

SOA governance: Examples of service life cycle management processes

Skill Level: Intermediate

[Prabhakar Mynampati \(pmynampa@in.ibm.com\)](mailto:pmynampa@in.ibm.com)
Advisory Architect
IBM

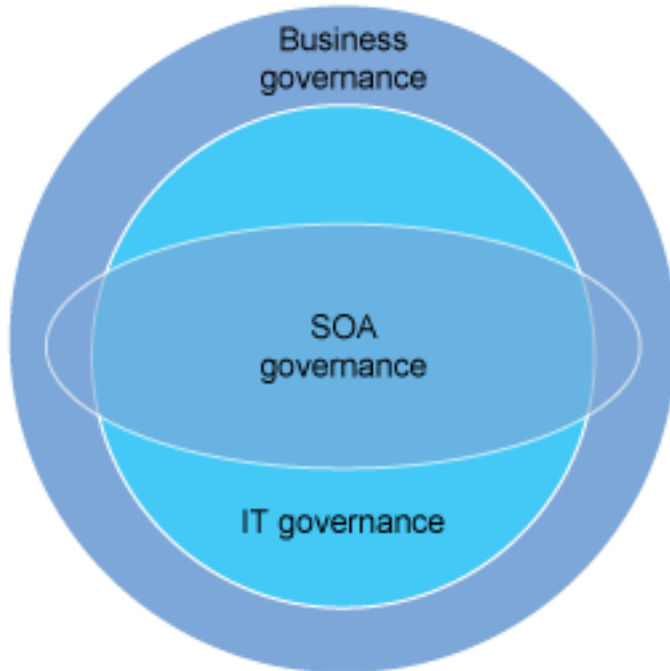
06 Nov 2008

You need an appropriate governance process model in place to effectively realize the benefits of Service-Oriented Architecture (SOA) development life-cycle activities. This article describes SOA governance processes based on scenarios that are adopted in a typical enterprise during the SOA development life cycle. Take a close look at important life-cycle activities, such as service identification, service creation and reuse, service testing, service versioning and change management, service-level management (quality of service), and service security. Learn about the challenges that your organization might face in a typical SOA development life cycle and how to address these challenges by implementing governance subprocesses and by delegating certain roles and responsibilities to the respective layers of a governance body for each scenario.

What is SOA governance?

SOA governance is the intersection of business and IT governance focused on the life cycle of services to ensure the business value of SOA. It's the effective management of this life cycle that is the key goal to SOA governance.

Figure 1. The definition of SOA governance



IBM's approach: SOA Governance and Management Method

IBM's approach to governance includes two distinct aspects for success: definition and enforcement. The SOA Governance and Management Method (SGMM) is the end-to-end approach to definition through designing, implementing, and improving SOA governance. SGMM gives you a prescriptive approach to identifying all the elements needed to ensure proper governance, including identifying new roles and responsibilities, defining policies and metrics, and placing checkpoints inside the processes that IBM follows. During enforcement IBM distributes those policies among the different enforcement points, and monitors and measures the related metrics to improve the governance model.

Governance scenarios in the SOA development life cycle

In addition to the SOA development life cycle described above, IBM also encourages using the SOA foundation life cycle. The SOA foundation life cycle starts with modeling business, translating the model into an information system, deploying that information system, and then managing that deployment. As the SOA development life cycle closely follows IBM's SOA foundation life cycle, a typical service development happens through the model, assemble, deploy, and manage phases. Let's break these down into more detail:

Model

The **model** phase is based on business requirements, and in it the business process

models are designed, business services are identified, processes are simulated with what-if business conditions, and their subsequent results are analyzed with respect to business objectives. If the objectives are not met, processes are redefined with modified business services. From these modeling aspects, identification of business services is an important activity, which is one-to-one mapping to an IT service realization.

Assemble

After defining the IT service, you come to the **assemble** phase for design, specification, creation, and testing. The constructed services are tested independently and with the composition of business processes. The tested and qualified services are brought into a runtime repository and get published for service consumers. As different service clients consume these services with little changes in functional aspects, different versions of the same services with varied functionality are built after change process approvals.

Deploy

In the **deploy** phase, different versions of services are published in repositories and deployed in service containers with an integrated business processes. The deployed services can also be dynamically selected at run time to build composite applications based on some selection criteria from the service requesters.

