KM-SORE: Knowledge Management for Service Oriented Requirements Engineering

Muneera Bano, Naveed Ikram
Department of Software Engineering
International Islamic University Islamabad, Pakistan
muneera@iiu.edu.pk, naveed.ikram@iiu.edu.pk

***Abstract*—Service-oriented Software Engineering is a new style for creating software using reusable services which are available over the web. The biggest challenge in this process is to discover and select the appropriate services that match system requirements. Currently, none of the proposed approach has been accepted by research community as a standard. There is very little empirical work available that addresses requirements engineering in service oriented paradigm. The aim of this study is to propose a framework for requirements engineering in SOSE. The framework is based on a new idea, that integrating Knowledge Management in Service Oriented development would improve requirement engineering phase as it does for traditional software engineering. The framework is developed in the light of the issues and challenges identified by published literature and the feedback of practitioners and researchers working on service oriented projects.**

Keywords- Service Oriented Software Engineering (SOSE); Requirement Engineering (RE); Knowledge Management (KM).

I. INTRODUCTION

Reusability of software is a major concern for software engineers. In current market conditions meeting deadlines and producing quality software is vital. In these conditions recoding what has already been coded in a good quality is wasting your time. Component Based Software Development (CBSD) was the result of the efforts by research community in software engineering for providing methods and techniques for effective, faster and economical software development by ensuring the reuse of existing software modules. Along with the benefits these solutions posed some new challenges for developers. CBSD though proved promising for software reuse and maintainability but it still faces issues like heterogeneity of platforms and protocols, and difficulty of locating required components and selecting them against system requirements [1]. Another effort to overcome these issues is the new paradigm of Service Oriented Software Engineering [2], which is a new architectural style for building applications that support loose coupling among web services. The basic building block of software in SOSE is a web service, which is accessible via internet. Web service is a ready to use software and can be accessed via interface or API over internet by using XML standard messaging format of Web Service Definition Language (WSDL). The service provider publishes specification of service in central repository Universal Description, Discovery and Integration (UDDI), which is explored by service requester. When a service is selected requester and provider make Service Level Agreement (SLA). For the last few years the number of services on the web has increased exponentially. Discovering appropriate web service which satisfies the requirements of the requester has become a challenge. In SOSE services are in ready to use state so the focus in this case is on identifying the services that accurately or at least appropriately fulfil system requirements. Requirements Engineering (RE) for SOSE can have different traditional development activities such as modelling, specification, and analysis but RE processes are carried out in different way [3]. The RE revolves around making a match between ready to use software components and user requirements and the result should be a compromise agreed upon by all stakeholders.

In this paper, we propose a framework for requirements engineering in SOSE. The framework is formulated on the results obtained from published literature and opinions of practitioners and researchers working on service-oriented projects.

Section II describes motivation for our research. Section III is prior related work done on SOSE and the gap analysis of it. Section IV gives details of proposed solution and Section V describes the proposed framework KM-SORE. Section VI outlines future work followed by references and appendix.

II. MOTIVATION

The additional task the requirement engineer has to perform in SOSE is to gain knowledge by exploring existing services with the aim of matchmaking between requirements and available services [4]. Service-oriented Software Engineering (SOSE) is a new field and is not fully mature yet. Though, recently a lot of interest has been shown by both industry and academia researchers [2][5][6][7][8][9], but there are still problems and challenges in this field [10][11]. Broadly, these challenges can fall into four categories that relate to main four phases of SOSE [12];

A. Specification (Planning) Issues

By specification issues we mean, the problems that are faced when we want to know the requirements for system, and during planning of acquiring these requirements and making them complete.

B. Discovery Issues

This category deals with searching for the services that actually meet the functional and non functional requirements.

C. *Composition Issues*

Services are selected based on their individual functionalities. Next, we need to see if they will work properly in a workflow by making composition that satisfies system requirements.

