who have worked on Service-Oriented projects either as technical team member or as a researcher. Due to the fact that it was online web survey, we used convenient and non probabilistic sampling. The instrument for survey was questionnaire based on the identified factors from issues. The items in questionnaire used Likert scale of five levels to measure agreement to the issues. We administered the survey on the web and sent the link through email to invite the practitioners around the world using online special interest groups. A total of 117 responses were received world wide. The survey provided us with verification and ranking of the issues and we were able to refine our list of issues from step 1 so that it includes the important issues on which they all agreed. The results of the analysis of survey data can be summarized in the identification of the following top five important issues [15] [16]; (1) *Alignment of business requirements and services* (2) *Non functional requirements gathering and assessment* (3) *Iterative service discovery process* (4) *Integration of Knowledge Management strategy to SOSE life cycle* (5) *Requirements change management*.

## 3 Interview Design

Interviews are considered to provide rich qualitative data when a deeper insight of a phenomenon is required. Our survey provided us with surface level of understanding about the issues of RE in SOSE. Interviews were our choice for a deeper and vertical understanding of the issues in their real life context. It was the best choice to get the opinion and reflections of practitioners by talking to them face to face and giving them freedom to express their thoughts on the subject. Based on the refined list from online survey, we designed the interview questions. We presented these issues to the interviewees and asked to them to provide their view on it with reference to the projects on which they have worked.

The criterion for participation was that the interviewee should be a practitioner in industry and should have experience of working as a development team member in Service-Oriented project. The practitioners were contacted using social networking approach followed by snow-ball or referral sampling technique. The interviews were semi structured and open ended. The interviewees were asked to provide their views on the issues with reference to the examples of the SOA based projects they had experienced. The interview questions were focused on the five issues identified (section 2.2) but for further understanding we took advantage of face-to-face interview style to raise follow up questions where ever it was required. Out of these 13 interviews, 2 were conducted online using Skype and rest were conducted face-to-face. Interviews took place during June-August 2012 in Sydney and each lasted between 30 minutes to one hour. They were audio recorded and partly transcribed. The names of the interviewees, the companies and the projects will not be disclosed in this paper as per confidentiality requirement of Human Research Ethics Committee (HREC) at UTS [17] for approval of this research. We will present here the results that we obtained after the qualitative analysis.

The 13 interviewees, who participated in this study, had diverse job experience in various IT based companies. Following are the profiles of their job experience;

1. *Program Manager at a Telecom Company in Sydney; who also owns a small consultancy company where the two Business Analysts are present in Sydney and rest of the development team is in India. The Interviewee had 20 years of working experience in Software industry in Sydney and has managed various Service Oriented Projects.*
2. *Independent ICT consultant in a large organization; also having past experience of working as Project Manager, Senior Business Analyst and Developer.*
3. *Business Architect; having over 10 years of experience working in Software industry as a Programmer, Business Process Analyst, Business Analyst, and IT Consultant. Currently the focus of his work is on integrated context-aware adaptive cloud computing environments (e.g. integrated cloud, social networks, mobile computing) for e-Government and e-Business Solutions.*
4. *SOA integration specialist; who has worked on 4 SOA based projects in Canada.*
5. *Systems Engineering Manager; Director of a Systems Engineering and Safety Management Consulting Company.*
6. *Senior Equity Trading System Development Architect; having experience of building low level mission critical, fault tolerant systems as well as web n-tier based enterprise systems using Service Oriented Architecture based design patterns in finance, telecommunications, retail industries, research organizations.*
7. *Director of IT Consultancy company in Sydney; has over 20 years of experience in the IT industry, in Australia and the UK. The interviewee worked in sales and delivery, focusing specifically on SOA-related activities.*
8. *Business Analyst at IT based company providing solutions to Educational Sector; experience of working on multiple SOA related projects in UK and Australia.*
9. *Knowledge Engineer and Interim CTO (Chief Technology Officer), Coach, consultant and trainer in agile development, IT Leadership, project management, business analysis and business process improvement.*
10. *SOA Service Design Team Lead and Integration Architect/ Lead SOA Service Designer with more than 15 years of experience as Solution Architect, Designer and Developer.*
11. *Product Delivery Manager, Managing development team, ensuring delivery of software product in an agile and lean manner, grow and directing the team to be more efficient and self-running.*
12. *Director of IT Consultancy Company in Sydney, with experience of Enterprise Architect, J2EE Manager and Senior Java Architect.*
13. *Developer in IT based company in Sydney for the last two years; mainly responsible for the detailed designed and implementation of the projects and does not have direct influence on the decisions related to the RE phase but works alongside with business analysts in the team to provide guidance on the technical feasibility of the system requirements.*

## 4 Findings from Interviews

The results from all 13 interviews were analyzed using qualitative analysis technique such as content analysis. The transcriptions of the interviews were analyzed for the thoughts and concepts presenting for each of the issues. In following we present the aggregated results which emerged after analysis with respect to the challenges of RE in SOSE. Along with the results we present the table for comparison of our findings from all three steps of our exploration.

