Introduction to the Value and Governance Model of Service-Oriented Architecture

© copyright IBM Corporation 2005. All rights reserved.
Table of Contents
Introduction to service-oriented architecture
  About this module 2
  Introduction to SOA and Web services (1 of 6) 3
  Introduction to SOA and Web services (2 of 6) 5
  Introduction to SOA and Web services (3 of 6) 6
  Introduction to SOA and Web services (4 of 6) 10
  Introduction to SOA and Web services (5 of 6) 11
  Introduction to SOA and Web services (6 of 6) 13
  SOA summary 14
  Web services defined (1 of 2) 15
  Web services defined (2 of 2) 17
  Tailoring the SOA discussion for the business person 19
  Defining SOA for the business person 20
  Simulating SOA for the business person 21
  SOA value roadmap 23
  Enlightening a business person regarding SOA 25
  Explaining architecture for business people (page 1 of 2) 28
  Explaining architecture for business people (page 2 of 2) 30
  Putting it all together: Service-oriented architecture (page 1 of 2) 31
  Putting it all together: Service-oriented architecture (page 2 of 2) 32
  Summary 34
  References 35
  Test your knowledge 36
Introduction to service-oriented architecture
About this module
This module introduces you to service-oriented architecture.

Objectives
Once you have completed this module, you should be able to:

- Briefly define service-oriented architecture
- Explain the relationship of Web services to service-oriented architecture

Before starting this module
Before reading this module, you should have a basic understanding of the concepts of SOA and Web services.

After completing this module
When you have completed this module, continue to learn more about value sell, by continuing to Business value of service-oriented architecture.
Introduction to SOA and Web services (1 of 6)

The exact origin of the SOA style of design is unknown. Gartner Group used the term service-oriented architectures in 1997 when describing an architectural principle in order to enable CORBA and DCOM programming model coexistence in the enterprise. The Object Management Group designed CORBA as a set of services. Still, no undisputable inventor of SOA has emerged. The absence of a clear creator of the SOA design style may be attributed to the natural maturing of architectural styles. The computing industry has seen an evolution of design styles for over forty years.

Regardless of the exact origin, SOA as a design style has existed for some time and has benefited from the lessons learned of its predecessor design styles. Design styles are built upon a set of principles that govern its use. These principles are the characteristics that provide the intrinsic behavior for the style of design. Systems that adhere to a style of design are expected to exhibit the principles of that design style, realizing the benefits offered by these principles. The principles of the SOA design style, also known as design characteristics, are:

- Modularity
- Encapsulation
- Loose coupling
- Separation of concerns
- Composability
- Single implementation

By designing solutions that follow the SOA style of design, these solutions can expect to benefit from these principles. Even though designing solutions for the sake of using a modern style of design might seem interesting, real business value must be realized to justify its use. The SOA style of design offers companies several benefits:

- Increased business responsiveness and agility
- Ability to transcend organizational boundaries
- Reduces product development cycle times
- Exposes commodities in business processes

Along with the benefits to the business community, the information technology (IT) community can benefit from the SOA style of design, as well. SOA can provide the following benefits to the IT community:

- Build services once and use often
- Services are built by contract
- Promotes process consistency
- Allows for localization of function and standardization of cross-cutting concerns
- Standardizes integration and reduces solution complexity

With the understanding of the principles of the SOA style of design and the benefits to the business and IT communities, an IT architect is able to determine the applicability of SOA as a design style when designing a solution.

Recently, Web services has popularized the term SOA. The computing industry often uses the terms Web services and SOA interchangeably when describing either a design style or an implementation technology. It is important to know that Web services is not the SOA, but rather, a technology that follows the SOA style of design. SOA and its underlying principles have existed longer than Web services. Only recently has the Web services SOA gained so much attention. This is due in part to the significant industry support for Web services from organizations like IBM, Microsoft, BEA, Oracle, Hewlett Packard, and others. In the remainder of this module, the definition and attributes of SOA will be identified.

Although stated earlier, it is worth repeating that the proliferation of Web services, or rather a large-scale deployment of Web services, does not move an organization closer to an SOA. SOA requires planning and introduction of some new processes and technologies over time in the area of services management.
It also requires an understanding of SOA and what constitutes a service.
Introduction to SOA and Web services (2 of 6)

Explaining SOA to the business executive

Imagine the following scenario.

You are an IT architect working on a project within the organization when suddenly one of the non-technical business executives takes you aside. The executive is starting to see the phrase service-oriented architecture (SOA) more and more in the news and hearing it bantered about by the IT group. Since this appears to be more than a passing thought, the executive asks you the typical business person's most-often asked questions:

- What is it?
- Why do you need it?
- What does it do for you?
- What will be relinquished if you don't have it?

