

Testing SOA based Applications: Challenges and Test Approach

This whitepaper aims at providing a brief introduction to the concept of testing applications based on Service Oriented Architecture (SOA).

As an organization grows, the level of complexity and operation of software architectures also grow due to disparities among various hardware, operating systems and applications. A major amount of cost, time and resources are dedicated towards the integration of these components/systems. Each pairing of applications necessitate an additional interface which further adds to the complex design and often, these efforts may not produce reusable logic.

Service Oriented Architecture (SOA)

Service Oriented Architecture (SOA) was proposed to address these problems. SOA provides a very flexible and controlled architecture that uses web service technologies to redefine functions within all underlying applications as independent services. These services are loosely coupled and can be invoked through well-defined interfaces to provide data and logic for various business processes.

The business benefits of having an infrastructure built on SOA are:

1. Increased efficiency and optimized performance of business processes due to streamlining of underlying systems
2. Cost reduction through reuse of existing assets as well as reduced maintenance and integration costs
3. Improved flexibility towards adaptive business changes
4. More control over services and flow of information
5. Abstraction level for ease of use to end customer

Architecture (SOA):

SOA integrates business processes within an enterprise by structuring large applications as a collection of smaller modules called services. Web services are mostly used to implement SOA.

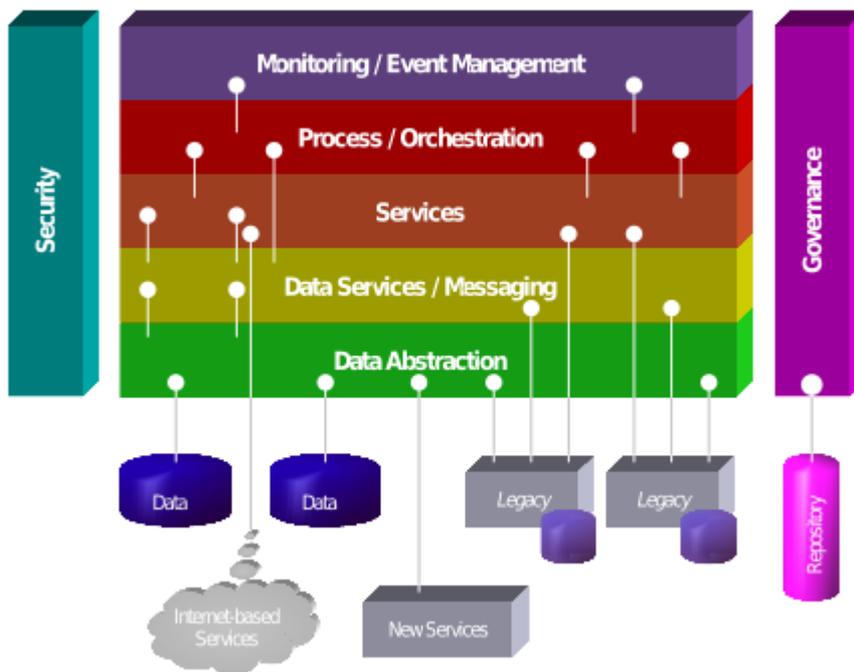


Fig1. *SOA meta-model [Service-oriented_architecture] – Wikipedia

Each module in SOA can act as a service provider, service consumer or service registry depending on the requirement of the invoking business process.

SOA Governance

As defined by Anne Thomas Manes, governance is: “The processes that an enterprise puts in place to ensure that things are done [...] in accordance with best practices, architectural principles, government regulations, laws, and other determining factors. SOA governance refers to the processes used to govern adoption and implementation of SOA.”

The SOA governance encompasses enforcement of following policies:

1. Service lifecycle management
2. Quality of Service(QoS) parameters - reliability, performance, scalability, robustness, capacity, accuracy, exception handling, accessibility, integrity, interoperability security and availability
3. Regulatory policies and compliance to standards/laws
4. Business and infrastructure policies like backup, disaster recovery etc.
5. Change management

Challenges in Testing SOA-based Applications:

The unique characteristics of SOA present testers with many challenges due to which careful planning is required for testing SOA based applications.