D. *Management Issues*

If a requirement is changed, or a new version of service is launched, or the service becomes unavailable due to any reason, re-composition and re-deployment of system is required.

The issues identified in literature against above mentioned SOSE phases are summarized in the following list, and details can be found in [12];

- *Web Service Discovery*
    - o *Matching user requirements and available service*
    - o *Automated and Dynamic Service Discovery*
    - o *Iterative Discovery Process*
    - o *Completing requirements with discovery*
    - o *High Level Language/Tool Support*
- *Innovation and creativity in RE*
- *Requirement change and evolution*
- *Semantic gaps in specification*
- *Knowledge Management in service-oriented SDLC*
- *Non functional Requirements gathering and assessment*
- *Web service Dependency discovery*
- *Platform dependence in selected web services*
- *Lack of standard RE process for SOSE*

With these challenges and issues on one side and the promises proposed for systems developed in service-oriented paradigm (cost effective, reduce time effort, re-usability, agility, platform independence, loose coupling etc.) there is a need of systematic investigation of the real nature of these problems and what solution can be proposed to overcome them. In Systematic Review conducted to explore SOSE challenges [13] it was found that total 8 challenges of RE were worked upon by researchers in year 2007 and 2008. The empirical work is not sufficient in this field. There is a need of new RE process [3], which should consider only the service-oriented paradigm of software development life cycle [1][14]. There has been no standard accepted so far for RE process in this domain [10][11]. This motivates further exploration and research in this emerging area.

III. RELATED PRIOR WORK

SOSE is currently under focus of research community from different perspectives. There have been many methods, techniques and tools proposed by different mega projects and research teams. They include; SeCSE [6], SODIUM [8][15], SENSORIA [7], IBM SOA [9][16], MICROSOFT [17], SOAF [18], SDLM [19], and work other researchers [20] [3].These approaches are proposing solutions in their own ways and are not sharing any common ground [10][11]. That is not going to be helpful in providing any unified approach in future as it happened with RUP and UML [10]. To provide unification there should be more standards, models and patterns proposed for this new field [10]. Some of the proposed approaches still lack validation from industrial feedback. According to the systematic review of Qing Gu et al [13], empirical work related to requirement engineering in SOSE is not sufficient. There is a need for further empirical work in this area [10], with real life projects to provide feedback for improvements in current methods and practices and also to enrich the knowledge in SOA domain and open further research directions.

Table IV in Appendix section shows comparison and gap analysis of existing technologies with respect to the phases of SOSE, along with associated issues reported in literature. Table V in Appendix section shows comparison for gap analysis of work done to the issues identified in section “Motivation”.

IV. PROPOSED SOLUTION

Knowledge Management has been proved helpful in core activities of traditional software engineering [21][22][23]. Considering SOSE as a sub field of software engineering, we have deduced that knowledge management process if integrated into all activities of SOSE, would improve the RE process and would be helpful in tackling the issues of RE in SOSE. Not equivocally, but the idea has been supported by some of the researchers in different ways. In [24], KMP is considered important to accomplish tasks of Business Process Management (BPM) in Service-oriented Architecture (SOA). In [25], the authors have highlighted the need for novel approach for sharing service knowledge and application specific information. Knowledge is required to build trust among distributed parties on heterogeneous platforms, when we do automated composition [26]. Knowledge management can improve cooperative work among services [27][28]. XML messaging data if managed can provide information regarding web service dependency by the calls one service make to the others [29].

The philosophy of service orientation is built on the idea of software reusability and agility of the process. The designers and developers must know what solutions are available in order to develop right system, with correct process. The required knowledge is about available services, previous decisions and their results, constraints of using any tool/technique/method, service interdependencies etc.