The executive sounds somewhat frustrated and asks if this is the next bright idea from the IT industry and wants to know what the IT group is doing about it.

How do you answer these questions? This module answers these questions and more.
Introduction to SOA and Web services (3 of 6)

Service-oriented architecture defined: What is it?

Service orientation is the integration of applications and information sources through the exchange of information based on common semantics or a vocabulary used to define the structure of information exchange. Service orientation enables loose coupling of service providers and service consumers, as there is no information to be shared initially between the two parties.

Service-oriented architecture is an architectural style, design style, and a design principle for application development and integration. SOA promotes business process orchestration of the enterprise-level business services using a distributed model consisting of disparate organizational, customer, supplier, and partner systems.

SOA is composed of a service provider, service requester, or the service consumer and an optional service directory, which together are leveraged to deliver services using application-to-application messaging for information exchange. The service provider creates a service and publishes descriptive information about the service in the service directory. The service requester queries the descriptive information in the service directory to locate the service and also gathers information about the service and the service provider. The service requester accesses the service hosted in the service provider infrastructure and, in turn, realizes the business value of leveraging the service.

Each element of a service-oriented architecture is described below:

- A service is a unit of work done by a service provider to achieve desired end results for a service consumer.
- Service provider systems are the providers of services accessible by well-defined and published interfaces.
- Service requester systems are the consumer of services accessible by well-defined and published interfaces.
- Service directory is a well-known directory of available services.
- Services are created and published by service providers and are made available on a suitable infrastructure for access by the service consumers.
- Service descriptions are created by the service provider and are published to the service directory.
Security and management are part of the overall SOA framework.

SOA is an architectural style intended to achieve loose coupling among interacting software assets or applications. The SOA framework is considered to be an evolutionary step in the future of application design and development. SOA enables the modeling of business problems in terms of discrete services, which can be securely interfaced and integrated with other applications over the Internet or other suitable network infrastructure.

The relationships between processes, services, and components are described by looking at each item. These relationships are illustrated below. Building systems using heterogeneous network-addressable software components is what SOA is about.

- Business processes get business value from things, such as software assets, for example, applications.
- Services describe what things do but represent a fundamental shift in thinking from the traditional approach of creating things.
- Components are what things are, not what things do.
- Deployment units or software artifacts are how things are constructed and deployed.

The graphic below illustrates the relationship between business processes and services. Simply stated, this means a business process comprises one or more business services. The business services are choreographed in a manner to fulfill a business process. Business services in turn implement one or more components that may or may not be implemented as a group. For example, an application can provide a service. Components are a collection of software artifacts or deployment units that are constructed for deployment on one or more nodes, such as platforms.

*Service relationship to component*
Service-oriented architecture is a natural evolutionary step from the object-oriented (OO), procedural, and data-centric approaches adopted for solution implementation until now. In fact, when creating the SOA system, individual services are typically implemented using one or more of these technologies.

Services are built from components that represent software artifacts or deployment units that are deployed on a platform.

Therefore, SOA is a way of designing software systems to provide services to either user applications or other services. SOA describes the components of the system and the interaction between the components. A well-defined, robust service layer is a key component of SOA.

The service layer and the components used to physically implement the service are illustrated.

The figure illustrates the service layer and the relationship between a services layer, its components, and
the business processes that consume the services. Both consumer and provider are illustrated in the figure.

Some key observations from this view are that processes or other services consume services. Components may easily lend themselves to service identification and exposure while others may require re-engineering or the use of wrapping technology to invoke a component as a service.
Introduction to SOA and Web services (4 of 6)

Understanding the SOA
SOA can be understood by examining three items:

- SOA building blocks
- SOA characteristics
- SOA design considerations

The SOA building blocks are the inherent components and the essential ingredients of SOA. The characteristics are the features of the SOA, and the design rules are the guidelines and the best practices for designing and building services in an SOA.
Introduction to SOA and Web services (5 of 6)

Understanding the SOA (continued)

SOA building blocks include the following:

- **Business level services** — Services are defined as business-level services that map closely to the real-world activities and business functions.
- **Infrastructure level services** — Services are defined as infrastructure-level services that do not contain business logic, but are necessary for the management of business-level services.
- **Services management** — Service management is a critical component of service-oriented architecture based solutions, where services are consumed and used in a collaborative way in a trusted or non-trusted environment. Services management provides both business process management of services (monitoring, metering, SLA, and QoS) and usage management of services (for example, routing, prioritization, and transformation). This also includes utilities that might be necessary for reporting or profile management.
- **Security** — Security provides authentication of service consumers (that is, applications and users) based on message authentication standards such as SAML and WS-Security. Authorization or access control of services and operations based on consumer entitlements that are also part of security.
- **Service directory** — Service directory provides a registry function for the discovery of services and a Web-based interface for administration, auditing, access control management, publication, and search.