1. Testing of interfaces and interdependencies amongst underlying applications
2. Validation of accuracy of results from interactions between individual and integrated services
3. Complex test scenarios due to heterogeneity of applications
4. Grey box testing for low-level implementation of architecture
5. Increased end-to-end testing for regression testing/maintenance phase
6. Simulated roles of a consumer of service rather than end user
7. Verification of business requirements
8. Involvement of multiple stakeholders
9. Identification of right tools to facilitate testing
10. Migration testing (in case of existing applications prior to SOA implementation)

Test approach for SOA:

Regular testing models require Unit, System, Integration, and Acceptance testing at corresponding phases for a product. These are in the same way applicable to SOA in terms of services as shown below:



Fig 2. Test Approach for SOA

However it is equally important that testers not only consider components but also business flows to ensure that the integration is seamless.

Most of the messaging and application integration in SOA is currently designed and implemented using XML, Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL), Universal Description Discovery and Integration (UDDI) so it is necessary that the tester is well-versed with these languages/protocols. Also, domain knowledge expertise is very important to design effective tests to check applications and services involved.

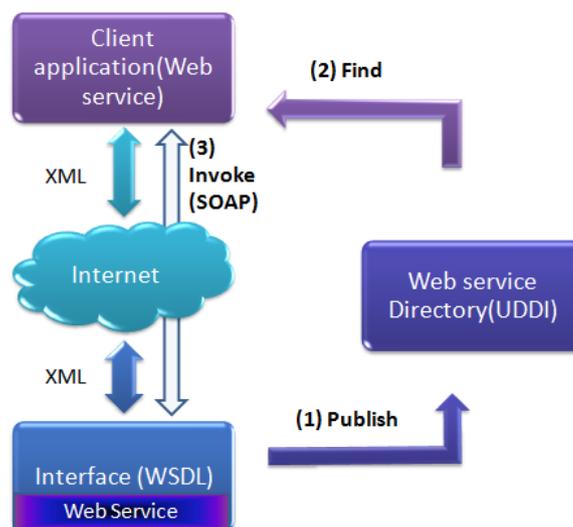


Fig 3: Key Elements of SOA model

SOA testing requires following types of tests to be executed including areas which are specific to an SOA setting:

1. **Governance Testing:** This is carried throughout the project lifecycle to ensure that all the policies and standards are adhered to. This includes specific tests, monitoring tools etc.
2. **Service-level testing:** Each service provided by an application is individually tested. This includes data and events testing
3. **Process-level testing:** This includes testing for workflows, configurations interoperability and reusability (consumability and composability) of service
4. **Integration testing:** This includes testing of Service-Application integration and focuses on service interfaces. Communication across layers and network protocols is verified
5. **System testing:** This is to verify that the SOA implementation has met all business rules, policies and standards defined
6. **Security Testing:** This includes checking for vulnerabilities within system and threats such as Denial-of-Service (DOS), unauthorized access etc. throughout project lifecycle

7. Acceptance testing (UAT): This includes executing end to end scenarios through key business areas of system by the end user and/or stakeholders
8. Performance testing: This is essential to determine latency issues, redundancy, service disruption due to synchronous transactions and other performance bottlenecks
9. Regression testing: This is retesting of system after a bug fix to ascertain that it functions as intended and no new defect has been introduced due to changes

Tools & Test framework for automation and performance testing

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.

