

# Learn About Service Oriented Architecture (SOA)

What is Service Oriented Architecture?

SOA defined

Why SOA?

Who does SOA?

What SOA isn't

Before starting a SOA

What is the SOA life cycle?

What are the benefits of SOA?

What are the challenges associated with SOA?

How can your organization get started with SOA?

## **What is Service Oriented Architecture?**

IT departments are managing increasingly complex IT portfolios. Yet as business needs change, these departments must still ensure that their technologies remain aligned with business goals. Failure to do so compromises organizational agility.

The problem for IT departments is typically not insufficient functionality; rather, it is that critical business systems such as customer relationship management (CRM) and enterprise resource planning (ERP) operate in isolation from other critical business systems—despite the fact that business processes often span multiple applications. To obtain an end-to-end view of a complex business process necessitates integration of information and process silos. In the past, this has been accomplished either through time-consuming manual interventions, or through hard-coded solutions that are difficult to maintain.

Service orientation is an approach to organizing distributed IT resources into an integrated solution that breaks down information silos and maximizes business agility. Service orientation modularizes IT resources, creating loosely coupled business processes that integrate information across business systems. Critical to a well-designed service-oriented architecture is producing business process solutions that are relatively free from the constraints of the underlying IT infrastructure, because this enables the greater agility that businesses are seeking.

Service Oriented Architecture (SOA) ultimately enables the delivery of a new generation of dynamic applications (sometimes called composite applications). These applications provide end users with more accurate and comprehensive information and insight into processes, as well as the flexibility to access it in the most suitable form and presentation factor, whether through the Web or through a rich client or mobile device. Dynamic applications are what enable businesses to improve and automate manual tasks, to realize a consistent view of customers and partner relations, and to orchestrate business processes that comply with internal mandates and external regulations. The net result is that these businesses are able to gain the agility necessary for superior marketplace performance.

## **SOA defined**

Service orientation is a means for integrating across diverse systems. Each IT resource, whether an application, system, or trading partner, can be accessed as a service. These capabilities are available through interfaces; complexity arises when service providers differ in their operating system or communication protocols, resulting in inoperability.

Service orientation uses standard protocols and conventional interfaces—usually Web services—to facilitate access to business logic and information among diverse services. Specifically, SOA allows the underlying service capabilities and interfaces to be composed into processes. Each process is itself a service, one that now offers up a new, aggregated capability. Because each new process is exposed through a standardized interface, the underlying implementation of the individual service providers is free to change without impacting how the service is consumed.

## **Why SOA?**

Complex, distributed IT resources are a concern for businesses. Too frequently, the existing IT portfolio does not adequately meet specific business needs, is costly to manage and maintain, and is inflexible in the face of business growth and change. The solution, however, is not to rip and replace systems or applications, nor to completely renovate them, but rather to find a way to leverage existing IT investments so that overall organizational goals are effectively supported. Service orientation helps to accomplish these goals by making systems more responsive to business needs, simpler to develop, and easier to maintain and manage. Implementing a solution architecture based upon service orientation helps organizations plan ahead for change, rather than responding reactively.

## **Who does SOA?**

Strictly speaking, SOA is done by developers and solution architects. However, stakeholders in a service-oriented solution span a range of roles, and it is critical that their interests not only be taken into account but that they actively drive the design of the SOA solution.

Starting with those interests, the business analyst is concerned with bringing IT investments more in line with the business strategy. For the developer, this means that the SOA solution must map the sources of business information—systems, staff, trading partners—into a unified and comprehensive view such that the business analyst has greater insight into the costs and benefits of various investments.

The chief technology officer (CTO) of the organization will work with developers to ensure that when designing a solution to meet the needs of the business analyst, the integrity of existing IT systems and applications resources are preserved, even as new capabilities are developed.

And the IT manager, concerned with effectively integrating distributed systems such that management is simplified, will work with the developer to ensure that these goals are also met. Ultimately, the developers and solution architects are concerned with creating dynamic collaborative applications that meet the goals of the various stakeholders. The service orientation approach enables them to do so in a way that meets the needs of the organization as a whole.

### **What SOA isn't**

There are numerous misconceptions about what SOA is—that it is a product that can be purchased (it is not; it is a design philosophy that informs how the solution should be built); that the goal is to build a SOA (it is not; SOA is a means to an end); or that SOA requires a complete technological and business process overhaul (it doesn't; SOA solutions should be incremental and built on current investments).