If we summarize the whole discussion, we can conclude,

- *SOSE is a shift of paradigm from SE*
- *Traditional RE cannot be applied to SOSE*
- *SOSE is facing challenges in RE*
- *KM has been proved promising in SE*
- *KM can improve RE in SOSE*

Therefore, we propose that if Knowledge Management is used in Service-oriented Software Engineering it would help in overcoming most of the issues of Requirements Engineering. We formulate the proposed framework on the basis of issues and challenges of RE in SOSE highlighted in published literature and then conducting a survey to get the opinion of practitioners and researchers working on service-oriented projects about the issues and impact of KM on SOSE. The first part was to conduct a literature survey, and the second part was a questionnaire-based web survey.

A. *STEP1: Literature Survey*

The aim of performing literature survey was to extract the list of issues and challenges of requirements engineering in SOSE that has been reported in published literature. The results from this phase are published [12]. The factors identified from this phase are listed in motivation section.

B. *STEP2: Survey*

The purpose of conducting the survey was to validate the list of issues of RE in SOSE, extracted from published literature, from practitioners working on service-oriented system development and to get their opinion on using KM in SOSE. The population comprised of those people who have worked on service-oriented projects either as technical team member or as a researcher. 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 level. We administered the survey on web and sent the link through email to invite the practitioners around the world. The duration for the survey was from 16th December 2010 to 23rd January 2011. A total of 117 responses were received from all around the world in this duration with almost 5.2% response rate. 20% of the responses were received from USA and 17% from India. 60% of the respondents have experience in relevant area between 4 to 9 years. 42% of the respondents are SOA architects. Out of 117 respondents 100 had experience as a practitioner and 77 had worked as a researcher in SOSE. After analysis 8 responses were rejected for providing incomplete information. And out of 109 responses, 9 were only researchers, 32 were only practitioners and 68 had experience both as a researcher and practitioner. The ranking of the factors was analyzed by grouping respondents into above mentioned three categories; only Practitioners (P), only Researchers (R) and experience of both practitioner and researcher (P+R). The ranking in Table I has been calculated for the factors on the basis of average agreement level they have achieved in their question items. The ranking provides an interesting overview of what is important for each group of respondents as they all have a different ranking for the measurement factors. The difference in opinion is mostly because SOSE is a new field, and a shift of paradigm from traditional software engineering. Most of its concepts are not fully mature yet and they are not fully understood and appreciated by designers and developers, resulting in a poor implementation of the SOSE concepts. This according to the respondents is one of the reasons for the resulting issues besides other. Knowledge Management in Service-oriented Software Engineering has got highest agreement level in overall ranking in above table.

Measurement Factors	**Agreement Percentage (Stronly agree+agree)**			
	All (109)	**Only P (32)**	**Only R (9)**	**P+R (68)**
Knowledge Management in Service-oriented Software Engineering	76	78	60	76
Matching user requirements and available service	75	73	73	77
Iterative Discovery Process	74	60	89	80
Semantic gaps in specifications	53	63	63	54
Automated and Dynamic Service Discovery	57	60	59	56
High Level Language Support	35	37	41	32
Eliciting requirements through service discovery	72	72	89	71
Service Testing	61	75	55	35
Requirement change and evolution	48	50	33	49
Innovation and creativity in RE	67	78	66	62
Non functional Requirements gathering and assessment	59	59	67	59
Lack of standard RE process for SOS	54	41	44	62

TABLE I. RANKING OF ISSUES AND CHALLENGES FROM SURVEY

It has been reported in literature that KM (Knowledge Management) improves the software development process. In traditional software engineering, knowledge management helps in; Decreasing development time and cost, and increasing quality, Making better decisions, Understanding the domain, Communication, Acquiring knowledge about new technologies, Accessing domain knowledge, Sharing knowledge about local policies and practices, Capturing knowledge and knowing who knows what, Collaborating and sharing knowledge [21]. In core SE activities KM can support in; Document management, Competence management, Expert identification, Software reuse (making developers aware of existing software contents/components) [21]. Similarly, if knowledge management is applied in SOSE, along with above mentioned benefits, it would help to increase understanding of the engineers and would address the issues that arise due to misunderstanding of this style of solution making. Overall the survey results have indicated that KM would have a good impact on SOSE life cycle. According to comments of respondents it would help to solve following issues of SOSE; *matchmaking between requirements and services, iterative discovery process, decision making, semantic gap, eliciting requirements through service discovery, re-composition, automating the discovery process.*