SOA characteristics include:

- **Loosely coupled** — Uses a well-defined interface designed to expose business functions and data, but also to hide underlying implementation details from service requesters.
- **Shared services** — Allows individual software assets to become building blocks that can be reused in developing other applications (application assembly).
- **Federated control** — Federated and policy-based security, management, and deployment.
- **Standards based** — Leverages open standards to represent software assets as services (for example, XML, SOAP, WSDL, and others).

“SOA design considerations are essential to the design of a service.”


Each of the characteristics can be traced back to one or more SOA principles that provide integrity to the principles and the characteristics. These include:

- **Coarse grain** — The level of granularity is a statement of functional richness for a service. The more coarse grained a service is, the richer the function offered by the service. Although coarse grained is desired, services can be fine grained if there is business justification. The key to granularity is the reuse potential.
- **Interface-based design** — Interface-based design allows for the development of a service that is independent of the implementation platform, language, and logic.
- **Discoverable** — The ability for a service requester to discover a service provider is critical for the wide adoption of SOA design. If no one knows that a service exists, it is unlikely that the service will ever get utilized.
- **Single instance** — The single instance characteristic helps reinforce the notion that only one implementation of a service should be running. Again, this is usually the best practice-based design consideration; although, there may be cases where a particular design suggests a different approach to service instances.
separation of concern principle allows for a cleaner handling of the complexity, as well as a means to achieve adaptability and scalability of software systems. A major benefit of an SOA design is the separation of concerns to reduce the complexity of today’s enterprise systems.

- **Asynchronous** — Asynchronous communication is not required, but is consistent with the other service characteristics such as coarse-grained services and loose coupling. It would be completely legitimate to design an SOA solution that used synchronous styles of computing and still achieve coarse-grained services and loose coupling. Asynchronous communication offers the benefit of scaling through asynchronous behavior and queuing techniques.

- **Stateless** — Services neither remember the last thing they were asked to do, nor care what the next service is. Services are not dependent on the context or state of other services. This is not to suggest that a service cannot maintain state. However, the suggestion is that a best practices approach to developing services would be based on a stateless design.
Introduction to SOA and Web services (6 of 6)

Service-oriented architecture: Why do you need it?
The combination of SOA and Web services is very close to being just what companies have been looking for in order to:

- Realize IT’s long-promised potential
- Justify IT expenses and capital outlays
- Provide non-technical people a clear understanding of what IT does, how they do it, and their intrinsic value

Service-oriented architecture: What does it do for you?

- Saves money, time, and people
- Eliminates frustrations with IT
- Justifies IT investments
- Provides business executives with a clear understanding of what IT does and its value
- Eliminates IT’s 6-6 answer; that is, the project will take 6 months and cost 6 figures
- Provides a business and competitive differentiator

Service-oriented architecture: What will be relinquished if you don’t have it?
It is very likely that business transactions will be significantly easier with an SOA. An SOA could be the difference between the success and failure of the next:

- Department, intra-company, or inter-company merger
- Acquisition
- Divestiture
- Product or service rollout
- Business partner, customer, or supplier addition
- Geographical expansion
- Competitive onslaught
SOA summary

A service-oriented architecture consists of a small set of simple and ubiquitous interfaces to all participating software assets. These interfaces are encoded with only generic semantics. These interfaces also should be universally available to all interested and authorized providers and consumers. Descriptive messages constrained by an extensible schema are delivered through the interfaces. None, or only minimal, system behavior is prescribed by these messages. A schema limits the vocabulary and structure of messages. An extensible schema allows new versions of services to be introduced without breaking existing services.

The fundamental rules governing an SOA are:

- The messages exchanged between participating systems must be descriptive, rather than instructive, because the software system providing the service is responsible for solving any problems. Descriptive messages provide information about the service and the associated inputs and outputs. The providers are responsible for the how to; hence, the need for instructive messages does not exist.
- The vocabulary and the structure of the messages must be understood by all parties. This common understanding by all parties demands limiting the vocabulary and structure of messages but is a necessity for efficient communication.
- The message structure should be extensible.

