ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Service Oriented Architecture Testing

[1] Mageusha.U,[2] Dr. Sudhamani
[1] Research Scholar,Rayalaseema University
[2] Research Supervisor, Principal, Vivekananda Degree College for women.
Mahalakshmi Layout, Bangalore.

Abstract: Service Oriented Architecture is a familiar concept nowadays. When asked for definition of SOA no two people would end up giving the same answers and most likely to receive different thoughts. Describing SOA as Infrastructure enablers and someone looks SOA as an efficient/ most effective way of delivering services to Information Technology. Service Oriented Architecture can be defined as "A loosely-coupled architecture, which is designed effectively to be reused with the aim of meeting the business need with flexibility and maintainability with an approach to render the services to the quick to market projects for software users".

Keywords: SOA, Information Technology, Service Oriented Architecture, Loosely – Coupled.

INTRODUCTION

Service-Oriented Architecture (SOA) – it is a strategy for software development methods to all the software deliverables of an organization, like a basic architectural design of a software that represents the various components and interaction and accessibility of a software. Therefore, service-oriented architecture is the creation of all assets of a software organization with strategy - oriented methodological services.

What is Service?

Services are features, functions or components of a program, designed and coded to easily interact with other features of functionalities. these services should be simple so that end users of the software can be easy to understand, and not as be a rocket science.

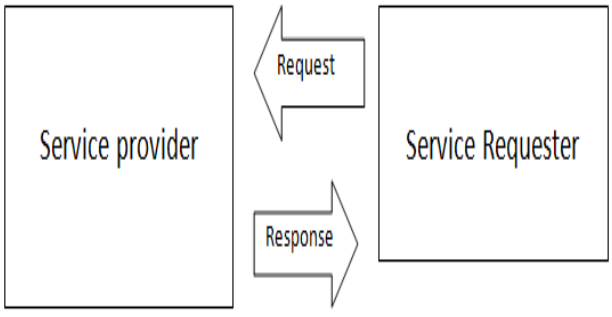

Fig 1: Overview of Service.

- Services can be a granular unit or a piece of code that delivers/ serves a business need, which can be referred or reused by any other software application to enable the quicker development.
- Services should be easy to refer and reuse with minimal configuration.
- Services can be compared to built-in components of any software and any application can be built on top of that based on the application requirements. Reusing them from the application or business process as and when need is possible.
- Services are defined more by the business requirements they demand rather than as a flow of data.

Service-Oriented Architecture (SOA) is a way of designing, developing, deploying, and managing enterprise systems where business needs and technical solutions are closely aligned. SOA offers many potential benefits, such as cost-efficiency and agility. However, adopting SOA is not without considerable challenges. For example, the most common way to implement a SOA-based system is with Web services, but the standards that define Web services are evolving rapidly and many of the Web services tools are still somewhat immature. There is also the question of how to leverage existing legacy assets within a SOA context. Perhaps most importantly, there are serious challenges related to the testing of SOA-based systems that must be addressed before the SOA paradigm will enjoy broad-based success.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

The Evolution of SOA

Service Orientation (SO) is the natural evolution of current development models. First object-oriented models; then came the component-based development model in the 90s; and now we have service orientation models (SO). Service orientation retains the benefits of component-based development (self-description, encapsulation, dynamic discovery and loading), but there is a shift in paradigm from remotely invoking methods on objects, to one of passing messages between services. Schemas describe not only the structure of messages, but also behavioral contracts to define acceptable message exchange patterns and policies to define service semantics. This promotes interoperability, and thus provides adaptability benefits, as messages can be sent from one service to another without consideration of how the service handling those messages has been implemented.

Fig 2: Evolution of SOA.

Service orientation has evolved to meet the demand of building loosely coupled distributed software that facilitates integration of existing components and adaptability. With the emerging of the Web Services, demand for service orientated architecture has made service-oriented software development possible with the support of development tools and wide industry reusability. service orientation is independent of architectural patterns and technology which would have access to old components.

SOA – Benefits

SOA benefits are basically categorized as follows:

- Reducing integration expense
- Increasing asset reuse
- Increasing business agility
- Reduction of business risk

These four main benefits offer quicker returns at many parts and hierarchies of the organization, mainly based on the set of business requirements and needs that the company applying the underlying architecture to.