Naturally, SOSE would face fewer challenges in integrating KM then tradition software development. The central repository UDDI contains all information about specification of available services. The conversion method of requirements into formal queries, search process, retrieved results etc. could all be codified and stored. Such that, if we face a problem the previously stored knowledge can be retrieved and analyzed. Integrating creativity and innovation in requirements engineering for SOSE would help in making new ways for solutions and maintaining a KM along with the process would help in not repeating the mistakes. KM would definitely take cost and efforts for its implementation as apprehended by respondents, but it is like an investment for improvement in the process where the benefits become visible with time. The respondents had a concern that whether there would be a need of some specific to field KMP for SOSE. Current KMPs for SE were not proposed considering service-oriented paradigm. How a KM for traditional SE can be adapted for SOSE, is yet an area of exploration. According to the systematic literature review conducted by Bjørnson and Dingsøyr on knowledge management in software engineering [22], the main emphasize of SE has been so far on technocratic category of KM in Earl’s taxonomy for Knowledge management strategies [30]. The KM process for SOSE will mainly cover technocratic and organizational sub category from behavioral of Earl’s Taxonomy for KM.

V. FRAMEWORK KM-SORE

The framework has integrated KM with SOSE life cycle in the light of findings from literature and survey. After analyzing the issues of requirements engineering in SOSE and results from the survey we found that the issues are somewhat interlinked because the phases of SOSE work in iterations. RE in SOSE is not a discrete activity but is related to the Discovery and Composition phase as well. There for any problem cannot be described as belonging to one phase. The issues of RE are also overlapping into other phases as well. All the phases of SOSE depend on each other for their functioning. Table II shows their overlapping in different phases of SOSE [12].

The framework proposes to integrate KM in all the phases of SOSE. These phases are interrelated and work in iteration for successful composition. J. Ward and A. Aurum [31] have given a refined list of KM process activities from literature. They have given list of seven activities. On highest level of abstraction three activities are required for KMP; Knowledge Creation (KC), Knowledge Storage (KS), Knowledge Retrieval (KR). Integration of KM in SOSE will require Knowledge creation, storage and retrieval in all four main phases. Table III summarize the overall idea of this integration.

In a common storage space, the codified knowledge from all phases will be stored to make it available for all phases, ultimately storing the information in organization’s central knowledge base. Figure 1 shows this integration graphically. The framework provides an overall view of how KM would be integrated in SOSE phases. Exactly what strategy and tools have to be selected for knowledge management, is to be decided by the organization. The literature has guidance available on how to select an appropriate strategy [32] [33] [34] and software tool [35] [36].

SOSE Phase	Related Issues
Planning	Matching user requirements and available service Following iterative Discovery Process Completing requirements with iterative discovery Semantic gaps in specification Non functional Requirements gathering and assessment
Discovery	Matching user requirements and available service Following automated and dynamic Service Discovery Following iterative Discovery Process Completing requirements with iterative discovery Semantic gaps in specification Non functional Requirements gathering and assessment
Composition	Semantic gaps in specification Web service Dependency discovery
Management	Dealing with requirement change and evolution

