An Improved methodology for Service Oriented Architecture

Marzieh Emadi
Department of Information Technology and Communication
Payam Noor University
Tehran, Iran
Emadi_marzieh@yahoo.com

Mohamad Davarpanah Jazi
Department of Computers and Information Technology
Foulad University of Technology
Fouladshahr, Isfahan Iran
mdjazi@cc.iut.ac.ir

Reza Asgari Moghadam
Department of Information Technology and Communication
Payam Noor University
Tehran, Iran
askari@pnu.ac.ir

Faranak Bahredar
Faculty of Engineering
Amin University
Fouladshahr, Isfahan Iran
f.bahredar@gmail.com

***Abstract*—With the growing trend of software systems in enterprises, normal approaches for software development could not satisfy these systems expectations. Service oriented architecture is proposed as a solution to this problem. Many methodologies were introduced toward this architecture, but they have some problems. In this paper an improved methodology is presented and its advantages over other methodologies in deploying a learning system have been studied.**

Service Oriented Architecture; Methodology; Service; Phase

I. Introduction

In this section, Service Oriented Architecture (SOA) and the characteristics of SOA development methodologies are introduced.

A. Service Oriented Architecture

In software engineering, a **Service-Oriented Architecture** is a set of principles and methodologies for designing and developing software in the form of interoperable services. These services are well-defined business functionalities that are built as software components that can be reused for different purposes. SOA design principles are used during the phases of systems development and integration.[8]

In order to efficiently use a SOA, the architecture must meet the following requirements:

- Interoperability among different systems and programming languages that provides the basis for integration between applications on different platforms through a communication protocol.
- Desire to create a federation of resources. Establish and maintain data flow to a federated database system. This allows new functionality developed to reference a common business format for each data element.

The following guiding principles define the ground rules for development, maintenance, and usage of SOA:

- Reuse, granularity, modularity, componentization and interoperability.
- Standards-compliance (both common and industry-specific).
- Service identification and categorization, provisioning and delivery, and monitoring and tracking.

SOAs provide a set of guidelines, principles and techniques in which business processes, information and enterprise assets can be effectively organized and deployed to support and enable strategic plans and productivity levels that are required by competitive business environments.

B. Characteristics of SOA Development Methodologies

According to the related related , the characteristics of SOA are listed below.[8,*10*]

Delivery strategy: Three common strategies exist in delivering a SOA, depending on the amount of front-end analysis of the business domain and the treatment of existing legacy systems [12]. The top-down strategy is closely tied to an organizations' existing business logic, from which the required services are derived. The bottom-up strategy is the opposite in that it focuses on legacy systems, and Web services are built on an as-needed basis. The meet-in-the middle (agile) strategy finds a balance between incorporating service-oriented design principles into business analysis environments without delay.

978-1-4673-0089-6/12/$26.00 ©2012 IEEE

Lifecycle coverage: Some proposed approaches aim to support the full SOA lifecycle, including planning, analysis and design, construction, testing, deployment, and governance activities, while others limit their scope to a subset of these phases, such as analysis or design.

Degree of prescription: SOA methodologies range from the most prescriptive ones that specify phases, disciplines, tasks, and deliverables for each of them, while others provide less detail, by purpose or not, leaving room for more flexibility and tailoring of the approach depending on the project context.

Availability: A number of methodologies proposed by industry players such as IBM, Sun, Microsoft, and others, are proprietary and the detailed specifications are not openly available. In contrast to open methodologies whose documentation is available to the interested public, for the proprietary methodologies it is difficult to fully analyze their capabilities and to make comparisons.

Process agility: A number of methodologies suggest an agile approach to Service Oriented Development in order to address risks and add flexibility to change. Yet, some others follow a more rigid approach in the process lifecycle, or do not address the issue of agility at all.

Adoption of existing processes/techniques/notation: A large number of SOA methodologies propose reusing proven existing processes like XP and RUP, and techniques like OOAD, CBD, and BPM, seeing service-oriented development as an evolutionary rather than revolutionary step in software engineering. Also standardized notations, such as UML and BPMN, are being adopted to visually model various artifacts.

Industrial application: It is important that a methodology be validated in proof-of-concept case studies to show that it has practical applicability and to refine it based on feedback from the case studies. Unfortunately, most of the existing SOA methodologies are at an early stage and have not been applied yet in industrial projects.

Supported role(s): A service-oriented methodology may support the provider view, the consumer view, or both the provider and consumer views in an integrated framework. In the consumer's view, development is declarative and business process oriented through service composition, while in the provider's view it is programmatic and component oriented.

