Effective governance through the IBM SOA Governance Management Method approach
White paper
August 2006

Rational® software

SOA governance: how to oversee successful implementation through proven best practices and methods.

William A. Brown, IT Architect
IBM SOA Center of Excellence

Murray Cantor, PhD, Distinguished Engineer
IBM Rational Software

Contents

Introduction

Service orientation

IBM defines service orientation as a way of integrating your business as a set of linked services. A service-oriented architecture (SOA) revolves around the key concept of a "service."

Services

A service encapsulates a set of resources that can execute a repeatable task within a process. Services are described by their behavior and interfaces. The service consumer rarely has visibility to the resources; rather, the consumer invokes the service through the service interface. For example, when you use an online banking service, you have access to the bank's IT system without knowing the implementation details behind the service.

SOA

IBM defines SOA in terms of service orientation. An SOA, then, is an architectural style suitable for an enterprise IT architecture that exploits the principles of service orientation to achieve a tighter relationship between the business and the information systems that support the business. SOA is a paradigm for organizing and using distributed capabilities that may be under the control of different ownership domains. It is natural in such a context to think of one person's needs being met by another person's capabilities; or, in the world of distributed computing, one computer agent's requirements being met by a computer agent belonging to a different owner. There is not necessarily a one-to-one correlation between needs and capabilities; the granularity of needs and capabilities varies from fundamental to complex. And any given need may require the combination of numerous capabilities, while any single capability may address more than one need. The perceived value of SOA is that it provides a powerful framework for matching needs and capabilities and for combining capabilities to address those needs.

For a more complete discussion of these definitions and related concepts, visit:

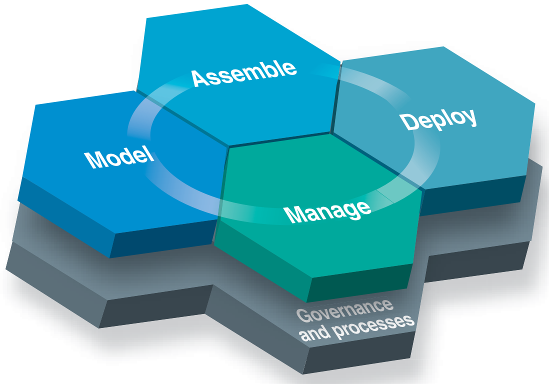
ibm.com/developerworks/webservices/library/ws-soa-whitepaper

The *IBM SOA foundation: an architectural introduction and overview* white paper will provide more information.

The intent of the SOA is to achieve better alignment between the business world and the IT world—in a way that makes both more agile. A key enabler of this improved responsiveness and flexibility is the use of services to rapidly build composite applications.

SOA lifecycle and processes

The IBM SOA foundation white paper mentioned above defines a four-phase services lifecycle, as shown in figure 1.



⬢
- Gather requirements
- Model and simulate
- Design

⬢
- Discover
- Construct and test
- Compose

⬢
- Integrate people
- Integrate processes
- Manage and integrate information

⬢
- Manage applications and services
- Manage identity and compliance
- Monitor business metrics

⬢
- Financial transparency
- Business/ IT alignment
- Process control

Figure 1: The services lifecycle

According to the white paper, there are four phases of the services lifecycle. They are described as follows:

- *The Model phase includes the process of capturing your business design from an understanding of business requirements and objectives, and translating that design into a specification of business processes, goals and assumptions—creating an encoded model of your business.*
- *The Assemble phase concerns the information system artifacts that will be used to assemble the business design. The enterprise architect working with the business analyst can begin to convert the business design into a set of business process definitions and activities, deriving the required services from the activity definitions.*
- *The Deploy phase includes a combination of creating the hosting environment for service-based (composite) applications and the actual deployment of those applications. This includes resolving the application's resource dependencies, operational conditions, capacity requirements, and integrity and access constraints.*
- *The Manage phase addresses the maintenance of the operational environment and the policies realized by the deployed SOA applications. This phase includes monitoring performance of service requests and timeliness of service responses; maintaining problem logs to detect failures in various system components; detecting and localizing those failures; routing work around them; recovering work affected by those failures; correcting problems; and restoring the operational state of the system. The Manage phase also includes managing the business model—tuning the operational environment to meet the updated business design.*

Rationale for SOA governance

We can generalize the benefits of SOA and their value proposition; however, it should be recognized that—for SOA benefits to be achieved and sustained—we often must be more specific about the SOA motivations and the audience those benefits will affect. For example, improved integration with SOA largely benefits IT; whereas business process flexibility primarily benefits the business. But applying SOA to improve productivity and time to market for new products through reuse has direct visible benefits to both IT and the business.

Service orientation alone cannot bring about the effect of improved productivity, faster time to market and reuse because there are other forces at play that operate as impediments and constraints. Effective governance of services through policies, principles, standards, procedures, processes, and cultural and organizational change will enable the full benefit of service orientation to be realized. This governance model needs to establish chains of responsibilities, decision-making rights and authority, and communication to empower people to bring about the necessary changes for service orientation and to sustain those changes so that the organization does not default to business as usual. This effective governance is what's called "SOA governance."

To understand SOA governance, we must put it in the proper context. We need to have a base understanding of what governance is, and then a base understanding of what IT governance is.

Governance defined

Governance is:

- *Establishing chains of responsibility, authority and communication to empower people (decision rights)*
- *Establishing measurement, policy and control mechanisms to enable people to carry out their roles and responsibilities*

Governance, then, is assigning the rights to make the decisions and deciding what measures to use and what policies to follow to make those decisions. The decision rights are assigned to roles in the organization, not to individuals. So, an aspect of governance is determining organizational roles.

Corporate governance, or enterprise governance, establishes the rules and the manner in which an enterprise conducts business, based upon its strategy, marketplace and principles of doing business. It defines for employees and for business associates the processes that are used to conduct operations and the manner in which people interact.

Beginning with the board of directors and cascading throughout the organization, there are many aspects and levels of corporate governance. All aspects of the business are touched in some manner, from the human resources department to purchasing and marketing.

Part of any governance solution is meeting the organization's compliance requirements. Compliance is documenting and proving that governance is in place and is being executed, i.e., the decisions are documented and the decision policies are followed. The compliance specifics depend on the client's needs. In the United States, client needs are likely to include compliance with Sarbanes-Oxley regulations, but they may also include domain-specific standards such a CFR-11 for the pharmaceutical industry or Basel II for banking. The compliance requirements should be determined early in the SOA governance and management plan phase and addressed throughout the lifecycle.

IT governance defined

IT governance refers to the aspects of governance that pertain to an organization's information technology processes and the way those processes support the goals of the business. IT governance represents a significant part of enterprise governance, and—given the horizontal nature of IT, where almost everyone in the enterprise uses IT assets to complete their responsibilities—it is also the most visible part of enterprise governance. Effective and ineffective IT governance can be assessed by employees rather easily.

IT governance defines a structure of relationships and processes to direct and control the enterprise. It helps to achieve the enterprise's goals by adding value while balancing risk and return regarding IT and IT processes. IT governance is a subset of enterprise governance; it deals with the management and control of IT assets, people, processes and infrastructures, as well as the manner in which the assets are managed and procured. IT governance also helps to define roles and responsibilities and to specify the decision rights and accountability framework that will help to encourage desirable behavior in IT departments and establish accountability for the use of IT assets. IT governance also helps to codify best practices and define monitoring practices.