### 4.1 Alignment of Business Requirements and Services

One of the differences between SORE and traditional RE is the process of matching and selecting available services against requirements. A service is developed free of context and the environmental details where it would be used. It is up to the business analysts to understand the functionality of existing services, which is context free, and then align them to business requirements which have a specific context attached to them. All the interviewee considered alignment of business requirements to services as most challenging and important due to following reasons:

- The development team pushes the organizations to compromise or modify the requirements according to the available solutions.
- Due to uncertainties or ambiguities in elicited requirements, incorrect services get selected resulting in a system that is useless for the customers.
- The service granularity or abstraction sometimes does not align properly with the requirements, which makes the task more challenging.
- A consistent interpretation of business requirements is required for correct alignment to the service.

**Table 1.** Alignment of Business Requirements and Services

Findings from Literature Survey	Findings from Online Survey	Findings from Interviews	Conclusion
The main difference of SORE and RE is the alignment of business needs to available service, which is the most challenging part.	The alignment of business requirements and services is challenging due to the fact that they are not at the same level of granularity or abstraction.	The service granularity or abstraction sometimes does not align properly with the requirements, which makes the task more challenging. Sometimes the development team pushes the organizations to compromise or modify the requirements according to the available solutions.	This challenge is the most crucial from RE point of view as it would decide whether the resulting system is what it was suppose to be. According to our analysis each job role in development team has a different perspective on this issue. The Project Managers would deal with this issue with consideration to organizational policy. Business analysts would consider the business needs of the customer and performing gap analysis. The developers would think about the feasibility of the alignment while considering the available technical support.

## 4.2 Non Functional Requirements Gathering and Assessment

Interviewees stated that non functional requirements in SOSE should be the part of the Service Level Agreement. A service is usually developed for a group of customer without considering a specific context where it will be used. Every organization can have varying criteria for quality and therefore satisfying the non functional aspects (e.g. security, scalability, availability, reliability, performance etc.) for unknown group of customers is another challenging part of RE in SOSE. Usually it is required from the service provider to publish the security and trust related issues along with the functional aspects of the service but in many cases it never happens. Due to variance in the quality level of services, at the time of composition (integration), compatibility among services provided by different providers would be an issue. Customers are mostly not aware of these issues until they actually use the system. This increases the overhead of testing on the part of development team, whereas ideally a service is considered to have been tested for the quality related concerns. For most organization, the matter of trust is crucial when it comes to using services from a third party. The organization may not have an open policy for services from unknown service providers. According to one of the respondent, for safety critical software, the use of loosely coupled services from third party is not even feasible as they have a rigorous process for safety critical requirements evaluation and testing. In case of services, passing this strict evaluation would not be possible as the organizations are not usually provided with the source code.

**Table 2.** Non Functional Requirements gathering and assessment

Findings from Literature Survey	Findings from Online Survey	Findings from Interviews	Conclusion
NFR can be used as criteria for selection of service against requirements when multiple options are available but gathering and assessing NFR in Service-Oriented development is very challenging.	Service design needs to satisfy the required NFR while considering target consumer group but it is challenging as the users are unknown. If NFR are not satisfied it leads to changes in requirements after the system is deployed which can be very costly.	Services are developed without knowing the context of the environment in which they will be used. Every organization has its own criteria of minimum level of required quality. Testing effort is increased to assess quality for specific context. Some domains are safety critical; the idea of service orientation is not a feasible choice for them.	NFR such as security, scalability and reliability are the most important quality aspects in selecting services in a specific domain against a particular requirement. Organizational policy plays a crucial role in deciding the quality criteria for the system.

