
Abstract

SOA is relatively new, so companies seeking to implement it cannot tap into a wealth of practical expertise. Without a common language and industry vocabulary based on shared experience, SOA may end up adding more custom logic and increased complexity to IT infrastructure, instead of delivering on its promise of intra and inter-enterprise services reuse and process interoperability. To help develop a shared language and collective body of knowledge about SOA, a group of SOA practitioners created this SOA Practitioners' Guide series of documents. In it, these SOA experts describe and document best practices and key learnings relating to SOA, to help other companies address the challenges of SOA. The SOA Practitioners' Guide is envisioned as a multi-part collection of publications that can act as a standard reference encyclopedia for all SOA stakeholders.

1.1 Intended Audience

This document is intended for the following audience:

- Business and IT leaders, who need to start and manage an SOA strategy across the enterprise/LOB
- Enterprise Architects who need to drive the vision and roadmap of the SOA program and the architecture of each implementation that falls under it
- Program Managers who need to manage a portfolio of sub-projects within an overall SOA business strategy
- Project Team Members, who need to map dependencies and develop a timeline that meets the business expectations
- Vendors who provide solutions and tools for new business capabilities to the business and IT
- Standards bodies which need a better understanding of use cases of how business and IT plan to leverage technology to meet their objectives.

Services Lifecycle Stages

1.1.1 Actors

- Project manager
- Architects
- Release management
- Build teams
- IT operations

1.1.2 Tools Used

Tools used include product deployment tools such as Maven, Cruise Control, Ant, shell scripts, and IDE-based tools such as Eclipse, Visual Studio, and packaged application deployment tools.

1.1.3 Artifacts (Deliverables)

Chief deliverables are the product domain model, packed application configuration, builds, and service dependencies.

1.1.3.1 Artifact Description

The primary output of this process is the product domain model which teams can leverage to assemble the builds for each of the services (if required), assemble all the metadata for products, and map the URLs that are physical end-points for running the services. Teams develop the physical deployment model from the logical model (SCA). Once the team has assembled the services for deployment, it needs to create the network topology, server configuration, service configuration, and server and server asset management. In addition, the team must also create runtime artifacts such as monitoring and event logs for correlation later downstream.

1.1.4 Service Lifecycle Stage Key Considerations

There are two key considerations for the stage of the service lifecycle: service deployment, and service managing and monitoring.

1.1.4.1 Service Deployment

Composite applications consist of a service or portfolio of services that can be implemented on various technologies. Release management requires planning builds at a service level and also providing the capability to assemble services for a target node. In addition, IT operations needs to put together plans and processes for modifying service and instance configuration parameters while assembling or provisioning the service.

IT operations might require a service assembly management server to help manage the various builds, assemble the services for target nodes, and manage assets. There are products in the market that provide this capability, but unfortunately there aren't any standards on service packaging or service provision. This is potentially an area for the SCA standards to address.

1.1.4.2 Service Management and Monitoring

Service management and monitoring can be sub-classified into two more areas:

- **Server management:** configuring the deployed services and server instances, ideally with a universal administration console. This should be consistent across all products and all the vendors if possible.
- **Server monitoring:** applying the monitoring policies to capture the required matrix for each of the services. This matrix should be based on the KPIs and management policies defined during the requirements and analysis stage of the services lifecycle. Teams also need to collect and correlate monitoring events to ensure the business service level agreements (SLAs) are met and also to publish the aggregated management summary information to the service registry and/or SOA repository.

1.1.5 Service Lifecycle Stage Recommended Process

This phase begins whenever the development team is ready to conduct integration testing, user acceptance testing, or performance testing, or is ready to deploy the application to the production environment. IT operations should follow these steps:

1. The development team completes the development and modification of services and performs unit-level testing as well as integration testing using the simulation capability.
2. The development team identifies the services and assembles them into a project that is ready for testing or deployment to the release management team.
3. The build or release management team uses product tools to assemble builds for deployment. These will prompt the user to enter all the physical end points and application parameters. In the testing environment, the user should be able to enter the URLs of the test services already running in the QA environment or point to the simulated services. When services are ready for deployment or production, users enter the URL for the services.
4. If the service is being decommissioned, the team creates a new project to decommission the service and redesign all services dependent on it. Sometimes, not all dependent services are defined in the SOA repository. Following is a process to mitigate risk.
 - a. Create a project to redesign and deploy all services dependent on the service to be decommissioned.
 - b. Keep the service targeted for decommission up and running and monitor it to confirm that it is not being consumed by any other service.
 - c. Deploy the project and create a new project to decommission the old service.
5. IT operations should use the deployment tool to assemble the services project for deployment.
6. Once IT operations deploys the service into production, the team can leverage various tools to monitor the service, application, middleware, OS, hardware, and network.
7. IT operations is responsible for keeping the services up and running. IT operations escalates support problems caused by application-specific issues to the application support team.

Contributing SOA Practitioners

Surekha Durvasula, Enterprise Architect, Kohls

Martin Guttman, Principal Architect, Customer Solutions Group, Intel Corp

Ashok Kumar, Manager, SOA Architecture, Avis/Budget

Jeffery Lamb, Enterprise Architect, Wells Fargo

Tom Mitchell, Lead Technical Architect, Wells Fargo Private Client Services

Burc Oral, Individual Contributor

Yogish Pai, Chief Architect AquaLogic Composer, BEA Systems, Inc.

Tom Sedlack, Enterprise Architecture & Engineering, SunTrust Banks, Inc.

Dr Harsh Sharma, Senior Information Architect, MetLife

Sankar Ram Sundaresan, Chief Architect e-Business, HP-IT