IT governance may be characterized by assigning decision rights and measures to processes, including, but not limited to, those defined by Control Objectives for Information and related Technology (COBIT) or Information Technology Infrastructure Library (ITIL). The text, *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results,* by Peter Weill and Jeanne Ross (Harvard Business School Press, June 2004) provides a framework for setting decision rights in IT organizations.

SOA governance defined

SOA governance is an extension of IT governance specifically focused on the lifecycle of services, metadata and composite applications in an organization's service-oriented architecture. SOA governance defines the changes to IT governance to ensure that the concepts and principles for service orientation and its distributed architecture are managed appropriately and are able to deliver on the stated business goals for services.

As a specialization of IT governance, SOA governance addresses how an organization's IT governance decision rights, policies and measures need to be modified and augmented for a successful adoption of SOA, thus forming an effective SOA governance model.

Since SOA is a distributed approach to architecture that crosses lines of business and IT, there is a greater need for effective SOA governance. In addition, SOA governance provides a framework for the reuse and sharing of services, a key value derived from leveraging SOA.

Because of its cross-functional aspects, SOA governance also provides a framework for examining several items necessary to manage services as another type of IT asset, such as:

- *Maturity of service orientation within the enterprise*
- *Infrastructure enhancements for managing the usage of services in areas of security, monitoring, performance, versioning and shared usage*
- *Enhancements to IT processes to address funding, sharing, incentives for sharing, and reuse of services, as well as for the identification, design and specification of services*
- *Education and training*
- *Roles and responsibilities*
- *Organizational changes*

SOA governance extends IT governance by assigning decision rights, policies and measures around the services, processes and lifecycle of SOA to address such concerns as:

- *Service registration*
- *Service versioning*
- *Service ownership*
- *Service funding*
- *Service monitoring*
- *Service auditing*
- *Service diagnostics*
- *Service identification*
- *Service modeling*
- *Service publishing*
- *Service discovery*
- *Service development*
- *Service consumption*
- *Service provisioning*
- *Access to services*
- *Deployment of services and composite applications*
- *Security for services*

The agility provided by SOA also provides a set of challenges to the IT organization. Undisciplined development and deployment of services can do more harm than good. Hence, organizations must address a set of key questions if they are to obtain value from SOA:

1. How do we regulate the deployment of composite applications?

2. What organizational change is required? What new organizational roles and structures will facilitate service identification, design and sharing?

3. How will we organize the IT function to build and leverage service-oriented capabilities?

4. What metrics will support investment, maintenance, vitality and sharing of services?

5. How will we decide to invest in service creation and maintenance?

The value of SOA governance

The distributed nature of services across various lines of business forces an enterprise that's looking for an SOA transformation to take governance more seriously. The proliferation of moving parts (i.e., building blocks in the form of services) that need to be maintained by different organizations both within and outside the enterprise makes governance more challenging. This cross-organizational nature of business services and the potential composition of services across organizational boundaries can function properly and efficiently if, and only if, the services are effectively governed for compliance to service level agreements (SLAs) and nonfunctional requirements (NFRs; e.g., security, reliability, performance). Thus, identifying, specifying, creating and then deploying enterprise services require strong service-oriented governance through a very strong, efficient body that can oversee the entire lifecycle of an enterprise's service portfolio. Policy rules, decisions and enforcement, together with the requirement of well-organized service lifecycle management, raise the implementation of SOA governance to the strategy and planning phases of an enterprise SOA transformation, rather than keeping it as an afterthought implementation.

During the first wave of SOA, governance was merely a nice-to-have discipline. But with the growing maturity and complexity of SOA implementations at real-world enterprises that have complex and integrated value chains, SOA governance enjoys a mandatory presence in the overall SOA transition strategy.

An enterprise that fails to realize the importance of an effective governance structure may not stand to benefit much from an SOA transition. In fact, an SOA implementation at such enterprises may prove to be disruptive, since these companies may lack the proper organizational structure to effectively follow SOA principles and reap the architecture's ultimate benefits.

Implementing governance

As discussed above, adopting SOA entails installing new processes, decision matrices, policies, principles, measures, organizational models and cultures, and/or possibly modifying these components from the current environment. Briefly then, to apply our definition of SOA governance, the governance team needs to:

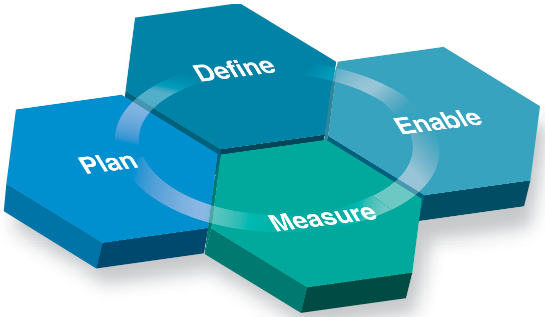
- *Understand the existing IT governance model. This includes processes, decision matrices, policies, principles, measures, organization and culture.*
- *Identify what can be reused from the existing model and what processes will be affected by adopting SOA.*
- *Design new additions to the model or select additions from leading practices.*
- *Understand the impact the additions and modifications will have.*
- *Make additions to or modify the existing model.*
- *Create a prioritized transition plan to transform the model with minimal disruption of operation.*
- *Communicate this planned transition to all involved.*
- *Implement the agreed-upon additions and modifications.*
- *Continuously measure the effectiveness of the model in place.*

Applying governance haphazardly will not result in the desired business benefits and will increase the likelihood of failure for the SOA implementation.

The SOA governance lifecycle

SOA governance is setting decision rights and measures for those who execute SOA processes. However, executing governance intentionally meets our definition of process: It requires a sequence of events through which the rights and measures are specified. The governance process enables organizations to make considered decisions about the structure and rigor of executing governed processes.

The process of governance has its own lifecycle, which is distinct from the process lifecycle being governed (see figure 2).



Plan the governance requirements
- Document and validate business strategy for SOA capabilities
- Assess current IT and SOA capabilities
- Define or refine vision and strategy
- Review current governance capabilities and arrangements
- Lay out governance plan

Define the governance approach
- Define or modify governance processes
- Design policies and enforcement mechanisms
- Identify success factors and metrics
- Identify owners and funding model
- Charter or refine SOA Center of Excellence
- Design governance infrastructure

Enable the governance model incrementally
- Deploy governance mechanisms
- Deploy governance infrastructure
- Educate and deploy on expected behaviors and practices
- Deploy policies

Measure, monitor and manage the governance processes
- Monitor compliance with policies
- Monitor compliance with governance arrangements
- Monitor effectiveness metrics

Figure 2: The SOA governance lifecycle

As shown in figure 2, the governance process consists of four phases, discussed below. Note that the process is repeated iteratively. Each cycle provides an opportunity for improving the governance approach.

The IBM SOA Governance and Management Method approach

Figure 3 shows a high-level depiction of the phases, activities and tasks of the IBM SOA Governance and Management Method approach. A more detailed description of the method is available as a plug-in to the IBM Rational® Method Composer solution and as a stand-alone HTML offering from the IBM developersWorks® service at:

ibm.com/developerworks

This method seeks to describe in detail how IBM implements SOA governance and its supporting mechanisms.