Manage

These services are monitored during the **manage** phase for timely responsiveness for different service requests, detecting failures at service levels, and restoring the operational state of the system. During this phase, services and the overall operational environment are tuned to meet business objectives. The services are managed for identity and compliance through proper service-level security design and implementation.

In this SOA development life cycle, the following challenges are likely to be encountered in the absence of clearly defined and enforced SOA governance processes:

- Struggling to identify new services and prioritize them.
- Significant issues with service creation and reuse, such as creation of redundant and inefficient services.
- Adoption test strategies and standards are haphazard.
- Governance of changes and versions for services are crude and unrefined

- No systematic way to enforce governance policies for service management, quality of service (QoS), and services security.

These challenges have led to requirements of defining and enforcing the following processes as part of the SOA governance model:

- Govern how new services are identified and prioritized (service identification).
- Govern how new services are created and reused (service creation).
- Enforce test procedures and strategies for the best realization of services (service testing).
- Enforce governance policies for service level management, or SLM (service level management).
- Govern change management and versioning of services (service versioning change management).
- Enforce security standards and policies for service run time (service security).

Service identification example

Any organization continuously increases new services with the change in requirements. Service identification is the first step in the SOA development life cycle. There are challenges related to extending the current IT governance model to embrace these new services and to understand the roles involved.

Issues

There are a variety of project risks due to inconsistent approaches adopted in the identification of business and IT services. Services might be noninteroperable, and after the identification of services many of them might be redundant. There might also be a situation where there's no responsible owner of the identification and delivery of services. Ultimately, all these risks can result in increased project costs and the inability to meet delivery time. But you can resolve these issues by applying the governance process, as shown in [Figure 2](#).

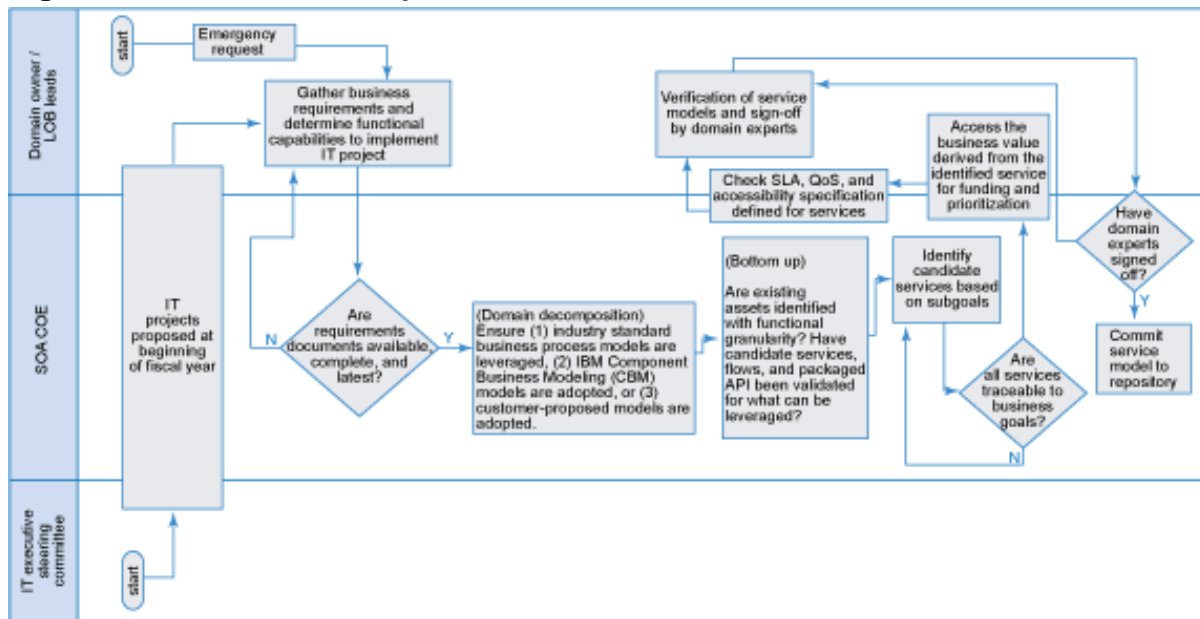
Governance subprocess for service identification

In this governance process scenario, the participants involved in defining policies and executing the governance process are members of the IT executive steering committee; they include members of the internal SOA Center of Excellence (COE) and business domain owners or line of business (LOB) leads. The IT executive steering committee is empowered to define the policies, and in this scenario, they

aim to:

- Build 70% of services through existing operational systems and only 30% as new services.
- Make identified services visible throughout the enterprise for reuse.