TABLE II. ISSUES AND THEIR RELATION TO PHASES OF SOSE

SOSE PHASE	KMP ACTIVITY		Related issues
Planning	KC	Initial composition design	Matching user requirements and available service Ease of iterative Discovery Process Completing requirements with iterative discovery Semantic gaps in specification Non functional Requirements gathering and assessment
	KS	Codifying created knowledge	
	KR	User requirements, results of discovery	
Discovery	KC	Queries and results of discovery	Matching user requirements and available service Ease of automated and dynamic Service Discovery Ease of iterative Discovery Process Completing requirements with iterative discovery Semantic gaps in specification Non functional Requirements gathering and assessment
	KS	Codifying created knowledge	
	KR	Initial composition design, user requirements	
Composition	KC	Workflow for composition	Semantic gaps in specification Web service Dependency discovery
	KS	Codifying created knowledge	
	KR	Results of discovery, user requirements	
Management	KC	Changes in workflow of composition and their reasons	Ease of requirement change and evolution
	KS	Codifying created knowledge	
	KR	User requirements, workflow of previous composition, results from previous discovery	

TABLE III. KNOWLEDGE MANAGEMENT IN SOSE

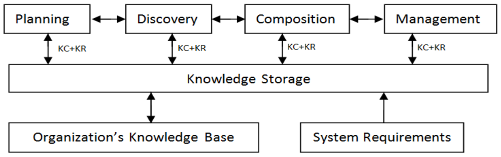

Figure 1. KM Activities integrated with SOSE phases

VI. Conclusion and Future Works

According to the results of our survey, KM has been acknowledged by practitioners to improve the issues and challenges of RE in SOSE. Currently, the framework has its foundations on the basis of results obtained from literature and survey. Our next task is to apply the framework in experimental setup to see the results it would produce. We will be conducting an experiment on two projects of SOSE one with proposed framework and one without it. The results from both projects will be compared. It will be an observational field experiment, where we will be evaluating the checklist of issues of RE in SOSE in both projects to see any difference in the results. Improvement will be assessed based on the checklist of issues of RE in SOSE, mainly based on time and ease of performing the task along with the phases of SOSE. The data collection units during observations will be: Accuracy rate of discovery results, Time of decision for selection of query, Composition success/failure rate, Time for accommodating service change request.

References

[1] H. P. Breivold and M. Larsson, “Component-Based and Service-Oriented Software Engineering: Key Concepts and Principles,” in *33rd EUROMICRO Conference on Software Engineering and Advanced Applications, 2007.* pp. 13-20.

[2] W. T. Tsai, “Service-oriented system engineering: a new paradigm,” in *IEEE international workshop on service-oriented system engineering (SOSE), Beijing*, 2005, pp. 3-8.

[3] W. T. Tsai, Z. Jin, P. Wang, and B. Wu, “Requirement engineering in service-oriented system engineering,” in *proceedings of International Workshop on Service-Oriented System Engineering*, pp. 661–668.

[4] G. Spanoudakis, A. Zisman, and A. Kozlenkov, “A service discovery framework for service centric systems,” in *Proceedings of the IEEE International Conference on Services Computing (SCC’05)*, 2005, pp. 251–259.

[5] A. Arsanjani, “Service-oriented modeling and architecture,” *IBM developer works*, 2004.

[6] S. Crew, *Service Centric System Engineering—EU/IST Integrated Project*. 2004.

[7] M. Wirsing and M. Hölzl, *Rigorous Software Engineering for Service-Oriented Systems—Results of the Sensoria project on Software Engineering for Service-Oriented Computing*. Springer, 2010.

[8] S. Topouzidou, “SODIUM, Service-Oriented Development In a Unified framework,” *Final report ISTFP6-004559. http://www. atc. gr/sodium*.

[9] A. Arsanjani, L. J. Zhang, M. Ellis, A. Allam, and K. Channabasavaiah, “S3: A service-oriented reference architecture,” *IT professional*, vol. 9, no. 3, pp. 10-17, 2007.

[10] A. Kontogogos and P. Avgeriou, “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, 2009. WETICE ’09.* 2009, pp. 254-259.

[11] E. Ramollari, D. Dranidis, and A. J. H. Simons, “A survey of service oriented development methodologies,” in *The 2nd European Young Researchers Workshop on Service Oriented Computing*, 2007.

[12] M. Bano and N. Ikram, “Issues and Challenges of Requirement Engineering in Service Oriented Software Development,” in *Fifth International Conference on Software Engineering Advances (ICSEA), 2010*, 2010, pp. 64-69.

