Abstract— Service Oriented Architecture (SOA) is a shift of paradigm in software development. It can be seen as an evolution of Component Based Software Development (CBSD), with web services used instead of Commercial Off-the-shelf (COTS) software. For the last few years the number of services on the web has increased exponentially. Among available services locating the best service that fulfills the user requirement is a challenging task for researchers. There is still no standard Requirement Engineering (RE) process defined for Service Oriented Software Development (SOSD). The traditional processes and those used for COTS selection cannot be used due to the architectural differences of SOSD with other domains. In this paper we have extracted a list of issues and challenges from literature under considerations by research community for RE process in SOSD. The issues of RE in CBSD are compared with those of SOSD, as CBSD is considered close in nature to SOSD. The results shows that there is a need of standard RE process for SOSD with proper guidance on how to perform different steps with details.

Keywords - requirement engineering; service oriented software development; component based software development; commercial off-the-shelf softwares (COTS).

I. INTRODUCTION

Component based software development was a shift of paradigm from traditional software development to facilitate the software development in effective, faster and economical way by ensuring the reuse of software packages known as component or COTS. Component is packaged software available in ready for use state. It is assumed to have gone through all phases of software development in standard manner. It can be leased, purchased or licensed to the users with the objective to reduce cost, and time for developing software rather than from scratch. But this process increases efforts for software process integration and dependence on vendors [1]. Further efforts are required for tradeoff between requirements and design depending on what components are available [2]. 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 [3].

The internet has an important role in the field of computing by providing appropriate infrastructure and technologies for distributed application development where developers are provided with transparency to the underlying working technologies. Therefore they can focus on designing application by locating the required components in distributed environment. This view takes us to the new paradigm of Service Oriented Software Development [4]. The software in this paradigm is presented as a web service, known by the term SaaS (Software as a Service) [5]. A web service is a software components available on the internet with web interfaces, accessed remotely via standard message format such as Web Service Definition Language (WSDL) [6] using internet protocols and infrastructure. Like component it can be leased to be used as a part of software. SOSD can be seen as evolution of CBSD [3] by providing the new architectural style for building applications that support loose coupling among web services. These services can be located in central repository Universal Description, Discovery and Integration (UDDI) [7][8] and after Service Level Agreement (SLA) [9] can be used by composing an application.

Components and web services hold many similarities like; search and selection of appropriate component that meets system requirements [10][11], components or services in ready to use state and have to be leased or purchased from third party, need of an iterative process to satisfy system requirements, evaluation of available components or services for selection against quality criteria or non functional requirements (NFR), reuse in multiple applications, and both separate the development of system from component or service development [3]. But there are differences, as web services have web interface or API and they can be located in a central repository (UDDI) so searching them is not an issue as it is in CBSD.

SOSD is now the emerging paradigm that seems to provide solution to the problems faced in CBSD so as to make application development easy, low cost even in heterogeneous distributed environment. Software is considered successful if it meets the objective for which it was made and to identify that objective we use RE process [12]. The process includes; Identification of stakeholders, gathering their needs and understanding them, documenting the specifications and analyzing it for subsequent implementation [13]. RE can be seen as made up of two phases:

- Requirement Development; it includes activities like, Requirement Elicitation, Analysis, Specification and Verification
- Requirement Management; it deals with managing and controlling changing and evolving requirements

In traditional RE process, the main concern is acquiring and finalizing the system specifications when you have to build a system from scratch. But in CBSD and
SOSD components and services are present out there in ready to use state so the RE process has to focus in this case on which of the components accurately or at least appropriately fulfill system requirements.

The aim of this paper is to highlight the challenges in research literature faced by service oriented software development in requirement engineering phase. By comparing it with issues of RE in CBSD, we analyzed the results to conclude about what is desirable for RE in SOSD. The results from literature show the directions in which the research community is putting efforts. It would provide guidelines to new researchers towards a new RE process for SOSD comparing with the work done for CBSD.

The paper is organized as follows; Section II shows motivation for the topic by highlighting the need for new RE process for SOSD. Section III shows work done in the area regarding CBSD and SOSD. Section IV gives list of challenges and issues presented in literature for RE in traditional Software Development, CBSD and SOSD. Section V is discussion about those issues and section VI gives conclusion.