Figure 2. Governance subprocess for service identification



The governance process has to be defined to make sure that the following requirements are met:

- Adopt the right approach in identification of services by using both domain decomposition business process models and bottom-up models.
- To start this governance process, the request is initiated either from the steering committee board as IT projects are proposed at the beginning of the fiscal year or as an emergency request initiated from domain owners.
- The SOA COE should ensure that all business requirements captured by the LOB leads are complete and are the most current before service identification starts. This ensures the Service-Oriented Modeling and Architecture (SOMA) steps are followed and that all the identified services are traceable and lead to business goals.
- The identified service models are propagated to the domain experts for their verification and sign-off.
- The approved models are published in a common repository to avoid identification of redundant services.

- The process clearly identifies the domain owners who should own the responsibility of services.
- This governance process drives better requirements to services and reuse at design time.

Service creation example

During the service creation scenario, after identifying the required services, the services' low-level specifications are defined and implemented. The governance model may not perform these tasks, but it makes sure that the tasks are being performed. The COE coordinates the teams that are creating and requiring services to make sure needs are being met and to avoid duplicating efforts. The governance model identifies owners who create such services and determines the priority of implementation of these services.

Issues

Currently, an organization is suffering from the development and deployment of redundant and inefficient services built across different lines of horizontal and vertical domains without regard for other repositories. Some of the services are getting realized in an inconsistent way of recognizing system functions. The maintenance costs are increasing because multiple copies of similar or the same services are being maintained, and there's no control over such services, which halts further development. All these issues ultimately demand regulation of the service creation process.

Governance subprocess for service creation

The governance process has been defined with the following steps, which are shown in [Figure 3](#):

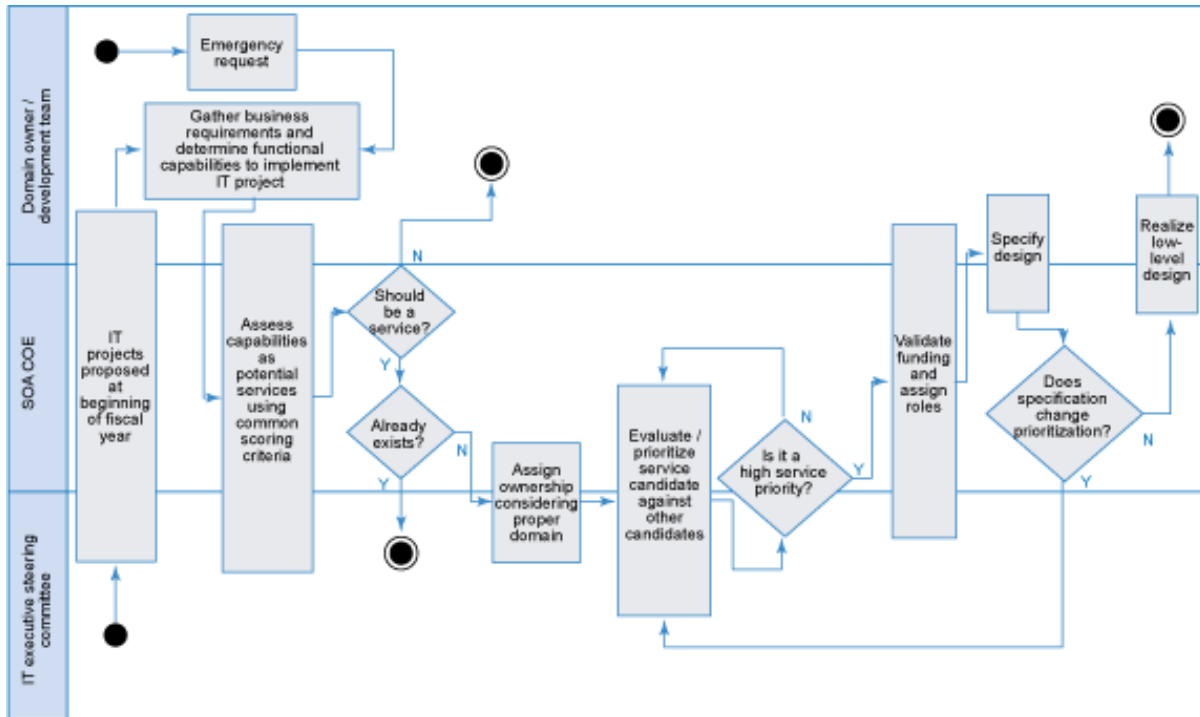
- While creating the new governance process definition, keep in mind the current IT planning process that reviews project candidates and decides which projects to pursue, and understand the role of ownership in various service domains that are accountable for those services. You need to understand the funding process and define a comprehensive approach for QoS as defined in the service specification phase.
- The governance process starts with a project by assessing capabilities as potential services using common scoring criteria, such as reuse, business agility, and deployment flexibility by the SOA COE.
- After deciding that it's to be made as a service, search through different service registries and ensure that a similar service doesn't already exist. If