The rest of the paper is organized as follows. Section 2 briefly discusses related methodologies. Section 3 presents the details of our improved service-oriented engineering methodology. Advantages of this improved methodology are described in Section 4. Section 5 concludes the paper..

II. PRELIMINARIES

SOA methodologies are listed in this section. There are many ways to produce a system in which we can develop service oriented systems. But the point here is that these methods do not satisfy all the requirements of enterprises.

A. SOAD(Sevice Oriented Analysis and Design)

This method is presented by IBM Company. SOAD proposes elements that should be part of a service-oriented analysis and design methodology, hence it is an abstract framework rather than a holistic methodology. SOAD builds upon existing, proven techniques, such as OOAD, CBD, and BPM. It also introduces SOA-specific techniques, such as service conceptualization, service categorization and aggregation, policies and aspects, meet-in-the-middle process, semantic brokering, and service harvesting.[13]

B. SOMA(Service Oriented Modeling and Architecture)

Service Oriented Modeling and Architecture has been used to conduct projects of varying scope in multiple industries worldwide for the past five years. SOMA is a software development life-cycle method invented and initially developed at IBM for designing and building SOA-based solutions. This method defines key techniques and provides prescriptive tasks and detailed normative guidance for analysis, design, implementation, testing, and deployment of services, components, flows, information, and policies needed to successfully design and build a robust and reusable SOA solution in an enterprise.[1]

SOMA consists of three steps: identification, specification and realization of services. It helps in the determination of services and components and realizes them. However, SOMA lacks openly available detailed description of the methodology, which makes it difficult to further analyze its capabilities.[9,1]

C. RQ(Repeatable Quality)

Repeatable Quality methodology is a proprietary methodology by Sun Microsystems that is based on a RUP-like iterative and incremental process consisting of five phases: inception, elaboration, construction, transition, and conception. UML compliant artifacts are used for documenting various deliverables of these phases.[5,12]

D. SOAF(Service Oriented Architucre Framework)

SOAF provides a systematic approach and a well-defined process to guide the design, evaluation and development of an SOA. This method consists of five phases: information elicitation, services identification and matching, service definition, services realization, roadmap and planning.[4]

E. SOUP(Service Oriented Unified Proces)

This approach is primarily based on the Rational Unified Process. Its lifecycle consists of six phases: incept, define, design, construct, deploy, and support. However, SOUP lacks detailed documentation and leaves room for adaptation. It is used in two slightly different variations: one adopting RUP for initial SOA projects and the other adopting a mix of RUP and XP for the maintenance of existing SOA rollouts.[6]

F. Papazzoglou

This methodology describes a service development methodology for Web services and business processes. The proposed methodology comprises six phases based on the Rational Unified Process, Component-Based Development and Business Process Modeling. The methodology also provides service design guidelines for Web service interfaces and service flow models that maximize cohesion and minimize coupling. However the proposed approach does not provide enough guidelines for explicit consideration of service model artifacts.[10]

G. Steve Jones'

The scope of this top-down methodology consists of the first steps in a project necessary to ensure that true SOA properties are satisfied in the final delivery. It is technology agnostic and takes a top-down business view reaching up to the point of service candidate discovery (i.e. identification). The methodology adopts a broadly four-step process (What, Who, Why, and How), of which the first three are covered in preparation for the fourth step.[11]

H. RUP for SOA

Rational Unified Process (RUP) is a unified method planning form for large business applications that provides a language for describing method content and processes. The well defined mapping of SOA and RUP leads to successful completion of RUP software projects to provide services to their users. In this method Digital Office Assistant (DOA) is presented. DOA is a multi user SOA type application that provides appropriate viewer for each user to assist him through services.[14]

I. MSOAM(Mainstream SOA Methodology)

This methodology is classified as "mainstream" because it provides a set of generic processes and practices that almost always require further customization when incorporated into enterprise environments. It is therefore best viewed as a starting point. The steps within the processes raise issues and considerations and further propose sequences and priorities, all related to the analysis and design of services.

This methodology is designed to address key decision points related to weighing the attainable strategic benefits of SOA via top-down strategies against the tactical preferences of bottom-up approaches.[7,8]

III. Improved Methodology

We present a method which consists of ten phases. These phases provide all the requirements of system architecture for planning, designing, implementing and maintaining the system. This method is presented by reducing the weaknesses and disadvantages of previous methods. First five phases present the architecture logical design and last five phases propose the physical plan for implementing the system and services (fig. 1). Characteristics of these phases are presented below:

1) Requirements Engineering and Analysis: one of the most important phases in producing a software system is to understand enterprise requirements. In some methodes the reqiurements are drown out before the first phase. But because of it's role in system cohesion, requirement identification and analysis is the first phase.