The phases in the diagram, and the steps within each phase, are independent of any process or work-breakdown structure. The advantage is that the same technique can be used with different processes. The method can be tailored and modified to match the needs of the organization implementing it. This modification—performed in a method adoption workshop—will allow for the inclusion of only the activities and tasks that are needed for each phase. If appropriate, the entire method doesn't have to be implemented. Of course, these steps do presuppose that certain other techniques have been executed previously and that follow-on techniques will be performed subsequently.

The IBM SOA Governance and Management Method approach is the engagement model used with SOA customers. Built upon the SOA governance lifecycle described earlier, the method supports the four phases of the SOA lifecycle: Model, Assemble, Deploy and Manage. The SOA Governance and Management Method plan is an iterative approach to implementing effective governance to support service orientation.

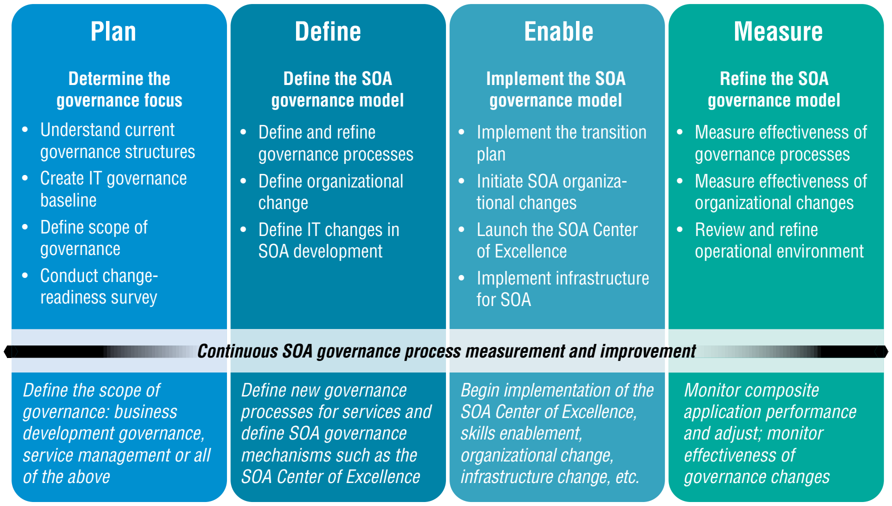
The method begins with a detailed look at the customer's existing environment, decision-making process and organizational structure. This set of activities seeks to identify components as reusable elements to build the new model. Once built, the model is implemented. This implementation may include an SOA Governance Center of Excellence as the implementation arm of the governance model, or it may entail an increase to the responsibility of the existing governance mechanisms, like an architectural review board. Measurement of the model implemented, the services deployed and the effectiveness of the established processes is essential to understanding whether services, processes and the governance model are meeting their stated objectives. This, then, constitutes the iterative governance method that will be continued by the customer once the IBM practitioners complete their engagement. Like the SOA governance lifecycle, the SOA Governance and Management Method approach has four phases:

- *Plan*
  - *Understand the current governance structures and environment.*
  - *Create an IT governance baseline.*
  - *Define the scope of the governance model.*
  - *Conduct change-readiness surveys.*

- *Define*
  - *Define and refine the governance processes, quality gates and decision-making matrix.*
  - *Define organizational change.*
  - *Define IT changes in SOA development processes.*

- *Enable*
  - *Implement the transition plan defined in the previous phase that outlines the actions needed to effect SOA governance.*
  - *Initiate SOA organizational change.*
  - *Launch the SOA Governance Center of Excellence (optional).*
  - *Implement the infrastructure for SOA.*

- *Measure*
  - *Measure the effectiveness of governance processes.*
  - *Measure the effectiveness of organizational change.*
  - *Review and refine development and operational environments.*



Plan

Determine the governance focus

- Understand current governance structures
- Create IT governance baseline
- Define scope of governance
- Conduct change-readiness survey

Define the scope of governance: business development governance, service management or all of the above

Define

Define the SOA governance model

- Define and refine governance processes
- Define organizational change
- Define IT changes in SOA development

Define new governance processes for services and define SOA governance mechanisms such as the SOA Center of Excellence

Enable

Implement the SOA governance model

- Implement the transition plan
- Initiate SOA organizational changes
- Launch the SOA Center of Excellence
- Implement infrastructure for SOA

Begin implementation of the SOA Center of Excellence, skills enablement, organizational change, infrastructure change, etc.

Measure

Refine the SOA governance model

- Measure effectiveness of governance processes
- Measure effectiveness of organizational changes
- Review and refine operational environment

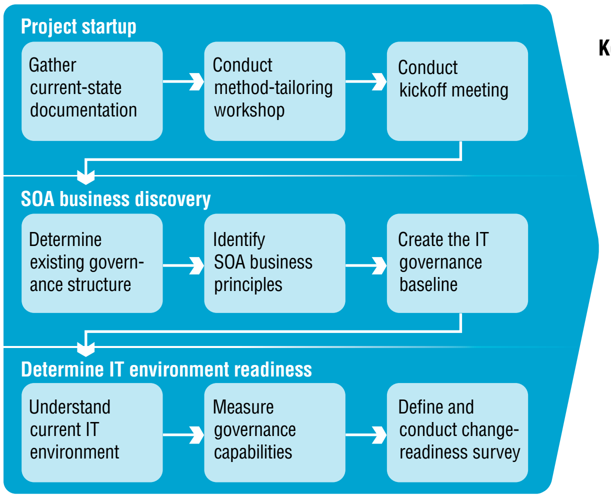
Monitor composite application performance and adjust; monitor effectiveness of governance changes

Continuous SOA governance process measurement and improvement

Figure 3: The SOA Governance and Management Method approach

Plan

Good IT and SOA governance results in better alignment of the IT organization and the needs of the business. It is in the Plan phase that the needs and priorities of the business are documented, along with the role of the IT organization in meeting those needs. Also, the state and maturity of the current IT organization governance are assessed and gaps are identified. From all this analysis, the governance vision and strategy, as well as the roadmap and plan, are documented. In the plan, the governance measures are put in place. These measures are used to assess how well the IT organization is aligned with the business and how well the business needs are met.


Project startup
Gather current-state documentation
Conduct method-tailoring workshop
Conduct kickoff meeting
SOA business discovery
Determine existing governance structure
Identify SOA business principles
Create the IT governance baseline
Determine IT environment readiness
Understand current IT environment
Measure governance capabilities
Define and conduct change-readiness survey

Key artifacts and deliverables

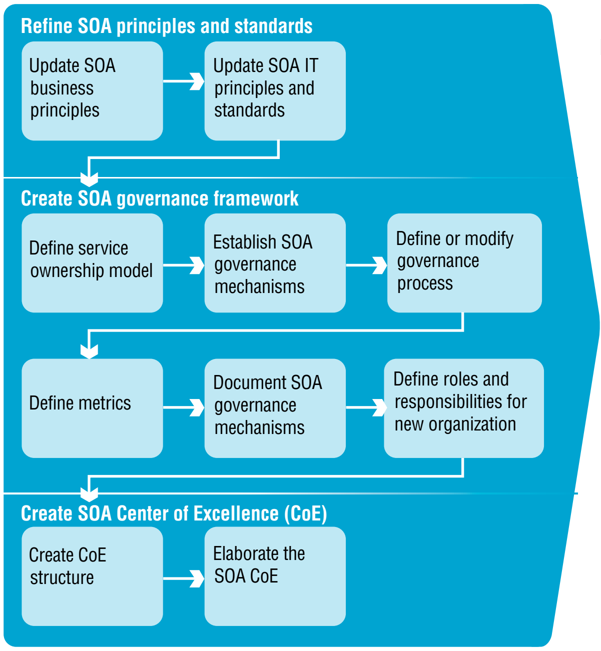
- Work-breakdown structure and project plan
- Current IT assessment
- Current organizational assessment
- Principles, policies and guidelines
- Standards
- Operational model
- Current IT environment
- Capability interview findings
- Change-readiness interview findings
- Change-readiness assessment
- Situational analysis and impact report

Figure 4: Plan phase activities and tasks

Define

In the Define phase, the detailed governance plan is put in place for the current cycle. In particular, the processes to be governed are specified and prioritized, and the decision rights, policies and measures for these processes are defined.