the service doesn't exist, the SOA COE assigns ownership to that service based on the business domain to which it belongs.

- The commonly used infrastructure and technical services ownership is retained by the SOA COE.
- The SOA COE prioritizes the service against other services. If the COE decides that the service is high-priority, based on the domain ownership the service belongs to, then the domain owner allots the funding.
- The service owner engages the proper resources in designing and developing the service with the help of those with specialized skills in the SOA COE.
- The identified service models are propagated to the domain experts for their verification and sign-off.
- The SOA COE architects are responsible for defining interfaces, message flows, nonfunctional requirements (NFRs), and lower-level functional requirements.
- After completing the specification of the service, the SOA COE again looks at the service for prioritization for further realization.
- In the realization phase, the SOA COE architecture team helps build low-level design of service, and maps the service to technical tooling, implementation, and so forth for the vertical domain development team.
- The SOA COE further ensures that services are adhering to standards and best practices, thereby reducing risks across deployment and management.

(See a larger version of [Figure 3](#).)

Figure 3. Governance subprocess for service creation



Service testing example

In this scenario, the realized services are tested for their required functionality and their suitability for integration and system testing. After qualification at both the integration and system test levels, the services are passed through NFR tests, such as a response-time test, resource usage test, transactional throughput test, and so on.

Issues

In this scenario, you look at a situation where a variety of groups are using different tools and test strategies for testing their services. There's no uniform approach for using tools, plug-ins, and test strategies for realization of services in the organization. This is attributable to the fact that compliance tests among some of the services were developed in vertical units, and some issues were raised that business requirements aren't correctly met with realized IT services. Some units aren't able to meet integration and system-testing deadlines in the stringent project schedule. And few project teams have trouble handling change in business requirements to systematic realization of IT services. All these issues have brought challenges to governance in a testing scenario.

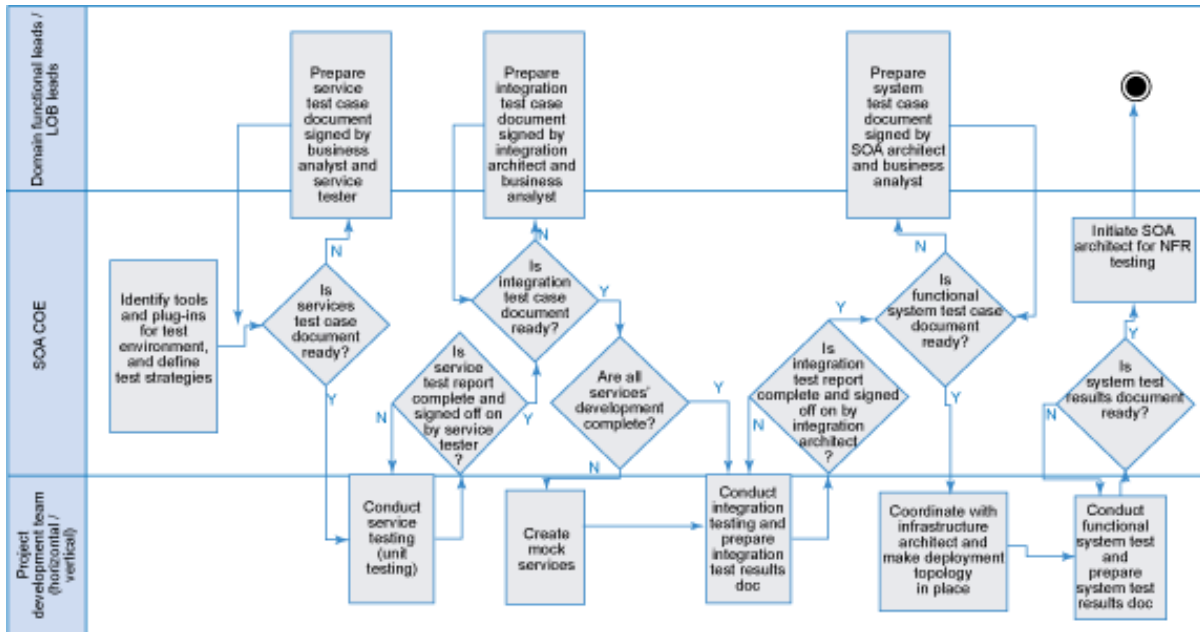
Governance subprocess for service testing