SOA being an abstract concept, understanding its core principles is essential to succeed. Stories of SOA implementation failures have been doing rounds primarily due to overlooking of its fundamental principles of business involvement and governance. The case with SOA testing is no different. The term is even more ambiguous, and any misinterpretations can result in failures of test implementations. The foundation of SOA testing lies on three basic beliefs - early testing, optimized coverage and faster regression. When it violates one or more of these, it is more likely to fail. From my experience, here are some of the typical situations where SOA testing does not give its intended benefits:

- Late start – to get the benefits of SOA it is recommended that as early as Services are layered in the architecture of core business functionality. possibility of Early detection and fixing of defects prior to integration of components is performed to avoid cost escalations due to rework.
- Testing of non-core services – services that are rarely usable and does not belong to core business functionality can be planned in later phases of testing like end to end testing to avoid repetition of testing such components.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

- Poor choice of SOA testing tool - Selection of a SOA testing tool is to be performed more consciously to meet the demand of business requirements.
- Ineffective automation – to avoid wastage of automation efforts and ensure right coverage of modules on time, Identifying the right modules or software components that undergo frequent changes can be considered for automation rather than automating anything and everything
- Redundant testing – collaboration with test experts on a technical skill should go hand in hand to plan the SOA testing in various test phases to ensure right system testing.

SOA Governance

SOA Governance is about ensuring that each new and existing service conforms to the standards, policies and objectives of an organization for the entire life of that service.

Why is SOA Governance needed?

In today's challenging business environment, SOA Governance plays an vital role in delivering the software objectives by providing right team structure with needed technical knowledge, business commitment and support that would be provided during various phases of software development life cycle as that of development, implementation and management of SOA.

SOA Governance provides the following benefits:

- Benefits of Business is realized
- Flexibility in using the Process in Business
- Improved time to market in a competitive market environment
- Maintenance of services with Quality
- Ensuring consistency of service
- Test the right software deliveries
- Coordination and clear Communication within businesses

SOA – based applications – Testing strategies

The nature and effectiveness of Service Oriented Architecture based applications calls for need of defining the test strategies and using of right toolset and test approaches. Older test strategies and approaches might not be adequate to test the latest SOA based software codes.

SOA consists of various technologies. Applications built using SOA has various services which are loosely coupled.

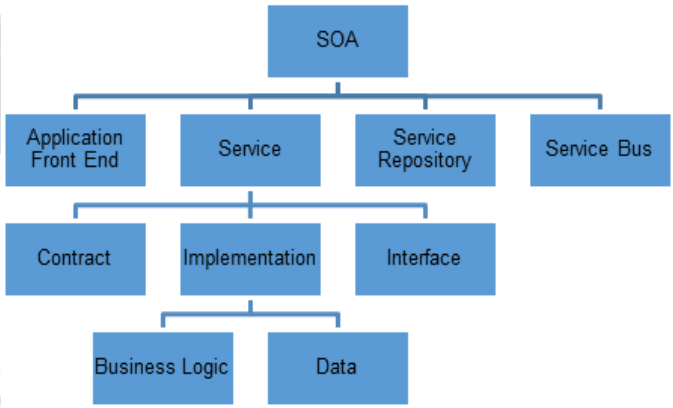

Fig 3: SOA Services

SOA Testing should focus on 3 system layers

Services layer –

Includes services exposed by systems which are derived from business functions.

For example –

- Service can display the respective data and date being entered. By the user, Services layer consists of the services which receives the data from the Database–
- Login Service

Process layer –

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Contain implementation aspects of business processes. let us focus on process models which are implemented in user interfaces and process layers...
The focus in this layer will be on user interfaces and process.
Creating, Adding, editing and deleting data

Consumer layer –
This layer mainly comprises of UI based end user interfaces
Based on the layer, the testing of an SOA application is distributed into three levels.

1. Service level
2. Interface level
3. End to End level
    - Top Down approach is used for Test Designing.
    - Bottom Up approach is used for Test Execution.

Challenges in Testing SOA-based Applications:
Planning should be done with utmost care while planning to test SOA based application, failed to do so will result with more challenges for testers.

- Increased test scenarios as and when the software reaches the stability
- Software requirement verification
- Identifying the right tool and right capable testers to facilitate smooth testing.
- Multiple stakeholder's involvement
- Validation of accuracy of results
- Testing dependent components in a functional flow.

SOA Test Approach:

- To break down the complex software solution into granular, demo able, measurable and manageable piece of code to build quality deliverables. SOA test approach should follow the sequence of Component testing, Service testing, Workflow testing, System testing, Integration testing, Link testing.
- The foundations to successful SOA testing are as follows: a right combination of Test tools, Technology experts and Domain experienced resource should be deployed.
- Test Approach should be designed along with project and Technical requirements for a Project.
- Test effort should be estimated and sufficient budget is allocated for testing.
- Implement certain measurable metrics thought the lifecycle of the project.
- Nonfunctional tests and security tests should be planned from the initial phase of a Project.
- Proper reviews should be followed during early stages of the project to ensure that test efforts are planned throughout the Development Life cycle.

