

Oracle's Approach to SOA: An Overview



ORACLE

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to explain the definition, rationale, and scope of:

- SOA Roadmap creation
- Strategy and planning
- Execution preparation

ORACLE

Agenda

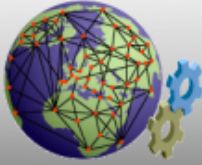
- SOA in Context
- Fundamentals of SOA Adoption
- Roadmap Creation
- Strategy and Planning
- Execution Preparation

ORACLE

1 - 3

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

What Is Service-Oriented Architecture?



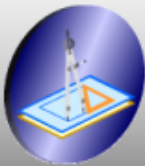
Web Services = SOA?

- *If not, why not?*



Is SOA a product?

- *A service bus?*
- *Some other product?*



Is SOA an architectural style?

ORACLE

1 - 4

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Definition: As its name implies, a *service-oriented architecture* (SOA) is an architecture, and that architecture is by definition associated with services.

Web service specifications define an interoperable platform supporting an SOA. However, although an SOA will likely be built with web services, building web services will not necessarily create an SOA.

Similarly, although there are software products that support an SOA implementation, using those products does not guarantee the success of that SOA implementation.

Service-Oriented Architecture: A Definition

SOA is a **strategy** for constructing **business-focused** software systems from loosely coupled, interoperable building blocks (called *services*) that can be combined and reused quickly, within and between enterprises, to meet business needs.

ORACLE

1 - 5

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

SOA is more than using WSDLs, and is more than just an application with a web service interface. SOA is a strategy—or an approach—that is business focused. To be successful, SOA must provide true business value. Without that value, services tend to be low-level utility services (such as logging and notification) that provide no real return on investment.

Traditional SOA principles such as loose coupling, interoperability, composability, and reuse are key traits of SOA, but only because they contribute to the business value of an SOA.

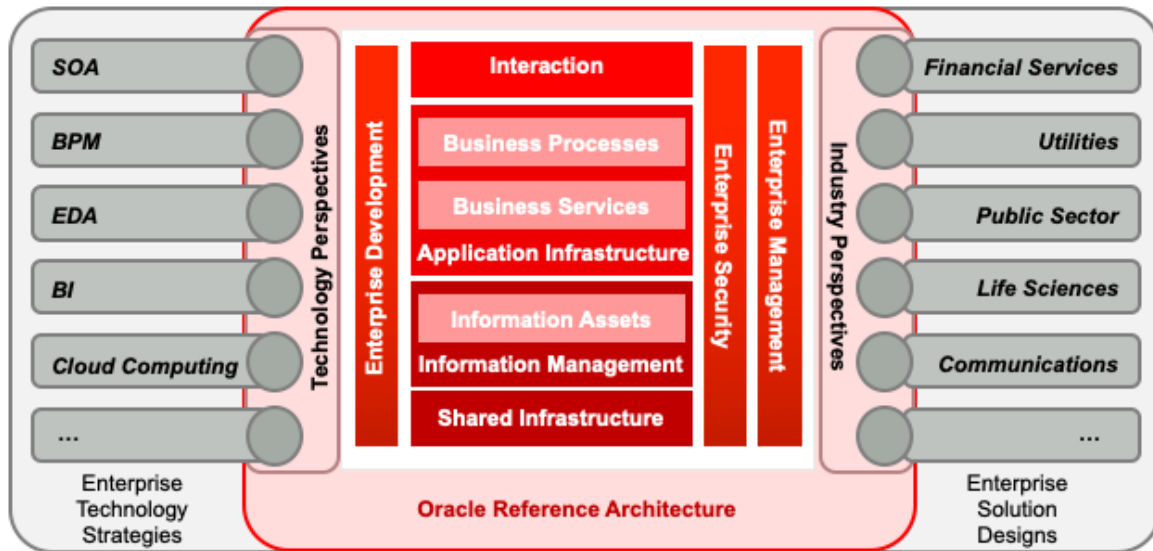
Within the scope of the architectural strategy, SOA describes an approach for enterprise systems development and integration that is both technology agnostic (that is, it operates across heterogeneous systems) and aligned with business imperatives. It provides loose coupling of coarse-grained components (services) for rapid and effective reuse of enterprise IT assets.