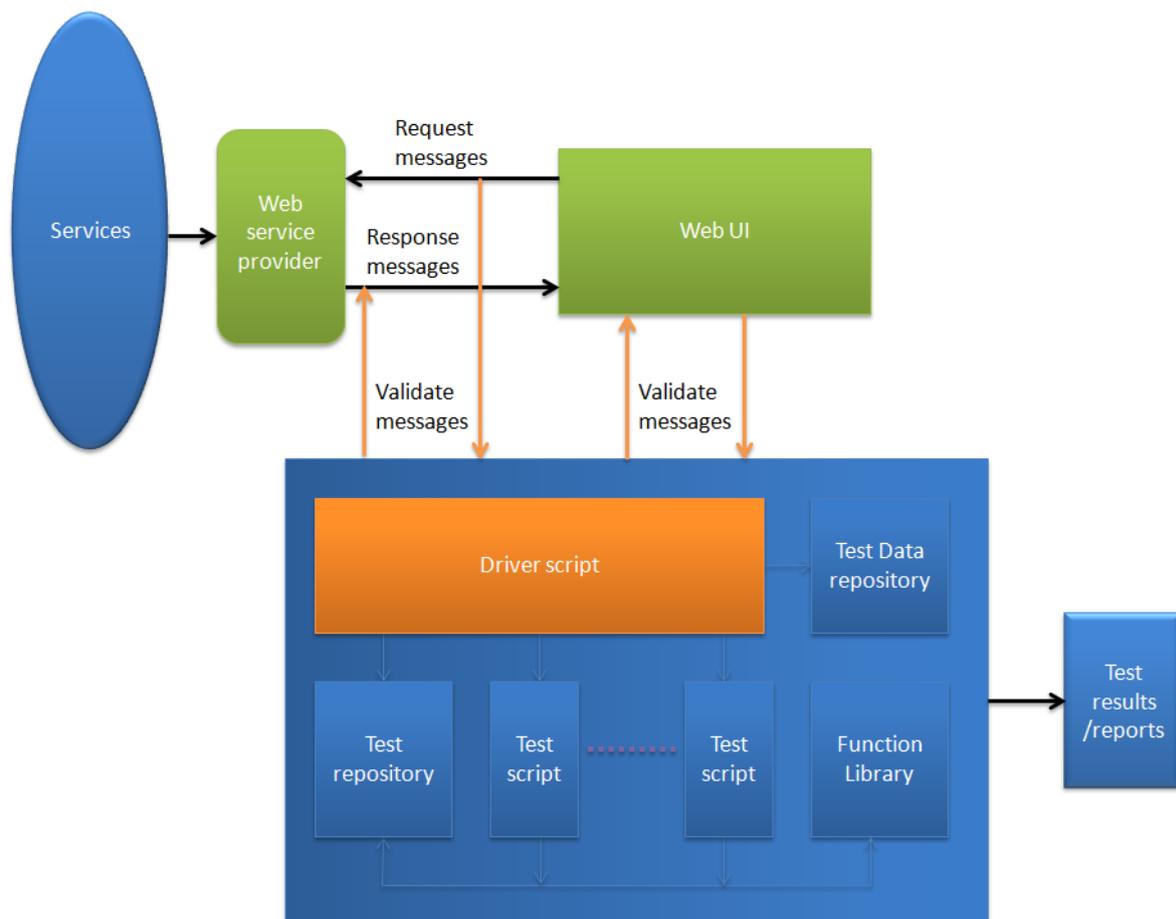


Fig 4: Test automation framework for SOA

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

Performance testing

Performance testing is used to analyse the throughput and load capability of SOA based applications for variations in input and user loads. It helps identify bottlenecks and dependencies among points of services. Most SOA tools have load and scalability test features integrated into them along with good reporting capabilities.

Conclusion:

SOA has simplified and built up robustness of an otherwise, distributed network of applications within enterprises. However, from testing perspective the complexities of such systems increase and become more challenging to test. Testers need to use their expertise to plan and execute SOA testing and acquire relevant skills necessary for testing the architecture and implementation of SOA coupled with knowledge of business rules that drive the workflows within SOA.

Author:

Kavita Pal
Associate Manager (Quality Engineering)
QA InfoTech