[13] Q. Gu and P. Lago, “Exploring service-oriented system engineering challenges: a systematic literature review,” *Service Oriented Computing and Applications*, vol. 3, no. 3, pp. 171-188, 2009.

[14] A. Brown, S. Johnston, and K. Kelly, “Using service-oriented architecture and component-based development to build web service applications,” *interactions*, vol. 1, pp. 2-4.

[15] S. Topouzidou, *SODIUM, Service-Oriented Development In a Unified framework, Final report ISTFP 6-004559.* .

[16] A. Arsanjani, “How to identify, specify, and realize services for your SOA,” *IBM Corporation, February*, 2005.

[17] A. Sehmi and B. Schwegler, “Service-oriented modeling for connected systems-part 1,” *The Architecture Journal*, vol. 7, pp. 33-41, 2006.

[18] A. Erradi, S. Anand, and N. Kulkarni, “SOAF: An architectural framework for service definition and realization,” in *SCC’06. IEEE International Conference on Services Computing,* 2006, pp. 151-158.

[19] M. P. Papazoglou and W. J. Van Den Heuvel, “Service-oriented design and development methodology,” *International Journal of Web Engineering and Technology*, vol. 2, no. 4, pp. 412-442, 2006.

[20] W. T. Tsai, “Service-oriented system engineering: a new paradigm,” in *IEEE international workshop on service-oriented system engineering (SOSE), Beijing*, 2005, pp. 3-8.

[21] I. Rus and M. Lindvall, “Knowledge management in software engineering,” *IEEE software*, pp. 26-38, 2002.

[22] F. O. Bjørnson and T. Dingsøyr, “Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used,” *Information and Software Technology*, vol. 50, no. 11, pp. 1055-1068, 2008.

[23] A. Aurum, F. Daneshgar, and J. Ward, “Investigating Knowledge Management practices in software development organisations–An Australian experience,” *Information and Software Technology*, vol. 50, no. 6, pp. 511-533, 2008.

[24] Jianmei Guo, Yinglin Wang, Xijuan Liu, and Hongxia Tong, “A Service-oriented Integration Model of Knowledge and Business Processes,” in *IEEE International Conference on Networking, Sensing and Control. ICNSC,* 2008, pp. 1674-1679.

[25] D. Huang, Y. Yang, and J. Calmet, “A Knowledge-based Security Policy Framework for Business Process Management,” in *Proceedings of the International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce*, 2006, pp. 154-160.

[26] G. Koumoutsos and K. Thramboulidis, “A knowledge-based framework for complex, proactive and service-oriented e-negotiation systems,” *Electronic Commerce Research*, vol. 9, no. 4, pp. 317-349, 2009.

[27] H. Kojima, K. Funaki, and T. Inoue, “Web Service Systems for Cooperative Work Support in Knowledge Creation Processes,” *Human Interface and the Management of Information. Designing Information Environments*, pp. 94-103.

[28] Jiang Jichen and Gao Ming, “A process-driven content-oriented integration framework for knowledge management systems,” in *SOLI ’09. IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics,* 2009, pp. 213-218.

[29] S. Basu, F. Casati, and F. Daniel, “Web service dependency discovery tool for SOA management,” in *IEEE International Conference on Services Computing, 2007. SCC 2007,* 2007, pp. 684-685.

[30] M. Earl, “Knowledge management strategies: toward a taxonomy,” *Journal of Management Information Systems*, vol. 18, no. 1, pp. 215-233, 2001.

[31] J. Ward and A. Aurum, “Knowledge management in software engineering-describing the process,” in *Software Engineering Conference, 2004. Proceedings. 2004 Australian*, 2004, pp. 137-146.

[32] K. Haggie and J. Kingston, “Choosing your knowledge management strategy,” *Journal of Knowledge Management Practice*, vol. 4, pp. 1-23, 2003.