## 4.3 Iterative Service Discovery Process

The idea of iterative discovery process is useful when the development team has to elicit requirements from the stakeholders. The customers are usually not aware of what they actually want and keep changing their mind about their requirements. The requirements provided by them are usually incomplete and do not match the abstraction and granularity of the available service specification. Following an iterative service discovery process gives an opportunity to present available solutions, which are present in form of services,

to customers. They can provide their feedback and hence the services can work as prototypes and would help in completing the requirements, or analyze various options like trade-off between requirements and solutions, cost estimation etc. This method is feasible when we have a large central repository across organizations with service description available in structured form. But this process introduces additional concerns. If the development team is using services within organization, then they already know about the solutions they have. In iterative process, service providers try to encourage the customers into accepting the available solutions. From RE perspective, it might not be a good idea to jump to the solution space without fully understanding the problem space. For using third party services from a distributed central repository, this process would be helpful for business analyst in negotiating requirements with users. But the concept of global central repository like UDDI failed around 2006 due to various difficulties faced by developers in using it [13], so this idea might only be useful within organization with huge repository of services available.

**Table 3.** Iterative Service Discovery Process

Findings from Literature Survey	Findings from Online Survey	Findings from Interviews	Conclusion
As in traditional RE, incomplete requirements are an important challenge but in SOSE, we can start with flexible incomplete requirements and iteratively discover solutions in form of services while we move towards completing our requirements.	Flexible or incomplete requirements can give room to further issues. Some requirements cannot be flexible or compromised. Once requirements are complete discovering services to match those requirements would be more focused thus reducing any duplicate effort. Services are part of the solution space not the problem space, in moving back and forth we are not focusing on the problem properly.	This type of development lifecycle is feasible in large organizations with huge central repository with properly managed information about available services. The problem with this approach is that there is a chance that the development team would use iterative process to manipulate customers in accepting available solutions rather than focusing on what they actually want. The development team would immediately jump to solution rather than understand the problem properly.	Although services can be used as a prototypes to help in the process of elicitation, but the whole point of requirements engineering is to understand the problem space before jumping directly to the solutions. It would be depending largely on the intentions of the development team whether they want to make the process customer-oriented or to develop what is easy for them.

#### 4.4 Integration of Knowledge Management Strategy to SOSE Lifecycle

SOSE requires organizations to take initiative in using IT infrastructure of SOA to bring solutions to their business needs. Knowledge Management (KM) requires organizational strategy for identifying, acquiring, storing, and sharing the knowledge to improve the business and benefit to the organization. According to the interviewees, the benefits of KM are proven for project planning, document management and service versioning control. They pointed out that it is critical for an SOA based organization to take initiatives for implementing a well defined KM strategy to meet demands of dynamic market conditions where there is inevitable change in technology and human resource. In the last 20 years many methods, tools and techniques for KM



have been proposed in the research literature. All of the interviewee confirmed the use of various KM Systems, but only large organizations had a proper strategy for implementation of KM. According to the respondents, for small organizations KM brings overhead for the development team in terms of knowledge codification efforts. The respondents referred to it as: *“tedious hard work consuming a whole day for filling up various data forms”*. For the development team, there seems to be a need for great deal of visible effort for invisible benefits. The organizations have to keep track of the Knowledge Base (KB), and specialized personnel are needed for that purpose, whose task is to make sure that the KB is up to date.

**Table 4.** Integration of Knowledge Management Strategy to SOSE lifecycle

Findings from Literature Survey	Findings from Online Survey	Findings from Interviews	Conclusion
KM would help in improving RE in SOSE as it has been proved to bring improvements in RE for traditional software development and CBSD. But it requires time and efforts to implement a KM strategy in any organization.	KM would not eliminate all the issues completely, but would definitely improve the process and would help in better understanding of requirements and lead to better design and composition. The concepts of KM are already in use in SOSE e.g. service registry. But for a KM strategy resources in terms of cost, management and implementation time are required before actual benefits would be visible. There is also a concern that it might increase complexity of development lifecycle.	KM would help in improving RE in SOSE but organizational initiative is crucial for implementing a KM strategy. Also it increases work overhead on development team for managing knowledge. Specialized human resource is required to properly look after the KM strategy implementation. It is a lot of visible effort for development team for the benefits that are invisible to them.	It is challenging to implement KM as it requires time and budget and also knowledge is a double edged sword; it is very crucial to know how much organizational learning is required and how to make the best use of it. A proper KM strategy encompasses organizational, business and IT aspects of a development environment to assist different roles of development team (i.e. Project managers, Business analysts etc.) in carrying out their respective tasks.