In preparation for the next phase, detailed deployment plans are set. In some cases, these plans may include specifying or updating the structure and staffing of the SOA Governance Center of Excellence.


Refine SOA principles and standards
Update SOA business principles
Update SOA IT principles and standards
Create SOA governance framework
Define service ownership model
Establish SOA governance mechanisms
Define or modify governance process
Define metrics
Document SOA governance mechanisms
Define roles and responsibilities for new organization
Create SOA Center of Excellence (CoE)
Create CoE structure
Elaborate the SOA CoE

Key artifacts and deliverables

- Work-breakdown structure and project plan
- Updated principles, policies and guidelines
- Updated standards
- Service ownership model
- Process definition
- RACI matrices (responsible/accountable/consult/inform)
- Change-readiness assessment
- Situational analysis and impact report

Figure 5: Define phase activities and tasks (continues on next page)

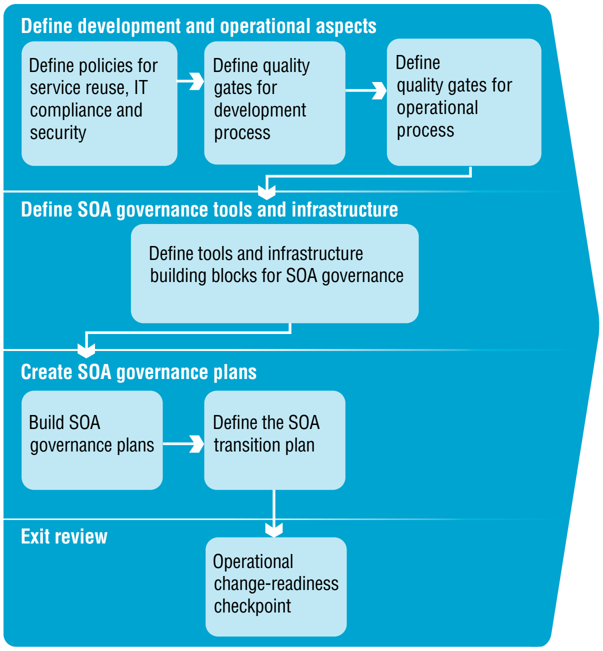

Define development and operational aspects
Define policies for service reuse, IT compliance and security
Define quality gates for development process
Define quality gates for operational process
Define SOA governance tools and infrastructure
Define tools and infrastructure building blocks for SOA governance
Create SOA governance plans
Build SOA governance plans
Define the SOA transition plan
Exit review
Operational change-readiness checkpoint

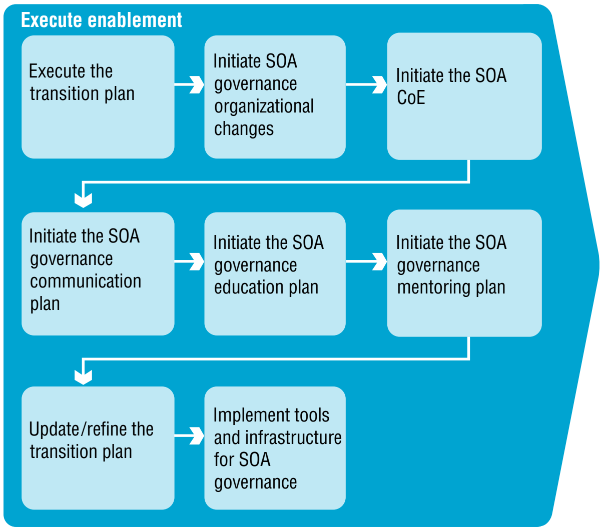
Figure 6: Define phase activities and tasks (continued)

Key artifacts and deliverables

- Measurement program
- SOA governance mechanisms
- Job roles—responsibilities and competencies
- Future organizational design
- Procedures documentation
- Quality gate check
- Key decision record
- Communications plan
- Education plan
- Mentoring plan
- Transition plan

Enable

The Enable phase is when the defined solution is rolled out to the organization. In this phase, roles are assigned, staff are trained, the decision rights may be automated in workflow tools, and the metrics collection and report mechanisms are put in place.


Execute enablement
Execute the transition plan
Initiate SOA governance organizational changes
Initiate the SOA CoE
Initiate the SOA governance communication plan
Initiate the SOA governance education plan
Initiate the SOA governance mentoring plan
Update/refine the transition plan
Implement tools and infrastructure for SOA governance

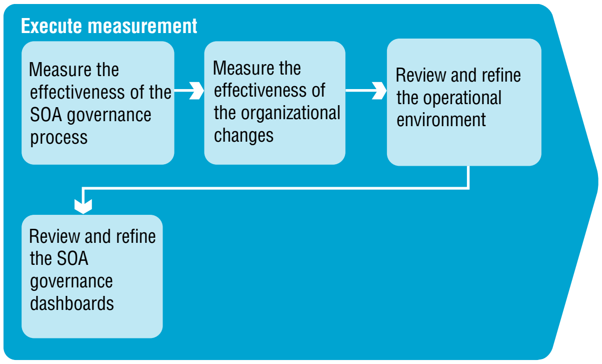
Key artifacts and deliverables

- Updated work-breakdown structure and project plan
- Updated procedures documentation
- Updated key decision record
- Updated transition plan
- Updated communications plan
- Updated education plan
- Updated mentoring plan

Figure 7: Enable phase activities and tasks

Measure

In this phase, the governance approach is executed and tuned. The governance metrics—those that show alignment with the business—are gathered. These metrics are used in the next cycle to revise the governance approach.


Execute measurement
Measure the effectiveness of the SOA governance process
Measure the effectiveness of the organizational changes
Review and refine the operational environment
Review and refine the SOA governance dashboards

Key artifacts and deliverables

- Measurement reports
- Updated procedures documentation
- Updated key decision record
- Updated transition plan

Figure 8: Measure phase activities and tasks

Tooling and SOA governance

The proper tooling can play an important role in enabling an SOA governance model. It can help a company to more effectively manage the service lifecycle within a shared environment; facilitate communication between service providers and consumers; help enforce policy, including security; measure quality of service; and provide many other beneficial contributions. However, the role of tooling must be understood when implementing an SOA governance model. Tooling should not be considered SOA governance itself; rather, it is an enabler to the underlying processes within an SOA governance model. This section will closely examine many of the common tools implemented within an SOA to facilitate the governance process.