II. Motivation

As Service-based Software Development is a shift from traditional development paradigm, several new methods have been proposed for it [4][14][15][16][17]. RE phase is affected by the Software Development Life Cycle (SDLC) used for development [12]. For example in traditional waterfall model, major effort in RE phase is to acquire, specify and analyze the requirements during analysis phase. But in case of agile process, requirements are taken only in informal scenarios as the main focus here is on coding phase. So the way in which the requirements are gathered, specified and used is different in various development life cycles [18]. RE for SOSD can involve activities of traditional process like modeling, specification, and analysis but the way in which these activities are carried out is different [18]. The main focus here is to identify the services that match the system requirements and then modeling a composition on the basis of selected services. SOSD is a different architectural style. Therefore, it requires a complete new RE process [18], which should consider only the service-oriented paradigm of software development life cycle [3][4]. There has been no standard accepted so far for RE process in this domain.

III. Work done in area

Many solutions are proposed in literature regarding different problems highlighted by researchers. We consider the case of CBSD first because these processes are providing guidelines to the researchers of SOSD.

A. RE processes for CBSD

There are different processes proposed in literature for requirement phase in CBSD. Some of them are;

- Off The Shelf Option method (OTSO) [23]: It defines hierarchical evaluation criteria and identifies four different sub processes. It highlights importance of requirements acquisition for COTS selection but gives no solution to it.
- Procurement Oriented Requirement Engineering (PORE) [24]: It is an iterative and template based approach for off-the-shelf evaluation and selection. In PORE it is not clear how requirements are specified in evaluation process and how products are eliminated.
- COTS Aware Requirement Engineering (CARE) [25]: The CARE approach works upon ideas available in current RE methodologies including RUP [26], MBASE and ACRE/PORE. The aim is to enhance these methodologies.
- COTS-based Requirement Engineering (CRE) [27]: This process incorporates quality requirements assessment as a way for making COTS selection right.

These above mentioned methods rely on pre-structured and established criteria based on fixed requirements which are not an appropriate approach in uncertain market conditions [28].

B. RE Processes for SOSD

SOSD is currently under consideration of research community from different perspectives. There have been many methods, techniques and tools proposed by different mega projects and research teams.

SeCSE [15]: Service Centric System Engineering project had the aim to create free and open source methods, tools and techniques for system integrators and service providers to support the cost effective development and use of dependable services and service centric applications. They have incorporated RE with three phases of service discovery that is Early Service Discovery, Design time Discovery and Run Time Discovery.

SENSORIA [16]: “Software Engineering for Service Oriented Overlay Computers” had the aim to develop a new approach for service oriented software engineering with foundation theories, techniques and methods. Their focus is on whole SDLC in Service oriented paradigm, from requirements to deployment including re-engineering of legacy systems. The proposed methods and tools make use of mathematical theories and models for ensuring correctness of the procedure and allowing a semi-automatic design process.

IBM SOA [29][30]: IBM has proposed Service Oriented Modeling and Architecture (SOMA) [31][14] which is an end-to-end software development method for building SOA based solutions. They provide guidance on identification of services just like objects are identified in Object Oriented Software Development.

SORE workshop [61]: Service Oriented Requirement Engineering workshops were held in 2004 and 2007. Their aim was to gather the research community and share the
ideas, knowledge and work on requirement engineering for service oriented systems.

**Other Researchers:** In [4], Tsai et al. have discussed Service Oriented Software Engineering. In [18], they have focused on RE phase of SOSE and proposed a SORE framework. Xiang et al. have proposed Service Requirement Elicitation Mechanism (SREM), which is based on Service Requirements Modeling Ontology (SRMO), which has taken basic concepts from agent-oriented modeling framework i* (i star) [32]. Value based software development concepts are used in Value Gap Model for eliciting requirements for service components [62].