An SOA may define a mechanism that enables a requester application to dynamically discover an application providing the requested service. The SOA introduces a set of concepts and ideas that are not evident in the traditional models. These concepts include:

- Component view of business services that can provide delivery of strategic business needs that require flexibility, value-add, and cost-effectiveness.
- Emphasis on the service operational management element based on services-based infrastructure.
- A business component implementation of service mediation and brokering.
- Enterprise view of business components, a sharp contrast to the traditional application functions that were tuned toward independent line-of-business business process requirements.
Web services defined (1 of 2)

A Web service is a software system identified by a URL, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML-based message conveyed by Internet protocols.

Web services are the most promising instantiation of service-oriented architectures due to the use of open standards, asynchronous implementation of decoupled systems, and a strong drive toward interoperability.

Web services relationship to SOA

- Some service-oriented architectures are implemented without using Web services, such as based on CORBA or WBI Message Broker.
- Web services is a leading technology choice to use for implementing SOAs today:
  
  a. Standards-based  
  b. Cross-platform and cross-language  
  c. Widely supported  
  d. Message-oriented  
  e. Faster tooling support speeds implementation of SOA

- There are many Web services implementations that are not SOA, such as connecting two heterogeneous systems directly together.

Web service components

The following diagram illustrates the key components of Web services.
The following items are components of Web services:

- **Service provider:**
  - Provides e-business services
  - Publishes availability of these services through a registry

- **Service registry:**
  - Provides support for publishing and locating services, such as telephone yellow pages

- **Service requester:**
  - Locates required services via the service registry
  - Binds to services via a service provider
Web services defined (2 of 2)

Web services base technologies
Base technologies for Web services are listed below:

- **SOAP**
  a. An XML protocol to invoke a *function* on a server to perform a given operation
  b. Request message is sent by service requester and optionally a response message may be sent by a service provider
  c. May be an asynchronous message, such as a notification

- **WSDL — Web Services Description Language**
  a. An XML vocabulary to describe service interfaces

- **UDDI — Universal Description, Discovery, Integration**
  a. UDDI servers act as a directory of available services and service providers
  b. A SOAP application used to update and query a registry for services

Example of how Web services are used to realize an SOA

*SOA design and Web services implementation*

---

**Integration**

**Design component**

**Use cases**

**Web services implementation**

**WSDL**

---

Perspectives on Web services
SOA is the map that will guide you down the road to competitive advantage.
Tailoring the SOA discussion for the business person

The beginning of this module introduced the concepts of an SOA and the Web services. It also answered some of the common questions around an SOA. The remainder of this module will now focus on applying this understanding of SOA and Web services to a business audience within your organization.
Defining SOA for the business person

Most companies today are being pressured by their customers and shareholders to drive growth by improving productivity and limiting costs in every aspect of the operation. But it is impossible to maximize efficiency if the company has rigid, expensive, proprietary IT systems. In fact, the most valuable thing a company can buy itself as an organization is flexibility, that is, flexibility to meet new market demands and seize opportunities before they are lost. To increase flexibility, the company has to look at its business as a collection of interconnected functions, discrete processes and services, such as check customer credit or authenticate user, and then decide which of those functions are core or differentiating, and which can be streamlined or even outsourced. If the company can mix and match these functions at will or on the fly, in response to changing business conditions, the company will have a tremendous competitive advantage in the marketplace. It is a powerful idea. But to achieve this degree of flexibility in the business operations, the company will need an equally flexible IT environment. It needs a service-oriented architecture (SOA).

As defined for the business person, an SOA is an application framework that makes it easy to reuse and combine the discrete business processes and services that make up the business. Think of it as a mosaic made up of individual functional components that can be arranged and rearranged. With an SOA, the company can build, deploy, and integrate applications and link heterogeneous systems and platforms together across the organization. While the SOA concept is not new, it has yet to reach its full potential, because most applications are still integrated using custom code, such as hardwired, and making it slow, difficult, and costly to rearrange the pieces in the mosaic. The industry has been waiting for a way to standardize the connections among all those components so they work the same way everywhere without requiring any programming. That standardization has finally arrived in the form of Web services.

Common components and common information from business processes can be reused multiple times in an SOA to give the company’s customer varied experiences. For example, information about a car is a static commodity, but its presentation can be dynamically based on perceived user preferences.