Registries

Registries are tools that allow a company to effectively manage the metadata of its services to ensure that service consumers and providers have a common understanding of the service. One main function of the registry is to allow service consumers to discover services. Discovery can take place at design time or at run time.

At design time, most registries will provide a user interface so that interested consumers can browse a catalog of services. Discovery can also take place through a development user interface that is integrated with the registry. Once a consumer finds the service that will fit his or her needs, he or she can download the required artifact(s), such as Web Services Description Language (WSDL), to understand how to interface with the service.

Registry products seem to be taking two different approaches to delivering artifacts to interested consumers. If a registry is based on Universal Description, Discovery, and Integration (UDDI) standards, the registry will hold a reference to the physical location of the artifact as opposed to storing the artifact itself. Other registries, which are non-UDDI-based, also are being branded as repositories because they store the physical artifacts directly in the registry, as opposed to maintaining a reference to the location of the artifact. The various artifacts that a service stores in these registries include WSDL specifications, COBOL copybooks, XML schemas or any other artifact that describes how to interface with the service.

For run-time discovery, most registries provide services-location transparency. Instead of having the enterprise service bus (ESB) store the location of the service, the location is stored in the registry, and the ESB caches this location. When the location changes, the provider can simply update the location in the registry; this prompts the ESB to recache the location, and then service requests are directed to the new location. Through this location transparency and level of indirection, the provider has much more flexibility if the physical location of the service has to change.

Another important feature for a registry is to support the concept of service classification. One of the fundamental aspects of an SOA governance model is to define the appropriate service taxonomy for that organization. It is less likely that this taxonomy will be consistent across the organization and enforced appropriately unless the registry mirrors that taxonomy with its classification functionality. Basically, the classification features in a registry are a convincing enabler for an organization to realize the service taxonomy that it has defined. The classification features for most registries are very configurable, giving the organization complete autonomy in defining its own specific categories.

The registry also provides an effective means to manage the service lifecycle within a shared environment. What happens when a service version is going to be retired? How will affected consumers be informed? The registry can play an effective role in serving as the required communication conduit, keeping all interested parties informed of these types of events. Many registry products will have the notion of a state model that will allow an organization to define the various states or phases that a service will transcend throughout its lifecycle. When a service reaches certain criteria set forth by the business or IT operations, the state of the service changes in the registry and all interested parties are notified.

So how does the registry keep everyone informed of such important events as a change in metadata or state? Many registries support the concept of notification. This notification functionality is usually implemented as a publish/subscribe—or "pub/sub"—messaging pattern. Service consumers subscribe to these events through the registry and when certain service events occur, such as a state change, a modification to the service's metadata or a change in end-point location, the registry sends a message to all interested subscribers informing them of the change.

However, service consumers are not the only interested parties who want to subscribe to these notifications. The registry is an excellent place to store policies around various quality-of-service criteria. Many of the parameters that are monitored on a service to ensure certain quality-of-service levels are stored in a registry as policy. Distributed monitoring agents (covered in the next section) can also subscribe to a registry to receive notifications of when/if these quality-of-service criteria change. Once these agents receive notifications regarding a change in quality-of-service criteria, they can monitor according to the new criteria.

Monitoring and management

Every piece of tooling within an SOA has its own independent management functionality. An ESB has functionality that allows an administrator to stop and start the broker and queues. Registries provide management capability through an independent management user interface. Each underlying implementation of a service also can be managed independently through its platform's user interface, so that the service can be brought up and down, or IT staff can perform other important administrative tasks. These simple management tasks become much more difficult when it is necessary to juggle various administrative jobs on all of these disparate systems where there may be unknown dependencies. This stresses the need for a centralized management solution that will allow SOA administrators to get a peripheral view of what is happening across all critical systems throughout the enterprise and to uniformly perform administrative tasks on these various components.

These management products work with many of the tools that serve as the foundation for an ESB and multiple application server environments that run the various services in an SOA. ESBs, registries, security and the services themselves can all be effectively managed through one management solution. Many of these management products are platform-agnostic and have the ability to consolidate the management of various Java™ Platform, Enterprise Edition (Java EE) application servers (IBM WebSphere®, WebLogic, etc.) as well as Microsoft® .NET technology.

Another important feature that management tools provide is monitoring. Ensuring specific levels of service is a critical responsibility for any SOA infrastructure, which stresses the importance of the monitoring capabilities of such an infrastructure. There are two important aspects of effective monitoring: monitoring the infrastructure itself to ensure that it is serving its intended purpose, and monitoring the services that are exposed to the infrastructure.

Most management and monitoring products use distributed agents that are installed on the native systems that an organization wants to manage. You can have an agent that is specifically designed to manage and monitor a product that serves as the foundation for an ESB or for each type of application server. These distributed agents send messages to and receive messages from a central management server that serves as the nerve center aggregating all necessary management events. These events can be viewed through a single management user interface or portal where SOA system administrators can react to certain events and/or perform scheduled administrative tasks.

The ability to manage and monitor an SOA is critical to ensuring that your SOA is delivering tangible benefits that are consistent with the architecture's original intent. A critical piece in gauging the effectiveness behind an SOA governance model is to define measurable benchmarks, so that the enterprise can easily understand what needs to be improved in subsequent iterations. Managing and monitoring solutions not only provide the means to measure these benchmarks, but they also provide a lever to control undesirable events that could negatively impact the performance of the SOA infrastructure.

Security

There are numerous approaches to security within an SOA. Do you implement it as a gateway before the request hits the ESB? Do you implement it as a service that gets called within a mediation of the ESB? Do you do both? Do you implement security at the transport layer? Do you do it at the message layer? Do you implement both a transport and a message layer? The topic of security within an SOA is far too exhaustive to cover within a single section of a white paper. Instead, this section will highlight tooling used within security and how tooling fits within the realm of SOA governance.

Two of the main questions an effective governance model needs to answer are, Who has access to do what, and How is that access enforced? These questions should be answered through the SOA infrastructure's security implementation. Many times, security is implemented in a piece-by-piece manner, with each application/service implementing an independent access- and identity-management solution. However, what happens if an organization finds out that a specific user is abusing his or her user rights and the organization wants to delete that user from all identity-management and access-management systems immediately? Does the organization know every identity- and access-management system where that user has an account? Would the organization have to go into each identity- and access-management system and delete the user from that specific system? This process wastes time and creates the possibility that the offending user doesn't get deleted from all systems. However, this problem is solved if the organization uses an integrated identity-management and access-management solution. This tool provides an enterprise with a means to integrate all autonomous identity- and access-management solutions into a single centralized identity- and access-control solution. Thus, all accounts throughout the enterprise can be managed in one central location.

By using an integrated identity- and access-management system, security is now managed at the enterprise level, allowing administrators to enforce their security policy more effectively and efficiently.

Testing

Testing is a crucial piece of any software development process, let alone an SOA development process. A good SOA governance model will define the recommended tools and processes to test various services, ESB mediations, etc. There are two types of testing to consider: functional testing and load testing. Functional testing ensures that the service has implemented logic that behaves consistently with the expectations of the business. Load testing basically ensures that all nonfunctional requirements, such as transactions per second or response times, are sufficiently met.