The detailed process has been outlined in [Figure 4](#) and should include the following steps:

- The SOA COE takes ownership of identifying functional and nonfunctional testing tools and plug-ins for the test environment, and defines test strategies based on project requirements.
- The SOA COE enforces and promotes these tools and test strategies to the other business units.
- The governance processes for a project starts with SOA COE involvement with the respective domain business analyst in preparing a service test case document and gets signed off by the analyst.
- The COE initiates the service unit testing with the identified development teams of that project or the resources identified by service ownership.
- The process looks for completion of a unit test report generated by the SOA tester and then initiates a request for service integration testing.
- An integration architect of the SOA COE is involved in preparing the integration test case document and gets sign-off from the business analyst.
- The members of the SOA COE look for the readiness of all services required for integration, developed either from various teams within a business unit or across business units.
- If any of the services required for integration aren't available and the service is still in the build process, thus causing delay, members of the COE need to decide to go with a mock service and initiate the integration testing.
- Similar guidelines are followed for system testing initiation and actively interact with the infrastructure architect in identifying correct versions and configurations of software and hardware required for system testing.
- After completing the system test report, the COE initiates NFR testing.

(See a larger version of [Figure 4.](#))

Figure 4. Governance subprocess for service testing



Service versioning and change management example

After releasing new services into production, the users of those services start needing changes. Bugs need to be fixed, interfaces need to be redesigned, new functionalities need to be added, and unnecessary functions need to be removed. Service versioning helps to run the system stable without disruption to existing proven service functions and lets you add new functionalities to those services.

Issues

Some of the business requirement changes don't necessarily need to be implemented in the IT infrastructure. In this scenario, however, there's no authority in the organization to decide whether required changes in the business processes needed to be implemented in IT and whether that change should be implemented in an existing service or as a next-version release. There's no common body in the enterprise to study and know the impact of these changes with respect to other service consumers. And there's no authority to decide runtime policy for versioning of services. System nonavailability complaints from the customers are arising due to production disruption when changing service versions. There's no rule-based automatic selection of service providers for smooth transition. There's no tracking mechanism to know which services are running in the system, which are actually used, and which are retired from the system. All these issues have forced an implementation of service versioning governance.