#### 4.5 Requirement Change Management

One of the promises of SOA based solutions was the agility of process where changes in requirements can be easily accommodated where services can be easily replaced according to new requirements. Respondents acknowledge that it is easy to accommodate changes in SOSE but it presents various challenges;

- Replacing an existing service with new might degrade the whole performance of system after integration.
- Change of service is very easy in SOSE, but change in requirement may not be that easy especially when requirements and services are not aligned properly in granularity and abstraction.
- A change within one service requires careful impact analysis as this service can be under the use of multiple clients and the change might not be what everyone would wish for. In this case proper versioning of services is required.

**Table 5.** Requirement Change Management

Findings from Literature Survey	Findings from Online Survey	Findings from Interviews	Conclusion
In Service-Oriented Paradigm Requirements Change Management should have been easier than traditional software development but it remains to be challenging.	Changes in service requirements are often as a result of end users not being part of the process early in the project.	Changing services is technically easy in SOSE, but change in requirements may not be that easy especially when requirements and services are not aligned properly in granularity and abstraction. Replacing an existing service with a new one might degrade the whole performance of the system after integration. A thorough impact analysis is required as one service can be under the use of multiple clients. A proper versioning system is required for this purpose.	Change is inevitable in current dynamic environment and one of the main objectives of SOA was to enable the development team to easily accommodate change. But it is still challenging for the development team to decide on how to accommodate a particular change request with considerations to system's business requirements and technical details.

#### 4.6 Additional Challenges

Some additional challenges of SORE were identified by the interviewees.

- The research in SOA is focusing too much on giving technological solutions where as the social aspect is not explored properly under this paradigm.
- Lack of real users during RE and testing phase of software development would result in unsatisfied users that could lead to project failure.
- The traditional RE challenge of ambiguities and incomplete requirements is exacerbated due to the distributed and collaborative nature of development style.

## 5 Discussion

The interviews have not only confirmed most of the issues that were identified as challenging in our online survey, but have also provided more details from real life context. The additional outcome of our interview analysis is that in real life context, the human-factor was found to be more challenging while considering all the issues of RE in SOSE. It is evident that further efforts are required for evaluating the existing practices and providing improved solutions to the social aspect of RE in Service-Oriented paradigm.

With the global marketplace and the focus of IT infrastructure in providing solutions to business needs of organizations, the software industry has adopted the methods that allow distributed, collaborative and dynamic development. SOSE was proposed to deal with a highly dynamic environment for software development by ensuring agility and reusability as central concepts of its philosophy. Over the last ten years improvements have been made as a result of research in technologies and standard protocols for network infrastructure to support distributed development and deployment of services. The technology is important, but how to make best use of that technology by understanding the real objective of the system to build and satisfaction

of those who will ultimately use it, is far more crucial. According to all the interviewees, agile methodology is frequently used in Service-Oriented software development. The objective of adopting agile methodology in SOSE was to make it human-centered, but unless the issues of RE are addressed properly, the results would be in form of conflicts between customers, who are not aware of technology, and the development team, who think only in technical way. This makes the role of RE in SOSE very important. It emphasizes the need for an understanding of the objectives of the system to build, business needs of organizations, full functionality of a service, the context of domain where it will be used and most importantly the group of users who will actually be using that service. Otherwise we might end up with services and service based systems that are useless though technologically very sound.

## 6 Conclusion and Future Work

In this paper, we presented the findings from the interviews we conducted as a part of our exploratory study. The focus of research in this field is largely on technological aspect of service orientation. But equally important is the acceptance of the solutions provided to customer developed under this paradigm. Without customer satisfaction and approval the results would be considered a failure no matter how advance the technology would become. From the analysis a pattern emerged, where the issues of SORE could be seen as belonging to one or more of the following categories;

- *Technical aspect: where the main concerns are related to the tools, techniques, methods and processes that can help in achieving the objectives. Developers take this perspective of requirements during implementation phase.*
- *Organizational aspect: which focuses on policies and decisions that give directions for achieving objectives. The role of the project manager is to deal with this aspect throughout software development lifecycle.*
- *Business aspect: which would focus on the customers/clients and the social and psychological factors that would help them understand the business dynamics for achieving their objective. Business/Requirements analysts and consultants are more focused on this aspect in early phases of development.*

All the issues identified from our analysis could belong to more than one of the three categories. Furthermore the relative importance of each of these categories is dependant on the role played by the practitioners. According to Mitroff and Linstone [18] technical perspective of any issue would be just a single view of a multifaceted problem and currently is used mostly by organizations to deal with various issues. They suggest multiple perspective concept, with two additional perspectives; organizational and individual. Any solution to the problem would turn out to be useless unless all three perspectives are considered in the process. Following the same concept for the three categories that we have identified in our analysis, the interconnectedness of these three concepts suggest that to improve SORE, it's not just the technology that requires attention but at the same time the social and organizational aspects requires equal importance. The highly dynamic IT industry makes continuous learning

essential for the organizations to become successful. The knowledge of organization is an important asset to help them improve themselves. According to the results of all three steps of our research, a KM Strategy seems a promising solution in bringing improvements in RE phase of Service-Oriented software development. It can assist in all three aspects described above to overcome the hurdles that currently prevent SOA from achieving its true goals.