Depending on the robustness of the development environment, these testing tools will be incorporated into the development tool or come as additional plug-ins. It is important for a governance model to not only standardize the tooling itself, but also standardize the processes on how to use the tooling. Thus, within the definition of an organization's standards and best practices, a governance model should define best practices around the programming of test suites and what nonfunctional requirements should be defined and confirmed for each service or mediation during its testing phase.

Conclusion

This section covered some of the tooling that could be used to help implement an SOA governance model. It illustrated how tooling could provide an effective means to manage the service lifecycle within a shared environment, to monitor service levels to ensure that the appropriate quality of service is being maintained, to implement identity management in a centralized manner, and to ensure that the testing of your services and SOA infrastructure is thorough and that test results can be interpreted uniformly throughout the organization.

It is important to realize that SOA governance is not a tooling problem. However, tooling is an enabler that helps enforce SOA governance and helps ensure consistency throughout the organization's SOA environment. Proper tools provide functionality that complements defined SOA governance processes, and if implemented and utilized correctly, the proper tools can be a tremendous asset that boosts the effectiveness of an SOA governance model.

Case studies—implementing the IBM SOA Governance and Management Method approach

We will present three case studies of actual engagements where the IBM SOA Governance and Management Method approach has been implemented. Each case study will describe the implementation for a specific type of engagement, starting with the governance model for an ESB.

Major automobile manufacturer

Project background

This SOA governance project was implemented by a large Japanese company that specializes in manufacturing. The company had originally defined two services that it wanted to use enterprise wide. The first service was to help centralize access to crucial customer information; the second service was to handle real-time messages during the manufacturing process.

This company realized that it wanted to implement a significant number of additional services and wanted to establish an SOA infrastructure that would allow the organization to accelerate the service-integration process between consumers and providers. This infrastructure consisted of three main elements: an ESB, a registry and monitoring tools. The ESB's role was to help promote connectivity and interoperability across various providers and consumers. The registry was implemented to facilitate communication between consumers and providers, and to provide a run-time layer of indirection. The monitoring tools provided the capability to monitor key events within the infrastructure and provided usage metrics for services. The tools also helped to ensure that these services were delivering what they promised.

To complement this infrastructure, the company understood that an SOA Governance Center of Excellence and an underlying governance model would be crucial in earning the required organizational acceptance for this initiative to be a success. The company realized that it was considering most of the necessary tooling elements to complement an SOA governance model, but it also had the foresight to understand that SOA governance is not something that can just be installed on a hard disk. The company believed that this governance model needed to define key foundational processes; the company also understood that the SOA Governance Management Method approach provided a clear path to realizing that goal. The following sections will describe how this governance model was implemented using the SOA Governance Management Method approach, focusing on the key phases of Plan, Define, Enable and Measure.

Plan phase

Significant effort was placed on the Plan phase of the project. This allowed the company to look within itself and understand its immediate and long-term goals, its existing organizational structure and IT governance processes, and its culture. The company examined the enterprise-wide initiatives that were successful and why these initiatives succeeded. Conversely, it also looked at what enterprise-wide initiatives failed and why these initiatives failed. Having a solid understanding of these inputs was the critical first step in defining a governance model that would result in the required organizational buy-in indicative of a successful SOA governance model.

To better understand the various factors listed above, interviews were conducted with all major stakeholders within the organization. These stakeholders included business managers and architects from the various application teams, chief architects within the organization who carried significant influence, and members of the senior executive steering committee. If someone had either a direct or indirect stake in the infrastructure, he or she was a prime candidate to be interviewed.

The key elements that were discovered during this assessment were:

- *This company was guided by a very strong consensus-driven corporate culture where new ideas required significant "socializing" throughout the organization.*
- *Reasons that past enterprise-wide initiatives succeeded were because the appropriate resources were provided to socialize the initiatives and execute the appropriate organizational change management. Failures were explained as a lack of socialization.*
- *All interviews identified the execution of organizational change management as the number one risk, not technology.*
- *The customer wanted to establish a Center of Excellence that was ESB-centric and could grow into an SOA Governance Center of Excellence as the organization gained momentum.*
- *The customer's initial goal was to accelerate the delivery of integration efforts to promote organizational buy-in of the SOA infrastructure.*
- *The current IT governance model used a multitiered approach.*

Once the interviews were finished, the governance team had the necessary inputs to understand the company's short- and long-term goals. To help solidify a plan of attack, the team outlined an SOA vision that would help it reach its goals using a phased approach based on the three main facets of SOA governance: financial transparency, business IT alignment and process control. Initially, the plan would focus on the process control aspects of governance. Once this governance model gained momentum over time, focus would shift to the business/IT alignment aspects of governance. The third phase of the governance adoption would be to incorporate greater detail on the financial transparency aspects.

Now that there was a plan in place, the team had the required outputs to move to the Define phase.

Define phase

After this customer spent the necessary amount of time assessing its current goals, culture and organizational dynamics, it was ready to start defining its Center of Excellence. The main elements defined within this phase were:

- *The organizational structure and appropriate roles*
- *The foundational processes used to govern*
- *The standards and best practices outlining the rules of engagement for the ESB*
- *The organizational change-management plan*

Extending the IT governance organization that was already in place, the organizational structure that was defined for this Center of Excellence was a direct reflection of this company's consensus-driven culture. Instead of defining individuals within a hierarchy, this company defined an organization that consisted of three layers of teams or committees. The company felt that teams would be more effective in operating under a culture that was a product of consensus and significant socializing of ideas. The first committee was the senior advisory committee, which consisted of the senior executives who were funding the organization and who met periodically to ensure that the organization was performing as promised. The second team was the leadership team, which ensured that the organization remained vital and evolved with the changing needs of the business. The third team was the core team that provided the application teams with a lot of the technical thought leadership around integration, such as ESB mediation development or integration architecture best practices.

Both the leadership and core teams were staffed with resources that were full time and rotational. The rotational resources came from the application teams, and the theory was that those rotational members would serve as liaisons between the Center of Excellence and the application teams. They would help to communicate the standards and best practices to the application teams, promote the value of the infrastructure, and voice the interest of the application teams back to the Center of Excellence. By providing the application teams with representation, the company knew that employees would be more prone to accept the organization and, ultimately, the SOA infrastructure.

The Define phase also required the definition of some key governance processes. The processes defined were:

- *A service-development process that provided various checkpoints throughout the shared-service lifecycle to ensure that various criteria were met before the service could graduate from one phase to another*
- *A process that defined the various checkpoints to ensure that the service was compliant with the standards and best practices defined by the organization*
- *A process that ensured the Center of Excellence would evolve and always remain vital to the organization*
- *A process that allowed application teams to challenge and appeal the Center of Excellence's methods*
- *A process to effectively communicate various Center of Excellence changes*

In order to breed the environment of consistency necessary for the SOA to be successful, the organization also developed an architectural guidebook that outlined the standards and best practices that all services on the ESB had to follow. This guide was seen as the rules of engagement for the ESB. Since this company already had an IBM Rational Unified Process®—or IBM RUP®—methodology in place, the guidebook was organized in accordance with the various RUP phases of development. Thus, an architect could look at the Elaboration phase section of the guidebook and understand the architectural decisions that the team would face during that phase, and know how to respond.