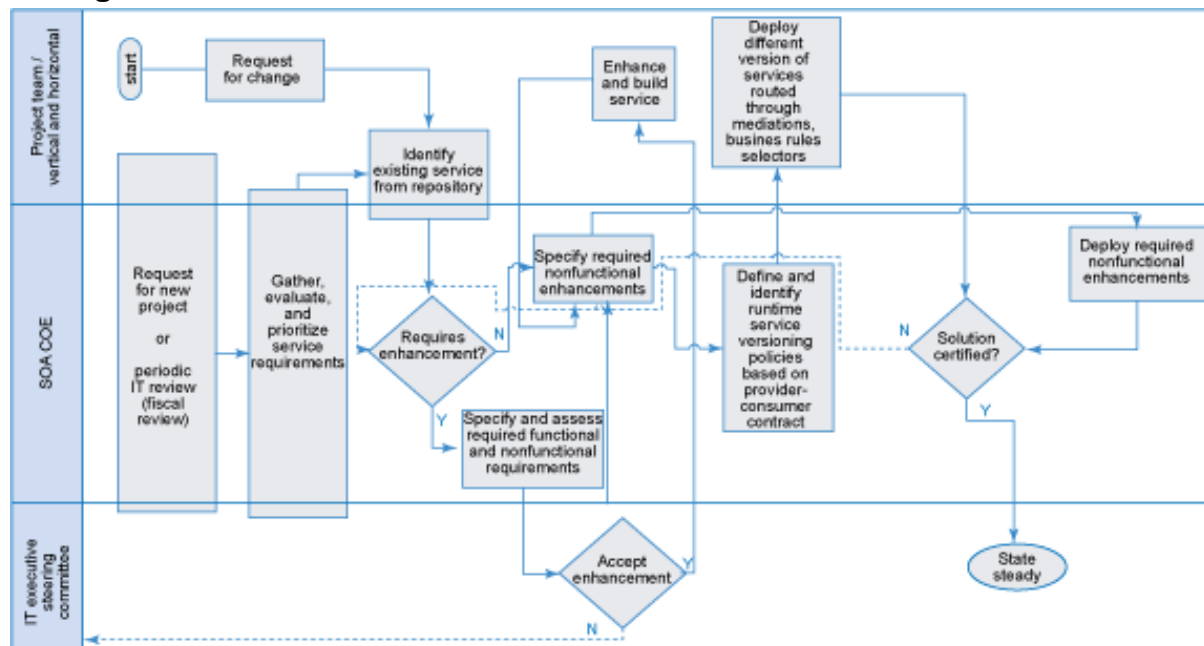
Governance subprocess for service versioning and change management

The detailed process is defined in [Figure 5](#) and should have the following steps:

- In this process, the IT executive steering committee plays an important role in the governance body. It's an empowered body that can grant changes in IT for business requirement enhancements.
- The body also decides which base version of services can be enhanced and which changes can be released in the existing version of service or as a next version. The inputs for the IT steering committee are driven by the SOA COE, which in turn gets service version details from project teams or vertical units.
- The SOA COE studies the changes of impact to the existing service with respect to other service consumers and, hence, puts forward the impact details to the steering committee.
- The SOA COE studies the existing changes, keeping NFRs in view. But if the change is an NFR enhancement, then you need to define and identify runtime service versioning policies based on the provider-consumer contract.
- Then the COE identifies the corresponding project teams to build and deploy different versions of services routed through mediators, business rules, selectors, and business dynamic assemblers.

(See a larger version of [Figure 5.](#))

Figure 5. Governance subprocess for service versioning and change management



Service management example

Service monitoring and management is an important activity after the services are moved into production. If any of the services stop working, the system should identify and detect the problems faced by such services. Service monitoring helps the system architects to detect and prevent problems before they occur. It can detect load imbalances and outages, providing warning before they become critical, and can even attempt to correct problems automatically.

Issues

In this service management scenario, you look at an organization that's facing difficulties in monitoring and managing the services that are exposed to different service consumers. This increases the dependency on service providers. The services weren't able to meet expectations of high availability, performance metrics, and defined service level agreements due to poorly designed service management architecture. The architecture and development team is not aware of the services and resources that need to be monitored based on service level agreements (SLAs). There's no uniform approach for adoption of management tooling that covers the end-to-end view of business application, and there's no uniform approach for providing detailed information about performance and availability metrics for individual resources. All these issues have forced an implementation of service management governance.