Dynamic presentation using the automobile

![Dynamic presentation using the automobile](image-url)
Simulating SOA for the business person

Service-oriented architectures are all about connections. A good analogy to explain SOA is the connections used in audio-video (AV) systems as shown in the figure below. (Web Services and Service Oriented Architectures, The Savvy Manager’s Guide, Your Roadmap to Emerging IT” ©2003, Douglas K Barry, pp 17-19.) Specifically, services in an SOA are similar to the components in an AV system. Web services are the connections between the AV components. For example, consider an AV system that has components that have been purchased over the years. Assume you want to add a DVD player to the system. The system has the usual cable box, receiver, VCR, CD player, speakers, and television set. One of the oldest components is the receiver, and the DVD has connections that the receiver can not handle, such as s-video and optical connections. It does, however, have the common three RCA connections. You decide at this point to change all of the connections in the AV system to RCA connections.

These components could be connected in different ways, depending on what you want to do. For example, you could set up your cable connection to go through your VCR or split the signal so that you can watch one program and record another. It was not too long ago that everyone had monolithic hi-fi or stereo systems. Then the industry settled on the various components in a stereo system, and later video was added. What does this have to do with SOA? It is all in the connections. Web services provides an infrastructure for creating connections similar to those with AV systems. And, just like AV systems, a user can assemble components in all sorts of ways because of those connections.

An SOA is essentially a collection of services. Services execute business processes. These services or business processes communicate with each other. The communication can involve simple data passing, or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed. Those connections are Web services. Note: The choice of the name Web services as the connection medium for “services” is somewhat unfortunate as it generates much confusion. When you see the term Web services, just think connection, or in the case of the AV example, you think of the RCA cable.

Audio-video SOA analogy
In the preceding figure, the entire picture represents the SOA. The services or business processes are the AV components and the cables numbered 2, 3, 4, 5, and 6 represent Web services, the connection mechanism.
SOA value roadmap

A recent survey of Fortune 500 companies indicated that over 80% had altered their business model in a given two-year period. Two-thirds of these, roughly half of the total respondents, claimed that this business change had been constrained by inflexible IT. In a survey by IBM Business Consulting Services, 90% of CEOs expect to transform their enterprise to become more responsive, particularly to customer demand, within the next five years. More than half expect to be engaged in significant company-wide transformation initiatives within two years.

A key part of many organizations' transformation programs is to reorganize information resources as substantially independent, reusable services that create an inherently adaptable environment. Common practice, however, is that business enterprises often take the easier choice by addressing a smaller set of objectives because it seems too difficult to do the job properly. Many organizations are now undertaking development of SOAs, but the probability is that most will result in suboptimal implementation because the relationship to the business requirements is not fully understood and the value proposition is not fully communicated. Consequently, business management may neither see nor realize the value of the SOA.

This module presents a method to explain the business value of SOA to your organization in terms a business person can understand and appreciate. Included is a checklist showing the business issues to cover, and following that, detailed text explaining each point. The checklist contains a set of logical business arguments to explain SOA value. It starts by examining architecture business fundamentals, works through process, then services or incremental delivery, and finally SOA. This education, coupled with customer and analyst references, yields the SOA business value package.

SOA value roadmap
The goals of this method are to help IT professionals:

- Show business executives that SOA can eliminate most of their frustrations with IT.
- Show that SOA can provide a way to quantify and measure IT business value.
- Stress that business units have to share the risk through a cooperative effort with IT.
- Show how a digital model of the business will provide business agility.
- Highlight current critical IT issues that the business side of the company needs to understand and appreciate. In discussions with customers, the seller would do well to keep these thoughts in mind and reiterate them as often as possible.
  - Architecture’s importance to the company
  - Architecture justification challenges and how SOA can solve these
  - The critical link needed between the IT group and the business units
  - A realization that the IT-business unit relationship needs to be immediately espoused if not currently in place
  - The importance of the business units accurately and succinctly describing their current or future business processes to IT

While this scenario provides information useful to you as you talk with the business executives in your company, you may not have time to go through the entire discussion. If time is limited, the following are considered the core issues to be covered:

- Need for an IT-business unit dialogue
- Creating a digital model of the business
- Aligning IT with business via processes and metrics
Enlightening a business person regarding SOA

One of on demand’s critical success factors is the close interaction between the IT group and the business units. Since SOA is the cornerstone of on demand, this partnership is an absolute necessity for a successful SOA implementation.