Developed through a series of architectural workshops that included application architects and chief architects within the organization, the document tackled tough issues around defining appropriate service granularity, choosing the appropriate mode of integration (i.e., extract, transform, and load [ETL] versus ESB), determining the appropriate use of the registry, establishing the required contents within SLAs, interfacing with security, determining naming conventions, developing and deploying mediations to the ESB, versioning, and service management. The architectural workshop approach served two main purposes: to define the standards and best practices, and to provide application teams with representation on the standards and best practices—using the notion that people are less likely to reject something that they helped define.

The standards document was not just a laundry list of generic standards and best practices to follow, but it was also a program focused on this company's specific implementation. It incorporated best practices on how the registry tooling should be used to effectively manage the lifecycle of a shared service. It also described how the monitoring tooling should be implemented to measure quality of service to enforce SLAs. One of the guide's themes was to define how the key governance processes could be complemented through the use of the SOA infrastructure's tooling.

The last element addressed in the Define phase was the effort around organizational change management (OCM). Given the company's explicit concern in this area, great emphasis was placed on this effort. The company understood that the only way the appropriate organizational acceptance of the infrastructure could occur was through effective OCM, and that level of acceptance would define the success of the organization's SOA initiative.

Through these efforts a communication strategy, communication plan, change-readiness plan and change management plan were defined that would ensure that the infrastructure's benefits were properly communicated throughout the organization and that any required organizational changes would be executed in a manner that would minimize disruption and still be effective. These plans would focus on what key targets to hit first, so that critical momentum would be established throughout the organization.

Once the governance plan was defined, it was ready to be implemented.

Enable phase

This customer is currently in the Enable phase of its Center of Excellence and underlying governance model. This phase of the work focuses on implementation and execution. Within this stage, the organizational structure will be initiated, the communication plan executed and the necessary skills ramped up.

To foster an environment of consistency, application teams will be required to use defined templates around SLAs, service descriptions within the registry, and other important templates that were established in the Define phase. The checkpoints defined in the service-development process to ensure that standards and best practices are being followed appropriately will be initiated using Center of Excellence resources as the enforcers.

It is in this stage that the tooling will be implemented to complement the foundational processes. The registry will be implemented to enforce appropriate lifecycle management of a service within a shared environment. The monitoring tooling will be implemented to enforce the SLA process.

Measure phase

In the Measure phase, the customer is looking to answer the question of how well it performed on defining and executing its Center of Excellence. In this stage, the customer will use the outputs of its monitoring to ensure adherence to service integration policies. The customer will also look at how well the organization satisfied the business needs. It will compile feedback from the application teams, the executive steering committee and other stakeholders so that it can evolve and remain vital to the needs of the organization.

This feedback is a result of various metrics established in the Define phase to understand the effectiveness of the organization.

Case study conclusion

This case is crucial in understanding the context around establishing an SOA governance Center of Excellence and the underlying governance model. The company performed a considerable amount of due diligence in the Plan phase to ensure that it was defining a Center of Excellence and governance model that were going to be consistent with its existing organizational dynamics, culture and goals. Putting this emphasis on the Plan phase ensured that this model would be accepted by the organization, which was pivotal to the initiative's success.

This emphasis on the Plan phase shows that governance is not a one-size-fits-all solution, which brings us to our next point. When implementing SOA governance, it is important to understand that governance is not something that can be purchased within a shrink-wrapped box. Technology is an enabler to governance, but not governance in and of itself. The approach that this customer took to implementing governance is to focus on the foundational governance processes first, and then focus on how those processes can be complemented by the tooling. SOA governance is more of an organizational behavior problem and less of a technology problem, which is why it is crucial to define processes that will be accepted by the organization.

A major telecommunications company

Project background

This IBM client identified operational excellence as one of the three pillars (others being growth and people) to drive value for shareholders, customers, suppliers and employees. The ultimate goal was to transform the quoting processes (business and IT) to realize improvements in four areas: productivity, quality, efficiency and cycle time.

IBM was chartered to design and implement a business transformation, IT governance and custom application development initiative applying an SOA infrastructure, governance and service-oriented modeling and architecture (SOMA) methodology.

Project objectives

Early in the process, the client established its project goals:

- *Identify business transformation and organizational improvements in*
- *the quoting business processes that have immediate payback and can be rapidly implemented.*
- *Assess current IT and SOA capabilities.*
- *Review current IT governing processes and expand to support the service-oriented model.*
- *Identify services and organizational impacts.*
- *Identify key performance indicators (KPIs) and align services to support SLAs.*
- *Define and implement a governance plan.*

Solution

The company requested that IBM design an SOA infrastructure supported by SOA governance processes that would be backed by key project stakeholders. Applying the IBM SOA Governance Management Method approach, IBM defined and implemented an end-to-end solution.

Plan phase

In the first stage of this SOA governance solution, the client and IBM teamed to determine the following conditions and perform early-stage tasks:

- *Assessments were conducted of the existing environment. One of the key issues discovered was an existing consensus-based decision model that caused a delay in—and sometimes a lack of—governance in key project areas. This was in part due to a breakdown in communications across departments, but also because the accountability lines were forming but had not been solidified among the key project stakeholders. It was common that a decision was made one day only to be reversed the next.*
- *A lack of confidence also existed between the business and IT teams partially due to the inability of some to make decisions in a timely and efficient manner, but also due to unmet expectations. The business users expressed that many times they would request service A only to receive service B and C, and the IT teams expressed related concerns.*
- *SOA-focused assessments revealed, among other things, that cross-teaming and communications were inevitable for reusable services to be designed and implemented effectively and bring about real business value.*
- *Services existed but the underlying infrastructure would require upgrades to support expected volumes and meet SLAs. New services would also require definition and creation.*
- *Software and hardware selection criteria were established to assist with the selection process in support of the new solution.*

Define phase

In the second phase, the IBM and client teams defined goals and took steps to achieve those goals.

- *Working with the client, IBM defined a one-team vision that would promote and encourage collaboration across domains and ensure that decisions were geared toward achieving one goal—a new and improved quoting solution.*
- *Change agents were brought on board to help design organizational changes and processes for implementing.*
- *An architecture team, review board, and supporting roles and responsibilities were put in place to accelerate the decision-making process.*
- *The overarching IT/SOA governance model was established to ensure that all decisions positively integrated with and impacted the enterprise, as well as aligned with business goals (see figure 9 below).*

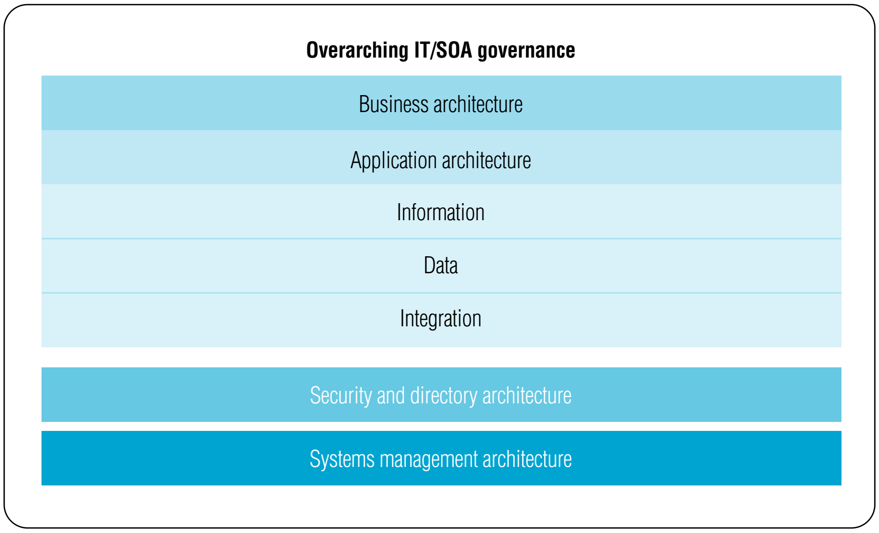

Overarching IT/SOA governance
Business architecture
Application architecture
Information
Data
Integration
Security and directory architecture
Systems management architecture