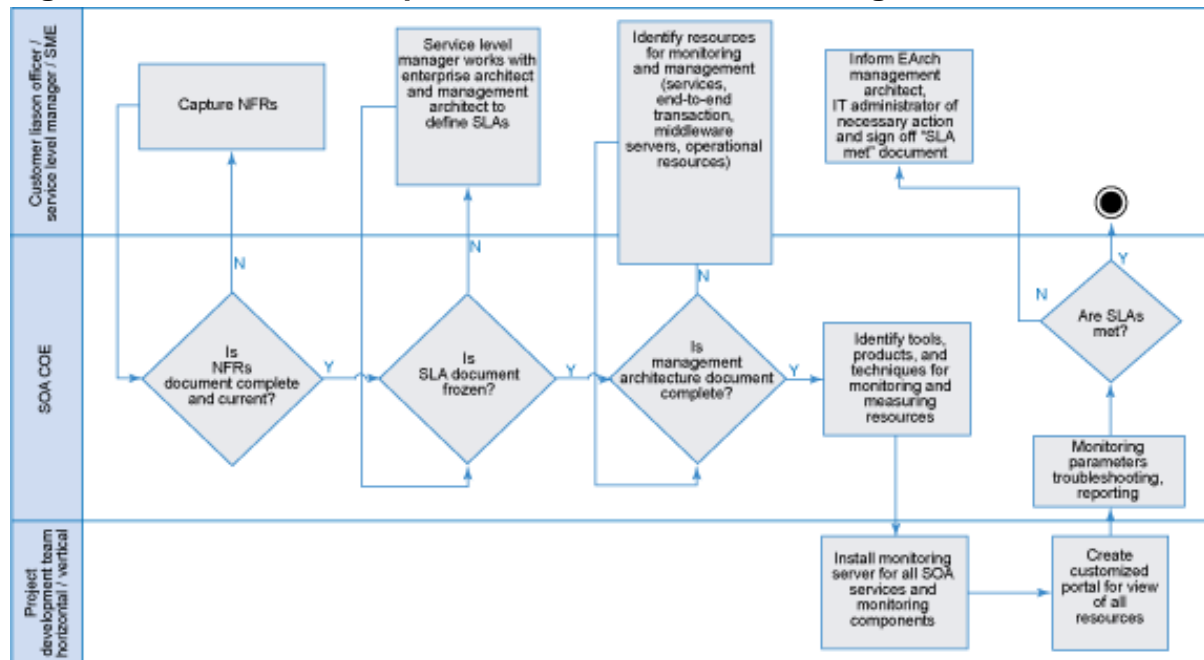
Governance subprocess for service management

The service management governance process should follow these basic steps (also represented in [Figure 6](#)):

- The SOA COE works with the business team to capture the NFRs from the customer.
- The SOA COE interacts with the LOB service-level manager and management architect in defining SLAs during the model phase of the SOA foundation life cycle.
- The SOA COE core team analyses these SLAs and designs the management architecture document. A management architect, IT administrator, subject matter expert, and application developer help identify the resources to manage the composite application.
- Members of the SOA COE also identify the management development tools that support the management architecture.
- With the help of the IT administrator and respective vertical unit service providers, the COE ensures that a common monitoring console is developed for monitoring both business and IT performance resources.

(See a larger version of [Figure 6](#).)

Figure 6. Governance subprocess for service level management



Service security example

SOA security helps restrict user access to services. It protects the data being exchanged between service providers and service consumers. Even among authorized users, not all users should have access to all data that the service has access to. Access to services has to be controlled and limited to authorized consumers. User identity must be propagated into services and used to authorize data access. Qualities of data protection have to be represented as policies within ranges. This enables consumers to express minimal levels of protection and maximum capabilities, and to be matched with appropriate providers who may, in fact, include additional protections.

Issues

In this scenario, an organization has no common strategy for attacking security threats and protecting services from external access. Multiple authentication and authorization of services are frustrating operators. There's no security policy management framework in place for adopting policies and tracking these policies from inception to implementation, and there's no responsible body for interacting and communicating with enterprise boundary organizations to maintain a common set of security standards. There are no agreed-upon security standards for interacting with other services and data, and no defined roles and responsibilities for policy administration. These issues have led to the organization needing to define and adopt service security governance processes, shown in [Figure 7](#).