Currently in many firms, the IT organization functions as an order taker for the business. Business people, envisioning business improvement, make a request to the IT organization to build a new solution. IT proceeds to build the business request. Both sides stay within their own domains with a wall between them. This wall is used to throw projects to IT. The two sides may talk about the perceived benefits of the new system, but no one follows up on it and IT’s business understanding is too shallow for collaborative development of the business process. IT has long talked about alignment and partnership with the business, but with limited ability to discuss in business terms, IT has been largely unsuccessful in having strategic impact on the business. A company can experiment with a few Web services applications and become successful. However, the real competitive advantage will come when Web services is married to a full implementation of SOA. Therefore, it is imperative that IT and the business units fully cooperate and that each group is fully respectful of each other.

If a peer business relationship between IT and the business units cannot be forged, SOA cannot be successful.

Start SOA at this point — Educating business people on the many business benefits inherent in an SOA is the first step in solidifying an active working covenant between IT and the business units. This relationship is the foundation upon which everything else in SOA, such as the business agility, is built.

Checklist: Attaining business change agility
This section describes how to speak to the business person about SOA, benefiting IT.

SOA discussion checklist
Unravel the word "architecture" from its perceived technical underpinnings and explain the business realities of this necessary business function. Points to convey are:

- Architecture is an investment principle.
- Architects can never predict all the applications and business units that will use the architecture they create.
- Architecture is a wager.
- Architecture is a set of constraints.
- Architecture is a living process – never static – always changing.
- Architecture is hard to cost justify because cost justifying bets is hard.

Clarify the architect's "current" roles in providing this function which is crucial to business success. Roles to convey to the business person are the architect's requirement to:

- Identify potential architectural investments and cost justify them.
- Communicate the architecture and its maintenance processes to the business users.
- Create, deploy, manage, and update the company's architecture to support business needs.
- Resolve architectural and most likely business function conflicts, between business units.
- Monitor the architecture's effectiveness and work with business units to enhance it.

Explain why IT needs to realign around service definition, that is, business processes.
<table>
<thead>
<tr>
<th></th>
<th>Emphasize the need for a meaningful, constructive, respectful, and regularly engaged relationship between IT and the company's business units.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Show how service orientation will create a digital model of the business. Use a top-down approach to explain it.</td>
</tr>
<tr>
<td></td>
<td>Business happens via business processes.</td>
</tr>
<tr>
<td></td>
<td>Business processes tie to business measurements.</td>
</tr>
<tr>
<td></td>
<td>Business process steps are delivered via IT applications.</td>
</tr>
<tr>
<td></td>
<td>Business process steps align with business services.</td>
</tr>
<tr>
<td></td>
<td>Business services create a digital model of the business.</td>
</tr>
<tr>
<td></td>
<td>Describe how IT will align with business via processes and metrics:</td>
</tr>
<tr>
<td></td>
<td>Services push IT to understand business processes</td>
</tr>
<tr>
<td></td>
<td>Which leads IT to examine business process metrics</td>
</tr>
<tr>
<td></td>
<td>Which leads IT to understand the business in business terms</td>
</tr>
<tr>
<td></td>
<td>Giving IT a new way to prove its value: business results measurement</td>
</tr>
<tr>
<td></td>
<td>IT aligns with business via incremental delivery:</td>
</tr>
<tr>
<td></td>
<td>Massive business process change is too risky.</td>
</tr>
<tr>
<td></td>
<td>Business changes most effectively in increments toward a vision.</td>
</tr>
<tr>
<td></td>
<td>Process impact drives project prioritization.</td>
</tr>
<tr>
<td></td>
<td>Services change alignment with business.</td>
</tr>
<tr>
<td></td>
<td>SOA provides the glue between business processes, services, IT justification, and business value.</td>
</tr>
</tbody>
</table>
Explaining architecture for business people (page 1 of 2)

An identifying characteristic of an SOA is that it is a style of IT architecture. (What Do IT Architects Do? N. Jones and S. Mingay, Gartner Research Note, May 8, 2001.) While this may be stating the obvious, you should remember that an SOA has all the qualities, demands, and risks inherent in any type of architecture. Since a business person will most likely have a technical connotation for the word architecture, you must challenge this myopic view by explaining the importance of architecture to the business and why the business person needs to understand architecture from the business benefit perspective.