This includes reviewing business goals and objectives that drive the development of business processes. The analysis phase examines the existing services portfolio at the service provider's side to understand which policies and processes are already in place and which need to be introduced and implemented.

2) Planning and Project Management: after investigating and analaysing the requiremnets, now it is time to manage the system on the hand. The Service Architecture event needs to be scoped and planned. This requires a certain amount of pre-planning to ensure any information required tokeavilable at the actual events such as:

- Identifying the project stages and their priorities.
- Determining the subsystems.
- Determining the programming language and database management system.
- Assessing the risks and presenting the technical solutions.
- Estimating time and cost of each stage.
- Determining management processes.

3) Reuse Management: SOA solutions are hybrid in nature and typically include multiple solution types. Reusebilty in SOA allowes modules to be used in other projects. The method content, like all SOA solution types, is separated from the variable method content that is dependent on specific solution types. The variable method content such as tasks, work products, roles, and guidance specific to each of the solution types is defined.

4) Services Identification: The key requirement in this phase is to understand the business environment and to make sure that all necessary controls are incorporated into the design of a service-oriented solution. Activities in this phase include identifing the services and subserivces which can be done in tow ways:

- Using bottom-up strategy to analyze the inheritable enterprise system.
- Using top-down strategy in which at first it analyzes enterprise scope and then the subsystems, sub-services and the business process will be determined. This is more costly and time consuming than the bottom-up strategy.

5) Service Definition: in this phase, details of services stated in the prvious stage will be defined. This phase consists of:

- Specifying the quantitative and qualitative requirements.
- Specifying the service policy to advertise supported protocols, the constraints on the content of exchanged messages and QoS features, such as security and manageability assertions.
- Naming the services.

6) Service Aggragation: Services may be offered at different levels with different granularity. Services will be assigned to various layer, such as: business layer, application layer and orchestration layer. For the defined services a variety of tasks needs to be performed as enumerated below:

- Service classification
- Service composition and decomposition
- Tasting the services by development team and if needed classifying and composing them again.

7) Service modeling: During which conceptual processes and services are transformed into a set of related, platform-agnostic interfaces. Designing a service-oriented application requires developers to model and define well-documented interfaces. Service design has its own special characteristics and techniques, which are:

- how to apply standards and technologies
- database model
- entity relationship diagram
- encryption model
- Detailed design and quality assurance services
- Effects of changes in services on other services performance
- Reusability

8) Implementaion: the developemnet team is divided into different sub-teams and programing will be done based on the information in previous phases. The service implementation phase also considers service requesters. On the service client side although the service requester progresses through similar lifecycle elements as the service provider, different tasks are performed during each construction step.

9) Test: Service testing is generally characterized as a validation exercise ascertaining that requirements have been met and that the deliverables are at an acceptable level in accordance with existing standards during the analysis, design and implementation phases. Some questions can be answered to control the results provided by services:

- Are the requirements provided for the services?
- Are the services in line with the requirements?
- Does each service implement the process needed?
- What are the authorized requests for accessing the services?
- What are the exceptions that services should deal with?
- Do the services provide the qualitative requirements?
- Can services be composed?
- Is the interface changed according to the changes in services?

After the final test and specifying the above questions, it is necessary to check the entire system in order to maintain the system integrity.

10) Deploy and support: Deployment means rolling out new processes to all the participants, including other enterprises, applications and other processes.

The service supporting phase concerns itself with service level measurement and monitoring is the continuous and closed-loop procedure of measuring, monitoring, reporting and improving the quality of service of systems and applications delivered by service-oriented solutions.

IV. Comparison with other methodologies

We have deployed our methodology on a learning system (such as university learning system), and after investigating the results, these results obtained:

- All the needed requirements are extracted during the first phase. But in some methodologies like Steve Jones' this phase is ignored.
- The second phase is composed of characteristics of some methods such as SOAD, RQ, RUP for SOA and MSOAM. So it can handle and manage the project more precisely.
- Phase three is embedded in all methodologies. But in our method all aspects of reusability are studied and concerned.
- Because of the importance of identifying the services, this stage goes in absolute phase, so the development team spends more time to recognize the services.
- This improved method has a great concern about maintaining and deploying the completed project.