SOA is also often equated with Web services, and the terms used interchangeably. While it is true that SOA is made easier and more pervasive through the broad adoption of Web services–based standards and protocols, the two are distinct. SOA is an approach to designing systems—in effect the architectural drawings or blueprint—that directs how IT resources will be integrated and which services will be exposed for use. In contrast, Web services is an implementation methodology that uses specific standards and language protocols to execute on a SOA solution.

### **Before starting a SOA**

Before a developer writes a single line of code, it is critical to identify both specific business drivers of the SOA endeavor and the dependencies between the business and the underlying technologies. Neglecting the business context can result in a project in which SOA infrastructure is pursued for its own sake, or where investments are made that do not line up well with the needs and priorities of the business.

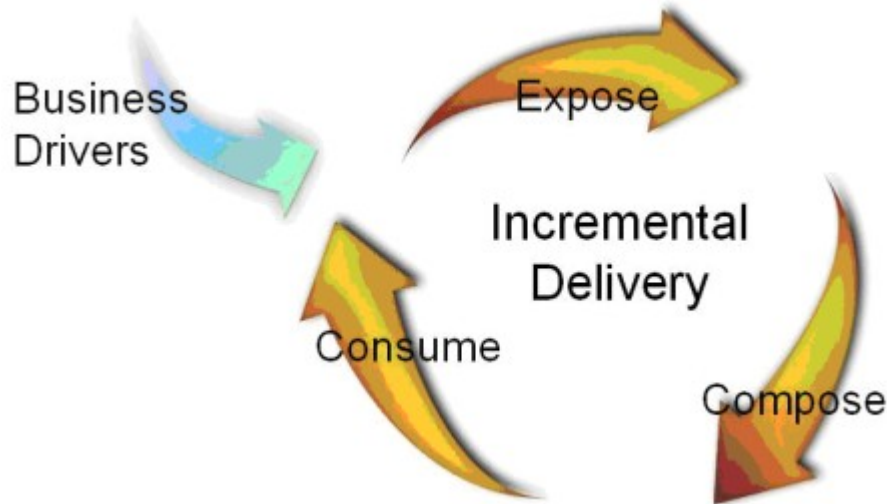
Two approaches are commonly pursued for implementing SOA: top-down and bottom-up. Both approaches have possible pitfalls that can prevent success. Many organizations that have attempted to roll out SOA infrastructure through a top-down approach have discovered that when the infrastructure is finally delivered it is out of sync with the needs of the business. Likewise, a bottom-up approach can fail as well, because it can lead to a chaotic implementation of services created without regard to organizational goals.

The “middle-out” approach is a successful hybrid of the two other approaches. Business drivers and strategic vision are first employed to set clear direction and priorities. Based on these, the organization takes multiple iterative steps to build out slices of end-to-end capabilities, with each iteration delivering a new, dynamic application back to the business that is used to create business return. Microsoft has long advocated this “real-world” approach to leveraging service-oriented architectures: The approach is focused on rapid time-to-value, and it delivers business results through iterative, incremental steps that facilitate close alignment of IT resources with changing business conditions.

### **What is the SOA life cycle?**

The core IT assets of any organization include its data, legacy systems, line-of-business applications, packaged applications, and trading partners. Each of these resources is a service provider responsible for producing numerous highly specific outputs, such as inventories and customer data.

Service orientation ties together these disparate and autonomous sources of information, bridging a wide range of operating systems, technologies, and communication protocols. The process by which it does this is an iterative one of creating (“exposing”) new services, aggregating (“composing”) these services into larger composite applications, and making the outputs available for consumption by the business user.



#### **Expose**

The expose phase of the SOA approach focuses on which services to create from the underlying applications and data. Service creation can be fine-grained (a single service that maps to a single business process) or coarse-grained (multiple services come together to perform a related set of business functions).

The expose phase is also concerned with how the services are implemented. The functionality of underlying IT resources can be made available natively if they already speak Web services, or can be made available as Web services through the use of an adapter.

#### **Compose**

Once services are created, they can be combined into more complex services, applications, or cross-functional business processes. Because services exist independently of one another as well as of the underlying IT infrastructure, they can be combined and reused with maximum flexibility. And as business processes evolve, business rules and practices can be adjusted without constraint from the limitations of the underlying applications.