The benefits of architectural investments span many projects, not all of which may be foreseen at the time of the original investment. This definition of architecture helps us understand what architects do, as well as many of the challenges they face. Several observations are very important as they relate to architecture and SOA, in particular:

- Architecture is an investment in process, technology, and interface standard for the purpose of improving the organization’s capabilities, maximizing business agility, or reducing the cost of IT development and operations.
- Architecture is not a specific set of tasks or skills, because every organization will achieve its architectural goals in different ways. Architecture can be seen as an investment. The goal of the architecture group is to select and implement the most effective investments in standards, procedures, technologies, and interfaces to support the organization’s business goals and business processes. The architecture group’s deliverables will depend on the types of investments that are most appropriate, given those business goals.
- The goal of the architecture must be articulated as part of the architecture. An architecture intended to minimize the cost of ownership is not the same as an architecture intended to minimize time to market or to maximize business agility.
- Architects can never predict all the applications and systems that will use the architecture created. Increasingly, the goal of architecture is to support general organizational business directions rather than specific planned projects. Architecture is, therefore, a calculated risk. Architects bet that if everyone or at least a large proportion of the organization follows the architectural rules, there will be a future benefit, such as productivity, cost savings, business agility, or reduced market time. They hope that their architecture will have value, even for projects that were not envisioned at the time they created it. The number and type of risks that an architect can make will be constrained by many factors, especially the organization’s attitude toward risk and how much the architect knows or can guess about the future. The only way an architect can have any hope of predicting future requirements is if he or she is fully engaged with the business units.
- Architecture is a set of constraints. It is about the greatest good for the greatest number. Individual projects and programs follow the constraints because, despite some inconvenience, the organization as a whole will benefit. Organizations that have no concept of the common well cannot implement wide-ranging architectures. Tension is inevitable because it is unlikely that any architecture, such as a set of constraints, will be appropriate for the entire organization. Therefore, a conflict-resolution process, known as governance, must be part of architecture. Conflict resolution will be highly unlikely unless the architect is fully engaged with the business units.
- Architecture is a changing process, not a static document. Architecture evolves in response to changes in the organization's business goals and the available technologies.
- Architecture is hard to cost justify because cost justifying risk is hard. The true value of architecture is often only proven retrospectively; if system development and operations are cost-effective and the organization’s business goals can be satisfied easily, then the architecture was well designed and executed. In practice, organizations often have an architecture that is only as good as they can cost justify, which is not always the architecture that will be most beneficial from a business perspective. However, an SOA implementation has the inherent quality of tying IT costs with specific business processes that are tied to specific business measurements.
- Architectures are seldom universal. Different parts of the organization have different goals.
and each may need to support different architectures. Any architectural deliverable should be scoped to determine what is needed. A one size fits all architecture is seldom successful. For example, a common goal of architecture in areas, such as mainframe applications, is to promote quality and low cost of ownership. The goal of architecture in fast-moving business and technology areas, such as wireless applications, is usually to promote time to market with acceptably low risk. There is generally a trade-off between depth and breadth of architectural investments.

Architecture is a wager and bets are difficult to justify; however, SOA can provide IT justification and business value measures.
Explaining architecture for business people (page 2 of 2)

Clarifying architect roles
While an architect’s role is crucial to a company's success, in the past it was rarely evident to, or even noticed by, the business people. Within SOA, some architects expand their knowledge of the business processes of the company, becoming the bridge between IT and the business units and the linchpin to business agility.

Educating the business person to these changes requires an understanding of architect’s current, non–SOA responsibilities, as it lays the groundwork to elaborate on the expansion of the business duties the architect assumes in a new SOA environment. In fact, SOA architects will need to work so closely with the company’s business units, a job classification of “super business analyst” would be more appropriate and more accurately reflect their post-SOA involvement. Current architect tasks contributing to business success include:

- **Identifying potential architectural investments and cost justifying them.** This requires architects to seek information from their technology clients and their business associates to identify which architectural investments are appropriate.

- **Communicating the architecture and maintenance process to its users** and supporting users of the architecture. This may involve training, publications, Web sites, groupware tools, hotline support, internal consulting, or internal competency centers or Centers of Excellence.

- **Roadmap management.** Few organizations have the luxury of starting new when defining architectures. In practice, most architecture groups maintain a roadmap that documents the current physical architecture and the desired target architecture and defines how one is to evolve into the other.

- **Conflict resolution and exception handling.** Conflicts between the goals of individual projects and the goals of the organization are inherent in architectures. The absence of conflict is a sign of danger, meaning either the architecture is being ignored or there are unsatisfied opportunities for standardization. Exceptions will sometimes be required and should be handled by a defined process.

- **Deploying the architecture.** Most architectural deliverables are implemented by a process of communication, training, and mentoring. In the SOA world, these deliverables will reflect the company’s business processes.

- **Monitoring the effectiveness of the architecture and encouraging or enforcing compliance.** SOA is a way of attaining business agility. Therefore, it is imperative that architects regularly consult with the business members of the team.