Test Planning Approach,

- Testers should understand the complete architecture of the application.
- Break down the application into independent services
- Three components should be organized Architecture – Data/ Back end, Services, and front end/ end user applications.
- Right business scenarios are identified and all components to be completely analyzed.
- Classification between Business scenario and application specific scenario is performed
- All the business scenarios are tracked through traceability Matrix.
- All the below categories of test phases can be followed while performing SOA testing.
    - Governance Testing
    - Service-component-level testing

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

- Service-level testing
- Integration-level testing
- Process/Orchestration-level testing
- System-level testing
- Security Testing

Test Execution Approach

- Every component should be tested for right service delivery.
- Data flows through services as validated through end to end or Integration testing.
- Complete end to end testing is performed in the system to test both front end, Backend services and their synchronization.
- Performance testing and Response time testing should be done with various data loads for optimum performance.
- SOA Architecture – Testing approach

We should understand, how to break down the architecture to its component parts, working from the very basic to the most complex components, testing each component, then the integration of the complete architecture.

Test automation framework for SOA:

Devising automation solution for SOA starts with feasibility study and tool analysis for suitability for SOA. It is necessary to design test scenarios that focus on individual transactions as well as business workflows so that issues, if any, can be identified and isolated.

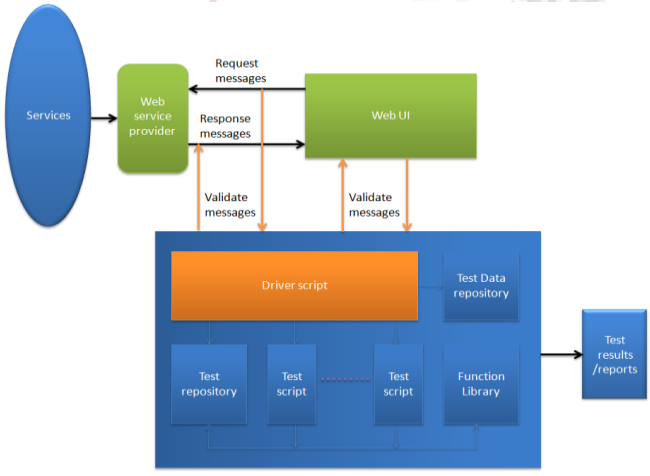

Fig 4: Test automation framework

Some popular SOA test tools available to test include: soapUI Pro, HP Service Test, Parasoft's SOAtest to name a few.

In some cases, web services or other components may not have an interface hence a UI may need to be developed to test them. Non-UI based testing is also quite a common scenario.

CONCLUSION

The study of Service oriented architecture helps the testers to understand the software architecture of enterprise including priorities, practices and concepts of application side, various services and its repository. Service oriented architecture is a collection of layers, and each layer has a unique functionality, the first layer (Services Layer) Includes services exposed by systems which are derived from business functions, and this layer delivers an appropriate business that provides logic and test data which is necessary for various phases of testing application. The second layer (Process Layer) can address various user interfaces and process. Finally, the third layer, (Consumer Layer) This layer mainly comprises of user interfaces.

SOA will provide the support for the business process / Services / process of the organization and ensure that right selection, understanding and utilizing of test tools blended with correct technology starting from early phase of the project results in quality deliverable with fixes done in early phases that results in cost reduction involved in maintenance of a project and quicker to Markets. .

REFERENCES:

1) Mathur, AP (2008) "Foundations of software testing". Dorling, Kindersley (India), Pearson Education in South Asia, Delhi, India.

2) Yoo, S, Harman M (2012) "Regression testing minimization, selection and prioritization: A survey". Softw Test Verif Reliab 22(2):67–120. https://dl.acm.org/citation.cfm?id=2284813.

3) B. Beizer, Software Testing Techniques, Second Edition. Published by
dreamtech, New Delhi, pp. 135-143, 2008

4) Z. Li, M. Harman, and R. M. Hierons, "Search algorithms for regression test case prioritization," IEEE Transactions on Software Engineering, vol.33, no. 4.

5) Testing Strategies and Tactics for Mobile Applications [Online]. Available: http://www.keynote.com/mobile-app-testing.html