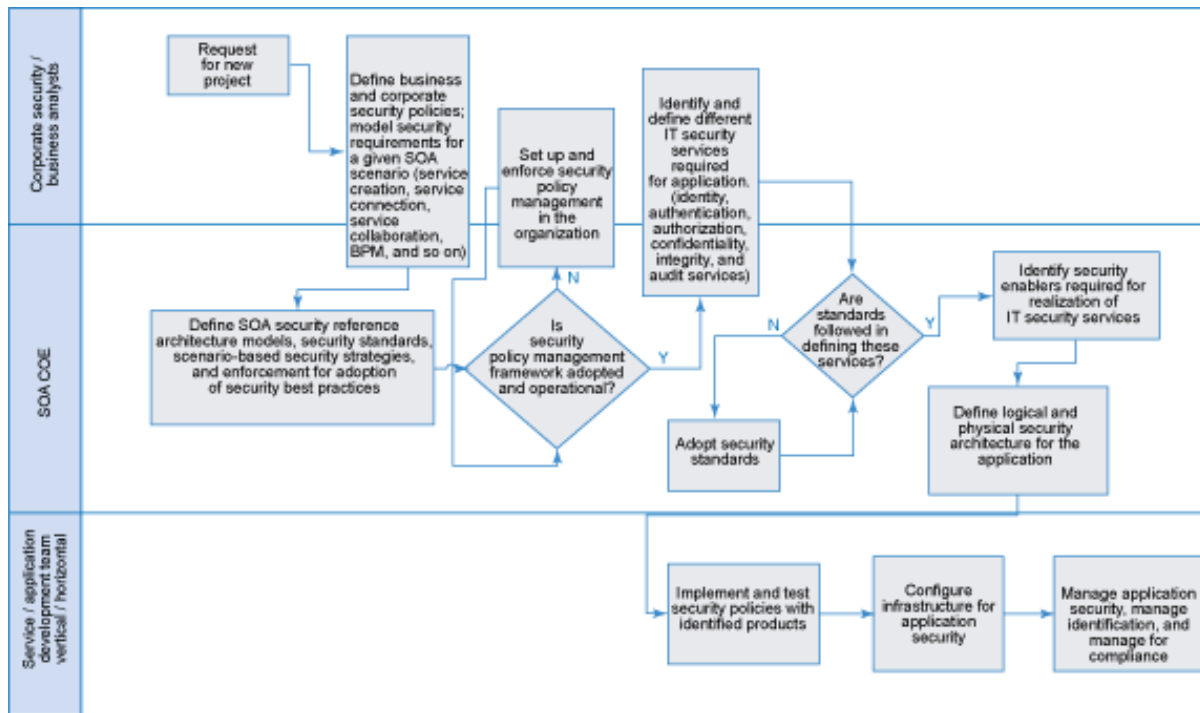
Governance subprocess for service security

The governance process should cover the following steps:

- In the model phase of service development, the SOA COE collaborates with business analysts and corporate security analysts in modeling security requirements for given SOA scenarios, such as service creation, service connection, service collaboration, and business process management, because security implementation is widely varied from scenario to scenario.
- The COE has built-in competency in developing SOA security reference architectures, security standards, scenario-based security implementation strategies, and best practices.
- The SOA COE, with the help of business analysts, identifies different IT security services required for implementation for a project and checks whether open standards are followed in implementing these services.
- The COE identifies various security enablers, such as cryptography technologies; registries and repositories used for storing user information; hardware keys; and firewalls for realization of IT security services.
- The COE reuses and shares the defined logical and physical architectures based on scenarios to the project teams for realization.
- The COE identifies various tools and products for implementing the security policies with the help of application developers of different vertical units.

(See a larger version of [Figure 7.](#))

Figure 7. Service security



Conclusion

This article identified some of the key scenarios that provide value in terms of SOA governance and how to mitigate the risk involved in the key scenarios. All the governance scenarios defined in this article may be tailored as required, because it might vary from organization structure, set up, and the context required for that process.

Acknowledgments

I would like to thank John Falkl and his team, Amar Raiker and Yaseen M Yaseen, for their support and contributions to the success of this article.

Resources

Learn

- Get more information on structure of SOA governance body from the article "[A case for SOA governance](#)" (developerWorks, Aug 2005).
- Read Bobby Woolf's article "[Introduction to SOA governance](#)" (developerWorks, Jun 2006).
- The [SOA and Web services zone](#) on IBM developerWorks hosts hundreds of informative articles and introductory, intermediate, and advanced tutorials on how to develop Web services applications.
- Play in the [IBM SOA Sandbox!](#) Increase your SOA skills through practical, hands-on experience with the IBM SOA entry points.
- The [IBM SOA Web site](#) offers an overview of SOA and how IBM can help you get there.
- Stay current with [developerWorks technical events and webcasts](#).
- Browse for books on these and other technical topics at the [Safari bookstore](#).
- Check out a quick [Web services on demand demo](#).

Get products and technologies

- Download [IBM product evaluation versions](#) and get your hands on application development tools and middleware products from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.

Discuss

- Get involved in the developerWorks community by participating in [developerWorks blogs](#).

About the author

Prabhakar Mynampati

Prabhakar Mynampati is an SOA architect working with BPTSE, IBM Software Labs, India. He assists IBM's ISV clients in building asset-intensive solutions using IBM SOA, CBS reference architectures.

Trademarks

IBM, the IBM logo, ibm.com, DB2, developerWorks, Lotus, Rational, Tivoli, and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. See the current list of [IBM trademarks](#).

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.