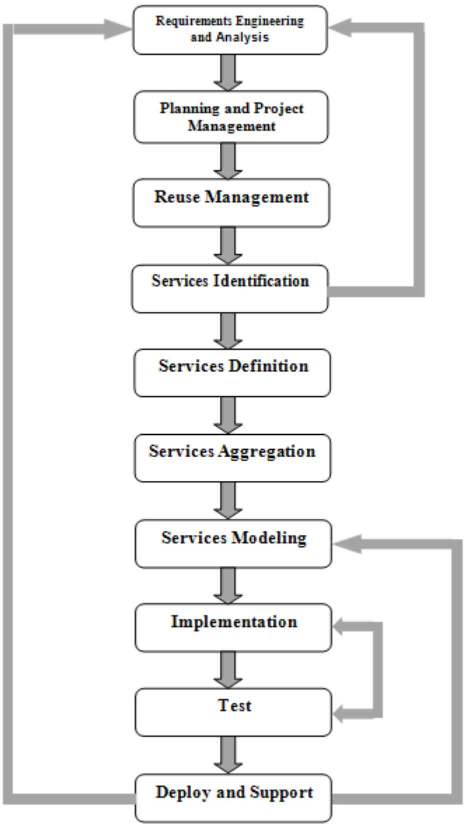

Figure1. Improved Methodology

V. Conclusion

In this paper we have described an improved methodology for Service Oriented Architecture. The methodology takes into account a set of development models (top-down and bottom-up and stresses some phases that are not in great concern in other methodologies.

In contrast to previous methodologies, the proposed methodology emphasizes activities revolving around service provisioning, deployment, execution and monitoring. Father research is being planned to use the proposed methodology in learning management systems at Iranian universities as this project.

REFERENCES

[1] A. Arsanjani, “SOMA: A Method for Developing Service-Oriented Solutions”, IBM SYSTEMS JOURNAL, VOL 47, NO 3, August 2008.

[2] A.Arsanjani, “IBM's SOA Method:SOMA, Service-oriented Modeling and Architecture”, Available at: https://www.ibm.com/developerworks/mydeveloperworks/blogs/AliArsanjani/entry/soma_service_oriented_modeling_and?lang=en, 2006.

[3] C. J. Paul Tablang, “Rational Unified Process (RUP)”, Journal of Computing Sciences & Information Technology, 2010.

[4] A. Erradi, “SOAF: An architectural framework for service definition and realization”, in Proceedings of the IEEE International Conference on Services Computing, pp. 151-158, September 2006..

[5] M. F. Gholami, J. Habibi, F. Shams, S. Khoshnevis, “Criteria-Based Evaluation Framework for Service-Oriented Methodologies”, International Conference on Computer Modeling and Simulation, pp.122-130, 2010.

[6] K. Mittal, “Service Oriented Unified Process (SOUP)”, Available at http://www.kunalmittal.com/html/soup.shtml, 2006.

[7] S. Boroumand, “Working with SOA and RUP”, Available at: http://www.soamag.com/I16/0308-1.php, 2008.

[8] T. Erl, “Service-Oriented Architecture: Concepts, Technology, and Design” Upper Saddle River: Prentice Hall PTR, 2005.

[9] M. Keith, “SOMA, RUP and RMC: the right combination for Service Oriented Architecture”, IBM® WebSphere® User Group, Bedfont, March 2008.

[10] M. P. Papazoglou, W. J. Van Den Heuvel, “Service-oriented design and development methodology”, International Journal of Web Engineering and Technology (IJWET) , 2006.

[11] E. Ramollari, D. Dranidis, A.J.H. Simons, “A Survey of Service Oriented Development Methodologies”, In Proceedings of the 2nd European Young Researchers Workshop on Service Oriented Computing, Leicester, UK, June 2007.

[12] SUN Microsystems, “SOA RQ methodology - A pragmatic approach”, available at http://www.sun.com/products/soa/soa methodology.pdf, 2007.

[13] O. Zimmermann, P.Krogdahl, C.Gee, “Elements of Service-Oriented Analysis and Design”, IBM Corporation, Available At http://www-128.ibm.com/developerworks/library/wssoad1/, 2004.

[14] S. Hussain, S. M. Saqib, B. Ahmad, S. Ahmad, “Mapping of SOA and RUP: DOA as Case Study”, JOURNAL OF COMPUTING, VOLUME 2, ISSUE 1, 2010, ISSN 2151-9617 HTTPS://SITES.GOOGLE.COM/SITE/JOURNALOFCOMPUTING, pp. 104-106, JANUARY 2010.