As enterprises drive toward greater cross-regional, cross-line-of-business and cross-departmental integration, such as SOAs, architecture will become important to the success of the organizations at the same time they will be recognized as a valuable discipline. A clear definition of the business goals of architecture and the roles of architects are necessary components of an architecture function.
Putting it all together: Service-oriented architecture (page 1 of 2)

Service-oriented architecture is the technical foundation underneath service-oriented IT. Services provide the incremental building blocks around which business flexibility revolves, but services need a supporting architecture for their deployment, delivery, and management. With an SOA, as the business changes, you implement many changes by configuration within the SOA rather than by changing application code. You use data gathered by the SOA to monitor service operation, gain insight into the related business process, and guide and control decisions on where, when, and how to change, providing both IT-level and business-level control. The applications used by your customers, suppliers, and internal users leverage the SOA’s broad connectivity to many access channels to employ services where and when they are needed and with the required quality of service.

Business process competency is critical for SOA. IT must increasingly drive toward, design for, and deliver to business processes. If this can be accomplished, then SOA will identify IT projects, applications, infrastructure, and costs with the business processes they support. Key ways to infuse a business process mindset in IT are:

- Create a business architecture
- Instill a focus on business processes
- Identify IT leaders to lead process transformation dialogues with the business units
- Raise the visibility of every project’s process value
- Develop business smart IT architects

End of the beginning of implementing SOA — An IT organization is SOA-based when it has the following characteristics:

- SOA’s role is thought of as delivering two primary categories of value: business services and access channels for services.
- SOA pursues services-based design in all its delivery efforts as opposed to thinking that services are only part of the job, used only in certain situations for certain application connections.
- It designs nearly all functions as services. Since IT cannot predict how the business will change and how it will need to reconfigure services, IT needs to be ready for the business requirements. Processes executed as services give it the most flexibility.
- It defines all of its tasks around services.

When a change in business process no longer requires a change to application programming logic, you have a successful SOA; your company has attained competitive business agility.
Putting it all together: Service-oriented architecture (page 2 of 2)

Summarizing SOA value

SOA value roadmap goal #1

Because: Business happens via business processes

And Since: Business processes tie to business measurements

And with an SOA:
1) Business process steps are delivered via IT applications
2) Business process steps align with business services

Therefore: Business services create a digital model of the business

Yielding Business Agility & Competitive Advantage:
A digital model of the business is powerful because:
1) Creating new business processes when business changes is now merely choosing from an inventory of stable services
2) Provides a structure to collect, examine, and align business and IT metrics

SOA value roadmap goal #2

Because: Business services push IT to understand business processes

Which: Leads IT to examine business process metrics

Which Then: Leads IT to understand business in business terms

Giving: IT a new way to prove its value: business results measurement

The analysts and a large percentage of your customers realize that SOA is one of the most important business issues of the day. You need to be knowledgeable and skilled in its benefits and uses to assist them in their implementations. Quite literally, it is the foundation for successful business execution and growth for the foreseeable future.
IBM SOA environment with Web services and workflow

SOA Custom Applications

App 1
- WS-1
- WS-2
- WS-3
- Workflow Engine X
- WS-5
- WS-6

App 2
- WS-4

Modularized ERP Applications

WS-A
WS-B
WS-C
WS-D

Databases

Workflow Engine Y

SAP

CRM Siebel

Back-end Databases

WS-I

WS-II

Notes
- WS = Web Service
- WS-A = Web Service to TxHub / DFMS, embodied in Workflow Engine Y
- Large applications modularized into smaller services & Workflow
- ERP business components are externalized as Web Services
- Business Components are exposed as Web Services
- Service Bus: services integration, messaging, routing, transformations, mediation
Summary
Now that you have completed this module, you should be able to:

- Define service-oriented architecture
- Explain the relationship of Web services to service-oriented architecture

From your review of this module, you should have concluded that service-oriented architecture represents the future for information technology. IBM has the means and the ability to be a leader in SOA.

Much of IBM’s future is focused around on demand technology. SOA is the cornerstone of on demand, making its success critical to IBM.
References
The following resources provide further reading or sources of help.

- Official IBM definition of a service-oriented architecture as given by the FAQ section of the Project Sonata announcement April 21, 2004.
Test your knowledge

The online version of this lecture contains a quiz you can take to test what you learned in this module. Correct responses and your score are displayed after you answer the questions, but the scores are not reported or recorded. You may take the quiz as often as you want.

To find the quiz, open the online version of Introduction to service-oriented architecture to the Test your knowledge page in the module.