#### **Consume**

Once a new application or business process has been created, that functionality must be made available for access (consumption) by either other IT systems or by end users. The goal of the consumption process is to deliver new, dynamic applications that enable increased productivity and enhanced insight into business performance. Users can consume the composed service through a number of avenues, including Web portals, rich clients, Office business applications, and mobile devices.

### **What are the benefits of SOA?**

Service-oriented architecture is, first and foremost, a means of attaining greater business agility from existing IT investments. SOA-based solutions connect systems and thereby automate previously manual information-transfer processes whether the goal is to develop new applications;

to connect systems, workgroups, or geographically distributed subsidiaries; or to collaborate with trading partners. At the same time, SOA solutions build in the essential services required to ensure that the appropriate resources are accessed by the appropriate users.

SOA benefits accrue for the organization at two different levels, that of the IT organization and that of the business user; in the end, all benefits add up to a dramatic increase in agility and productivity.

From the IT department's point of view, SOA-based integration simplifies management of distributed resources across multiple platforms, requires less hardware, is more reliable, is standards-based, and is less costly.

From the business point of view, SOA enables development of a new generation of dynamic applications addressing a number of top-level business concerns that are central to growth and competitiveness. SOA solutions promote:

- **Stronger connections with customers and suppliers.** By making dynamic applications and business services available to external customers and suppliers, not only is richer collaboration possible, but also customer/partner satisfaction is increased. SOA unlocks critical supply and demand chain processes—such as outsourcing of specific business tasks—from the constraints of underlying IT architectures, thereby enabling better alignment of processes with organizational strategy.
- **Enhanced business decision making.** By aggregating access to business services and information into a set of dynamic, composite business applications, decision makers gain more accurate and more comprehensive information. They also gain the flexibility to access that information in the form and presentation factor (Web, rich client, mobile device) that meets their needs.
- **Greater employee productivity.** By providing streamlined access to systems and information and enabling business process improvement, businesses can drive greater employee productivity. Employees can focus their energies on addressing the important, value-added processes and on collaborative, semi-structured activities, rather than having to conform to the limitations and restrictions of the underlying IT systems.

## **What are the challenges associated with SOA?**

SOA confers obvious business benefits associated with integration and the creation of new services. However, insufficient attention to governance—the management and monitoring of services, their performance and reliability, and especially their security—can cause inefficiencies and disrupt business processes and the end users they support.

As business needs evolve, it is critical to have policies in place that help determine how to prioritize new business processes and services under consideration for implementation, who will be responsible for designing those processes, how they should be implemented, and how the success of the new implementations will be measured. This is especially important given the inherent cross-functional nature of SOA solutions.

Reuse of services, once touted as a primary SOA advantage, is really a byproduct of the approach rather than the goal itself. Reuse is also proving to be more challenging than expected. An existing service may not provide exactly what a different business process requires and so may call for additional work. And designing a service so that it can be reused in the future requires accurately predicting what future needs will be, something notoriously difficult to do.

## **How can your organization get started with SOA?**

1. Make sure that you have sound business drivers. When an organization struggles to justify their SOA projects, it is almost always because they are trying to “do SOA” rather than address a business need.
2. Top-down approaches do not work in the real world. Bottom-up approaches are not manageable either. In contrast, organizations that are successful with SOA often adopt a middle-out approach. These organizations all have something in common—they start with clear business challenges and focus on creating business value.
3. Try to avoid subscribing to the “build it and they will come” approach. Some organizations spend 18 to 30 months building a services infrastructure. When they finally reach the service consumption or user-experience layer, they find that the business needs have changed, rendering the investments a waste of time and money. It is often more practical to partition your usage scenarios into small sets and build out the entire scenario top to bottom, from the data through to the application consuming the services. Partitioning functionality in this manner can help you track changing business needs much more effectively.
4. Demonstrate value in rapid iterations. Time-to-value is a critical, healthy metric. The “trust-me” approach is not a healthy model for successfully leveraging SOA.
5. Last, but not least, organizations that have successfully adopted a SOA solution often use a “snowball” approach. How do you build a big snowball? You start with a small snowball. This is probably the most important take-away with respect to leveraging SOA to drive business value.

Adapted from Microsoft.com by Ali Khademi ([www.khademi.com](http://www.khademi.com))