IV. CHALLENGES HIGHLIGHTED IN LITERATURE

With considerations to the similarities of CBSD and SOSD [7], it is evident that we will have to look into the issues of RE that are faced by CBSD, to evaluate if they are inherited to the SOSD. To find out the challenges on which the researchers are focusing from year 2000 onwards; we conducted an exhaustive literature review and extracted issues from them until they started repeating. Traditionally RE faced challenges like;

- Issues regarding stakeholders [33][34][35]
- Capturing, modeling and analyzing Non Functional Requirements [35]
- Reuse of requirement models [12]
- Formal representation of requirements from natural language [36]
- Requirement change and evolution [12]
- Conflict resolution in requirements [34]
- Creating requirements [20]

These challenges are inherited during the development of component or a web service itself, with wider market in consideration. But when a system based on these ready to use component/service is to be developed, new issues are introduced which are discussed in following sections.

A. RE in Component based Software Development

CBSD life cycle is different from traditional SDLC. CBSD life cycle comprises of; requirement analysis, software architecture selection, construction, analysis and evaluation, components identification, selection and customization, system integration, testing and maintenance [37][38]. Besides the issues of traditional RE in CBSD, new issues have been introduced due to the shift in paradigm [39], they are as follows;

- **CB1:** No standard RE process exists for CBSD, specially for searching and selection of components [37][24]
- **CB2:** NFR can play an important role in Quality comparison among multiple components providing same functionality. As components have black box nature they can be only tested against user criteria for quality [37][38][40][41][42]. Systematic evaluation and testing of components is required against user requirements [41].
- **CB3:** Specifications of existing components are also to be considered when building new system’s requirements. Traditional approaches cannot be applied here [40]. Tradeoff is required between user requirements and component selection, so a flexible and iterative RE technique is required for making decisions for component selection [40][41][42][43].
- **CB4:** Search and selection process is the most important phase in CBSD life cycle [40].
- **CB5:** Incompatible components can fail the system at the time of integration [42].
- **CB6:** As components are black box in their nature [40][41] the source code is not provided and they are inflexible for customization [42].
- **CB7:** Versioning in components can cause problems as new version may not match the existing system requirements [42].
- **CB8:** The process should be an iterative and should be able to do knowledge sharing which is supply chain of components/products, skills and experiences and personnel [24][25].
- **CB9:** COTS selections and RE should be performed in parallel as requirements and design both depends on available components [2].

B. RE in Service Oriented Software Development

The issues in the literature on which the researchers focus are to some extent inherited from CBSD along with those issues that were present in traditional software development. But some are specific to SOSD. Based on the comparison with CBSD we have gathered following list of issues, which should be considered in an RE process for SOSD;

- **SD1:** Service Discovery is the most important phase so effective mechanisms are required to locate correct service according to user requirements [44]. It’s comparable to issue CB4 but searching is not a big issue due to central repository UDDI but finding all required services is a challenge.
- **SD2:** SOSD should have automated dynamic service discovery based on user requirements specifications [44], and high level language support for requirement specification specially in automated process [45][46].
- **SD3:** An iterative discovery process is required, to refine the requirement specifications [47][48][49]. It is comparable to issue CB8.
- **SD4:** Web service discovery can be used as a process to complete the requirements [47][48][49][50].
- **SD5:** Innovation and creativity in RE should be the part of SOSD [20][21][22].
- **SD6:** SOSD should be able to redesign and redeploy the composed service when user needs change over time [45].
- **SD7**: The process should bridge the semantic gaps which are inevitable when services are brought together from hybrid environments [45].
- **SD8**: It is required by the process to manage the knowledge of group/cluster of services with similar functionality.
- **SD9**: NFR should be used for evaluating and selection among multiple available candidate services [18][51][52][53][54][55] as compared to issue CB2.
- **SD10**: Service composition should be able to give orchestration mechanism such as to avoid deadlocks in service invocation.
- **SD11**: Web service dependencies should be discovered from the messages by which they communicate with each other [56].
- **Issue CB 5,6,7,9** would be inherited with considerations of similarities of web services and COTS.

V. **DISCUSSION ON ISSUES HIGHLIGHTED**

A. **RE perspectives in SOSD**

Considering the web service architecture [57][58] we take two perspectives for RE; service provider and service requester. We compare this again to the CBSD paradigm, where we had component providers and component integrators [59].

1) **Service Provider**

From this perspective, the requirements are to be elicited from market demands. The nature of requirements in this case would be highly volatile and would change quickly. As market demand is a major factor in causing requirements to change [60]. The service provider will have to quickly update the service accordingly. The provider will also have to create requirements and compete in the market as well. So RE will be an innovative process in SOSD [20][21][22].

