Test Automation Framework for SOA Applications

Ganesh Moorthy
Harish Perumal
Sreeja Unnikrishnan
Table of Contents

• Introduction
• SOA Architecture
• SOA Testing
• Our Automation Framework for SOA
  – Methodology
  – Key components
  – Benefits
• Q & A
Introduction

According to a Gartner report, “SOA will be used in more than 50 percent of new mission-critical operational applications and business processes designed in 2007 and in more than 80 percent by 2010.”

• SOA is an evolutionary trend

• Applications converted to services
  – interfacing or logic
  – Interoperable, managed, independent or dependent and interactive

• Accelerated and unplanned development and deployment of services leads to unplanned and insufficient testing

• The need of the hour is to have a test automation framework
  – Well-defined methodology with a measurable process incorporating early lifecycle testing
  – The right automation tool
SOA Landscape

SOA comprises of three major entities

• Service providers
  – Provide access to business functionalities or systems

• Service consumers
  – Utilize service components to fulfil transactions

• Service Governance
  – Defines, manage and monitors interactions between providers and consumers
SOA Components

- **Business processes**
  - Defines workflow model for business functionalities
- **Service**
  - Provides independent functionality
- **Service adapters**
  - Provides wrapper service
- **Service integration layer**
  - Manage service interaction and integration
- **Service contract**
  - Describe properties of hosted service
- **Data**
  - Actual message in the request to the service and the response
SOA testing

Testing SOA applications is different from traditional testing approaches. Besides regular functional or regression testing, we also need to ensure:

- **Service Level test**
  - Unit testing of service components against specifications

- **Integration test**
  - Test service contracts, service interoperability and service binding

- **Orchestration test**
  - Test service workflow defined for business functionality by monitoring service calls and Fault injection testing

- **QoS test**
  - Test for performance, reliability, availability and scalability against SLA

- **Governance test**
  - Validating standards and policies by monitoring service calls

- **Client test**
  - Test response from service consumers in a near production environment
Governance and Regression Testing should be incorporated into all life cycle phases
SOA Test Automation Framework

• Automation Feasibility Matrix
  – Service level
  – Test stage level
  – Tool suitability

• Automation strategy
  – Top-down
  – Bottom-up

• Automation Design
  – Tool set up
  – Transaction modeling
  – Scenario creation
  – Test script generation

• Automation execution and reporting
  – Metrics based
SOA Test Automation Framework

- Automation Feasibility Analysis
  - Service level
  - Test Stage level
  - Tool suitability

- Automation Strategy
  - Top-down
  - Bottom-up

- Automation Design
  - Tool Setup
  - Transaction Modeling
  - Scenario creation
  - Test script generation

- Automation Execution and Reporting
  - Metrics based

- SOA applications need different approach and parameters for feasibility analysis

- Home-grown or external tools need to be evaluated to identify the best fit for the automation requirement with reduced customization cost
SOA Test Automation Framework

- Automation Feasibility Analysis
  - Service level
  - Test Stage level
  - Tool suitability

- Automation Strategy
  - Top-down
  - Bottom-up

- Automation Design
  - Tool Setup
  - Transaction Modeling
  - Scenario creation
  - Test script generation

- Automation Execution and Reporting
  - Metrics based

- Top-down approach involves full fledged test automation without the need for a proof of concept. It is successful when there is prior experience in SOA testing and a readily available tool requiring no customizations.

- Bottom-up approach necessitates a proof of concept usually undertaken for “Green-field” projects and if tool fitment analysis was inconclusive or the automation tool requires a lot of customizations.
SOA Test Automation Framework

In top-down or bottom-up approach, the strategy is followed by a testing lifecycle using the tool as depicted below:
SOA Test Automation Framework

- Automation Feasibility Analysis
  - Service level
  - Test Stage level
  - Tool suitability

- Automation Strategy
  - Top-down
  - Bottom-up

- Automation Design
  - Tool Setup
  - Transaction Modeling
  - Scenario creation
  - Test script generation

- Automation Execution and Reporting
  - Metrics based

- Tool Setup:
  - Setting up the communication parameters by configuration or building custom components

- Transaction Modeling:
  - Understanding transaction activities and providing the necessary input / output parameters and message standards

- Scenario Creation:
  - Map Transactions with appropriate communication parameters and sequence transactions to complete an end-to-end scenario.
  - Include business rules for workflow enablement, data propagation or transformation between sequenced activities

- Test Script Generation:
  - Specify data for input and generate test scripts automatically and store it in centralized repository
SOA Test Automation Framework

- Automation Feasibility Analysis
  - Service level
  - Test Stage level
  - Tool suitability
- Automation Strategy
  - Top-down
  - Bottom-up
- Automation Design
  - Tool Setup
  - Transaction Modeling
  - Scenario creation
  - Test script generation

- Automation Execution and Reporting
  - Metrics based

- Execute in
  - iterative mode to isolate configuration, message or data issues with each atomic transaction
  - batch mode to run the scenarios in an uninterrupted manner

- Perform regression testing by baseline test results and comparing output of subsequent runs
- Perform non-functional/QoS testing
- Generate atomic and aggregate reports
# Automation framework benefits

<table>
<thead>
<tr>
<th>Traditional Utility Scripts</th>
<th>Automation Framework</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demand –Driven Script design</td>
<td>Planned Script Design</td>
<td>Measurable Quantitative results</td>
</tr>
<tr>
<td>Silo Based Script Generation</td>
<td>Collaborative Script Generation</td>
<td>Increased Productivity</td>
</tr>
<tr>
<td>Disparate Storage of Scripts</td>
<td>Centralized Script and Data Storage</td>
<td>Increased Productivity, Enhancement Change Management</td>
</tr>
<tr>
<td>Unsynchronized Requirements and Scripts</td>
<td>Enhancement Requirements Traceability</td>
<td>Accurate Scope Coverage</td>
</tr>
<tr>
<td>Silo based execution</td>
<td>Workflow based execution</td>
<td>Effective Testing leading to lower cost of quality</td>
</tr>
<tr>
<td>Utility Reporting requiring additional scripts for aggregate dashboard</td>
<td>In-Build dashboard reporting</td>
<td>Effective Test Management leading to lower time to market</td>
</tr>
</tbody>
</table>
Thank You

For any details, please contact
Ganesh_Moorthy@infosys.com
Harish_Perumal@infosys.com
Sreeja_Unnikrishnan@infosys.com