Figure 9: Overarching SOA/IT governance domains

- *Design and development standards were created and distributed across teams to ensure communications and consistency.*
- *KPIs and SLAs were documented and built into the service model.*
- *The existing SOA Center of Competency was reviewed and recommendations were made to support the new processes and business requirements.*
- *Vendors were required to proactively participate in design discussions, decisions and sign-off solutions before and after realization.*

Enable phase

This client is currently in the Enable phase of the project.

- *An SOA framework was defined and realized, and the review board continues to ensure that solutions adhere to standards.*
- *The ESB and business-rules engine were implemented as vehicles to control tight coupling and to eliminate embedded business logic with hopes that this will support a flexible architecture and also contribute to improved IT/business relations.*
- *Proofs of concept (PoCs) were initiated in earlier phases of the project to ensure interoperability and integration of diverse infrastructure components. In the Enable phase, additional PoCs were conducted to ensure functional viability.*
- *Roles and responsibilities were validated and security infrastructure was implemented leveraging a portal-based solution along with IBM Tivoli® Access Manager software.*
- *Weekly architecture review meetings were held to not only allow for decision making but also bring about collaboration between business and IT team members.*
- *A governance model was also put in place to ensure consistency among U.S., vendor and global development teams, including frequent design and development peer reviews.*

During this engagement, IBM helped the client establish an architecture decision flow model. An example model is highlighted below.

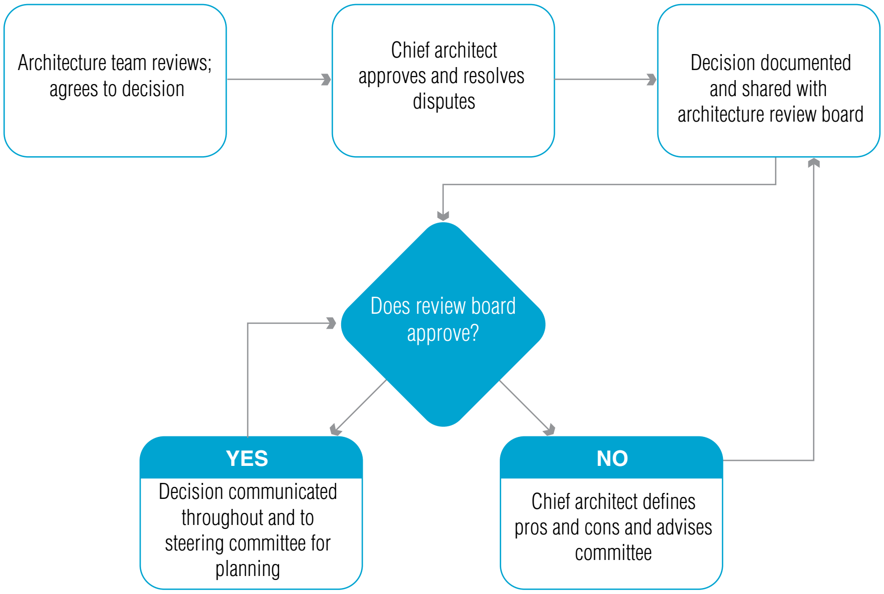

Architecture team reviews; agrees to decision
Chief architect approves and resolves disputes
Decision documented and shared with architecture review board
Does review board approve?
YES
Decision communicated throughout and to steering committee for planning
NO
Chief architect defines pros and cons and advises committee

Figure 10: Architecture decision flow

Manage and Measure phases

Because this was an iterative development solution, some governance metrics were captured, and the team was afforded the opportunity to address and recover, but this phase is also still in process.

Case study conclusion

This case study is crucial to understanding the context around establishing an SOA/IT process and governance model. The critical success factor was the alignment of business and IT in establishing a governance model. A significant amount of work was conducted in the Plan and Define phases of the project to build a collaborative business and IT model.

Other critical success factors were the focus on the architecture purview in figure 10 and the underlying enablers, as well as the focus on a common vision. This helped to enhance the end-customer quoting experience.

It also helped that this client had defined an SOA Center of Competency, but a technique was required. Additionally, it was beneficial that the organization had spent time convincing the team to buy in to new ideas and implementation techniques.

It is very important to establish a technique to capture measurable SLAs and identify tools that will measure them. More time could have been utilized in this space.

Finally, the introduction of change agents to lead the identification and realization of organizational changes and impacts was crucial to the project's success.

Conclusion

The intent of SOA is to achieve a better alignment between the business world and the IT world—in a way that makes both more agile. Service orientation alone cannot bring about the effect of improved productivity, faster time to market and reuse, because there are other forces in place that operate as impediments and constraints.

This paper has shown that effective SOA governance of services through policies, principles, standards, procedures, processes, and cultural and organizational change enables the full benefit of service orientation to be realized.

The IBM SOA Governance Management Method approach establishes chains of responsibilities, decision-making rights and authority, and communications to empower people to bring about the necessary changes to drive SOA success and to sustain those changes.

Appendix

The authors of this paper would like to acknowledge the following contributors, who are listed below in alphabetical order:

- *Marty Burns, Executive IT Architect, IBM Architecture and Technology CoE*
  mjburns@us.ibm.com
- *Claudio Cozzi, Distinguished Engineer and Chief IT Architect – Legacy Transformation to SOA, IBM Global Services SOA CoE*
  ccozzi@us.ibm.com
- *Richard Disney, IT Architect, IBM SOA CoE*
  disneyr@us.ibm.com
- *Pamela K. Isom, Certified Senior IT Architect IBM GBS EA&T SOA CoE*
  pisom@us.ibm.com
- *Tilak Mitra, Certified Sr. IT Architect, IBM GBS EAD Group*
- *Garry Moore, Consulting IT Architect, IBM Global Services*
  gmoore@ca.ibm.com
- *Krishnan K. Ramachandran, Certified Executive IT Architect, Enterprise Architecture & Technology CoE, SOA & Web services CoE*
  ramachan@us.ibm.com
- *Jon Richter, Solution Architect, IBM Software Services for WebSphere*
  jonrichter@us.ibm.com
- *William Tegan, Associate Partner, Application Innovation, IBM Global Services*
  wjtegan@us.ibm.com
- *Raghu Varadan, Chief Architect, SOA and Web Services CoE, IBM AIS/Global Business Consulting Services*
  rvaradan@us.ibm.com

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
08-06
All Rights Reserved

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided “as is” without warranty of any kind, express or implied. In addition, this information is based on IBM’s current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.