Our next step is to evaluate the impact of organizational KM strategy on Service-Oriented software development and to evaluate its effectiveness in addressing some of the important challenges we have identified in our research.

## References

- [1] Stojanovic, Z., Dahanayake, A.: Service-oriented software system engineering: challenges and practices. Idea Group Publishing (2005)
- [2] van Eck, P.A.T., Wieringa, R.J., Gordijn, J., Janssen, M.: Requirements Engineering for Service-Oriented Computing: A Position Paper. In: First International Workshop on e-Services at ICEC 2003, Pittsburgh, Pennsylvania, USA, pp. 23–28 (2003)
- [3] Emic, C., Cefriel, E.S.I., Eng, A.: Service Centric System Engineering
- [4] Topouzidou, S.: SODIUM, service-oriented development in a unified framework. Final report ISTFP6-004559 (2007), <http://www.atc.gr/sodium>
- [5] Wirsing, M., Hölzl, M. (eds.): SENSORIA. LNCS, vol. 6582. Springer, Heidelberg (2011)
- [6] Arsanjani, A.: Service-oriented modeling and architecture. IBM Developer Works (2004)
- [7] Arsanjani, A.: Service-oriented modeling and architecture: How to identify, specify and realize services for your SOA (2009), <http://www.ibm.com/developerworks>
- [8] Erradi, A., Anand, S., Kulkarni, N.: SOAF: An architectural framework for service definition and realization. In: IEEE International Conference on Services Computing, SCC 2006, pp. 151–158 (2006)
- [9] Papazoglou, M.P., Van Den Heuvel, W.J.: Service-oriented design and development methodology. International Journal of Web Engineering and Technology 2(4), 412–442 (2006)
- [10] Kontogogos, A., Avgeriou, P.: An overview of software engineering approaches to service oriented architectures in various fields. In: 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE 2009, pp. 254–259 (2009)
- [11] Ramollari, E., Dranidis, D., Simons, A.J.H.: A survey of service oriented development methodologies. In: 2nd European Young Researchers Workshop on Service Oriented Computing, Leicester, UK (2007)
- [12] Gu, Q., Lago, P.: Exploring service-oriented system engineering challenges: a systematic literature review. In: Service Oriented Computing and Applications, vol. 3(3), pp. 171–188 (2009)
- [13] Atkinson, C., Bostan, P., Hummel, O., Stoll, D.: A practical approach to web service discovery and retrieval. In: IEEE International Conference on Web Services, ICWS 2007, pp. 241–248 (2007)
- [14] Bano, M., Ikram, N.: Issues and Challenges of Requirement Engineering in Service Oriented Software Development. In: 2010 Fifth International Conference on Software Engineering Advances (ICSEA), pp. 64–69 (2010)

- [15] Bano, M., Ikram, N.: KM-SORE: Knowledge Management for Service Oriented Requirements Engineering. In: The Sixth International Conference on Software Engineering Advances, ICSEA 2011, pp. 494–499 (2011)
- [16] Bano, M., Ikram, N., Niazi, M.: Knowledge Management in Service Oriented Requirements Engineering (KM – SORE), Keele technical Report (TR/2011-02). School of Computing and Mathematics, Keele University, Keele, UK (2011) ISSN: 1353: 7776
- [17] <http://www.gsu.uts.edu.au/policies/hrecpolicy.html>
- [18] Mitroff, I.I., Linstone, H.A.: The unbounded mind: Breaking the chains of traditional business thinking. Oxford University Press, USA (1995)
- [19] <http://www.computing.co.uk/ctg/news/2189825/amazon-loses-customers-web-services-failure>
- [20] Zave, P.: Classification of Research Efforts in Requirements Engineering. *ACM Computing Surveys* 29(4), 315–321 (1997)
- [21] Wiegers, K.E.: In search of excellent requirements. *The Journal of the Quality Assurance Institute* 1 (1995)