[33] J. Robertson, “Developing a knowledge management strategy,” *Knowledge Management Column*, 2004.

[34] B. Choi and H. Lee, “Knowledge management strategy and its link to knowledge creation process,” *Expert Systems with applications*, vol. 23, no. 3, pp. 173-187, 2002.

[35] M. Lindvall, I. Rus, and S. S. Sinha, “Software systems support for knowledge management,” *Journal of Knowledge Management*, vol. 7, no. 5, pp. 137-150, 2003.

[36] M. Lindvall, I. Rus, and S. S. Sinha, “Technology support for knowledge management,” *Advances in Learning Software Organizations*, pp. 94-103, 2003.

APPENDIX

Related Prior Work	**Specification**	**Discovery**	**Composition**	**Management**	**Issues** [10][11]
SeCSE (Analysis and Design)	Tools and methods for testing service specification and quality Early Service Discovery (ESD) to complete requirements	Supporting framework for runtime service discovery, and search engine	QoS aware service composition in order to solve re planning problem Architecture-based Service Discovery (ASD)	Self healing service composition, requirement monitoring Run Time Service Discovery (RTSD)	Lack of Industrial case studies Unable to handle heterogeneous service composition Covers analysis and design of SDLC only Only workflow technique is used (not supporting semantic web techniques)
SODIUM (Complete Composition Life Cycle)		XML based Query language (USQL) search engine for USQL, Languages for Unified Discovery and Composition	Heterogeneous service composition USCL (Unified Service Composition Language) A Methodology for Service Composition VSCL (Visual Service Composition Language)		The project is focused mainly on providing methods/tools for composition of heterogeneous web services. There focus is different and is not from requirement engineer’s perspective.
SENSORIA (Complete SDLC)	UML Profile for Service-Oriented Systems(UML4SOA) Ontology for SOAs, SENSORIA Reference Modelling Language (SRML) Prototype Language for Business Policies		Mathematical models for simulation and verification of service composition		Not evaluated by researchers yet.
IBM		Service identification method, Service classification and categorization method	Service realization method		Covers analysis and design of SDLC only SOMA lacks openly available detailed description of the methodology, which makes it difficult to further analyze its capabilities.
SOAF	Information elicitation	Service identification and service definition	Service realization	roadmap and planning (management)	It has given the framework for whole life cycle about what to do but details on guidelines of how to do are missing. Lack of tool support.
Papazoglou et al Framework SDLM	Planning and Analysis (Using methods of BPM, RUP, CBD)		Service construction, testing, provisioning, deployment	Monitoring	The proposed approach does not provide enough guidelines for explicit consideration of service model artifacts.

TABLE IV. COMPARISON AND GAP ANALYSIS OF EXISTING TECHNIQUES WITH PHASES SOSE AND ASSOCIATED ISSUES

Issues from Literature	**SeCSE**	**SENSORIA**	**SODIUM**	**IBM**	**SOAF**	**SDLM**
Matching user requirements and available service	Yes	No	No	Yes	Yes	Yes
Automated and Dynamic Service Discovery	Yes	No	No	No	No	No
Iterative Discovery Process	Yes	No	No	No	No	No
Completing requirements with discovery	Yes	No	No	No	No	No
High Level Language Support	Yes	Yes	Yes	No	No	Yes
Innovation and creativity in RE	Yes	No	No	No	No	No
Requirement change and evolution	Yes	Yes	No	Yes	Yes	Yes
Semantic gaps in specification	No	Yes	No	No	Yes	Yes
Knowledge Management in service-oriented SDLC	No	No	No	No	No	No
Non functional Requirements gathering and assessment	Yes	No	No	Yes	No	No
Web service Dependency discovery	No	No	No	No	No	No

TABLE V. COMPARISON AND GAP ANALYSIS OF EXISTING TECHNIQUES TO THE ISSUES OF RE IN SOSE REPORTED IN LITERATURE