2) **Service Requester**

From requester’s perspective, the general steps are;

- Planning system objectives and gathering initial system requirements, and making initial design
- Discovering available services that meet initial requirements and evaluating their functional and non functional requirements
- Iteratively performing the discovery process to enhance the requirements, unless they are agreed upon with available services and design
- Integrating and composing the system, and evaluating if it meets system requirements

B. **Categorization of the issues**

The issues identified in sub-section C of Section IV fall into four categories; some of them overlap in different categories making them interlinked to one another. They are;

1) **Specification Issues**

By specification issues we mean, what the actual application is that we are going to build, what are its requirements and what services are required that will fulfill these requirements, how we will get these requirements and how we will make them complete.

2) **Service Discovery Issues**

This category deals with searching for the services and finding out which of the services actually meet the functional and non functional requirements. As the number of services has increased over the internet in past few years; it is a challenging task to find and select appropriate one among all especially when it comes to automate the system.

3) **Knowledge Management Issues**

We would need some knowledge of previous composition of the knowledge about the functionalities of services, so that it would help in both specification and discovery. Clustering the specification of services according to their functionality by putting same functionality services in same category to reduce the search efforts and increase domain knowledge. These issues can be seen as improvement in central repository where all information is managed.

4) **Composition Issues**

When services are discovered based on their individual functionalities then we need to see if they will work properly in a workflow by making composition and see if the integrated system meets the original requirements.

### TABLE I. **CATEGORIZATION OF ISSUES WITH RE PERSPECTIVES IN SERVICE ORIENTED SERVICE DEVELOPMENT**

<table>
<thead>
<tr>
<th>Specification Issues</th>
<th>Service Discovery Issues</th>
<th>Knowledge Management Issues</th>
<th>Composition Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD4, SD6, SD7, SD11</td>
<td>SD8</td>
<td>SD7, CB7</td>
<td>SD9, SD10, CB5, CB6, CB7, CB9</td>
</tr>
<tr>
<td>SD2, SD3, SD6, CB5, CB6, CB7, CB9</td>
<td>SD1, SD2, SD3, SD8</td>
<td>SD3, SD4, SD5, SD6, SD7, SD8, SD11</td>
<td></td>
</tr>
</tbody>
</table>

Table I shows classification of issues with respect to service provider and requester’s perspectives. The knowledge management issues though not explicitly much stressed by the researchers, but they do cover most of the issues from service requester’s perspective. It highlights a need to have Knowledge Management strategy to be integrated in SOSD life cycle to overcome the major issue of RE e.g. in issue SD3 and issue SD4 knowledge of previous discovery should be structured for next iteration to compare the results and find out if the changes in query are bringing the required results or not. But integrating knowledge management process with SOSD is still an open research challenge.

VI. **CONCLUSION AND FUTURE WORK**

The focus of requirement engineering phase in SOSD is not on how to build the system but rather what services are required for the system we wish to build. The number of web services launched over internet has increased
exponentially over the past few years. Locating suitable web service is a challenge for Service Oriented System developers especially during requirement engineering phase. In SOSD, the RE phase has two perspectives; service provider and service requester. In both cases the developers are dealing with different issues which we have mainly categorized, based on our finding, into specification issues, service discovery issues, composition issues and knowledge management issues. The issues highlighted in literature shows that the research community is still trying to resolve problems of RE in this new paradigm of Service Oriented Development. The processes proposed for CBSD provides guidelines on how RE should be in SOSD keeping in view the web service as a software component with web interface and which is available on internet. Further efforts are required for inheriting CBSD processes for Service Oriented paradigm for doing further research on existing processes. This would enable us to know what kinds of modifications are required to make them suitable for Service Oriented Architecture. There is a need for a standard RE process for SOSD to guide both developers and service providers for performing steps systematically and which should enable them to know how to tackle different issues that arise in this process. From the findings, we have formulated hypothesis that SOSD process requires the knowledge management process to be integral part of the SOSD life cycle. This might help in resolving RE issues especially when the process is to be automated.

REFERENCES