The ultimate goal of SOA is to facilitate the creation of new business solutions through the composition of services—without the need for complex programmatic code development that might otherwise duplicate existing capabilities. This leads to a more agile and efficient enterprise that can respond more rapidly to changing market and regulatory demands.

SOA is also an architecture-driven design discipline with the goal of increased interoperability (information exchange, reusability, and “compose-ability”), increased federation (uniting resources and applications while maintaining their individual autonomy and self-governance), and increased business and technology domain alignment.

As a strategy, SOA is broader than merely a reference architecture (RA) and should encompass engineering practices, governance, and other aspects of an enterprise engineering activity.

IT Strategies from Oracle (ITSO)



ORACLE

1 - 7

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

IT Strategies from Oracle (ITSO) is a collection of documents and supporting collateral designed to help organizations plan, execute, and manage enterprise architecture and IT initiatives. ITSO does the following:

- Provides ideas and recommendations for horizontal technology adoption and vertical industry solution design
- Positions products but is not product documentation
- Is endorsed by product development
- Is available at www.oracle.com/goto/itstrategies

The structure for ITSO is described in the top-level diagram in the slide. The scope comprises all Oracle products. Oracle Reference Architecture is the core of ITSO, as illustrated at the center of the diagram. On either side of ORA are Enterprise Technology Strategies (ETs) and Enterprise Solution Designs (ESDs).

Oracle Reference Architecture

In the IT industry, it is typical to see a different reference architecture for each technology strategy that is developed: for example, separate reference architectures for service-oriented architecture (SOA), for event-driven architecture (EDA), and for business intelligence (BI). Oracle Reference Architecture (ORA) defines a single reference architecture for designing, building, and integrating enterprise-class solutions based on modern technology from Oracle and other vendors. The reference architecture offers concepts and definitions, multiple architecture views (conceptual, logical, product mapping, and physical), principles, and guidance based on recommendations from technical experts across Oracle. It covers a broad spectrum of concerns pertaining to technology architecture, including middleware, database, hardware, processes, and services. ORA consists of a set of core documents that address fundamental architecture concerns, plus a number of perspectives that focus on specific IT strategies.

Enterprise Technology Strategies

The core of ORA includes elements that every enterprise architecture needs to have. On the left side of the diagram are the horizontal Enterprise Technology Strategies (ETS). These strategies offer guidance on the adoption of horizontal technologies for the enterprise such as SOA, BPM, and EDA. Each ETS extends ORA by adding unique capabilities and components provided by that particular technology. In addition, each ETS explains how to successfully execute on a strategy by addressing concerns pertaining to architecture, technology, engineering, strategy, and governance.

Each ETS includes *technology perspectives*. These perspectives are part of the ETS but are also part of ORA. They extend core ORA documentation with reference architecture content that is specific to a technology strategy. For each technology perspective, there is a *foundation* document that establishes the concepts and terms that are used throughout the ETS. The *infrastructure* document provides the logical perspective, the product mapping, the use cases, and the use case views using the set of products that is appropriate for that technology perspective.

Practitioner Guides

Practitioner guides are essentially how-to documents. A practitioner guide addresses specific aspects of a technology strategy that can be most troubling, frequently causing concern or setbacks. It explains how to pragmatically address the problem and the benefits and trade-offs of a recommended approach.

The content of each guide addresses topics specific to that ETS. For example, the SOA practitioner guides include the following:

- *A Framework for SOA Governance* (highlights a framework for defining and building an SOA Governance model)
- *Creating an SOA Roadmap* (guide and supporting material for creating an SOA roadmap)
- *Determining ROI of SOA Through Reuse* (guide and workbook for determining ROI of SOA through reuse)
- *Identifying and Discovering Services* (discusses an approach in identifying and discovering Services as part of an overall SOA Engineering effort)
- *Software Engineering in a SOA Environment* (details an engineering approach to

developing services, a maturity model, and method development)

Maturity Model

Each ETS also includes a maturity model, which measures maturity and adoption across eight common domains. It evaluates an extensive list of capabilities and measurement definitions. It helps to visualize current state and plan incremental improvement strategy. The maturity model captures the best practices for a particular ETS. It identifies the things that you need to do to be successful.

Method Development

From a change-management point of view, it makes sense for an enterprise to continue using whatever engineering method it already has in place. However, for a given ETS, there are pieces that change. For that reason, each ETS addresses method development, but only to focus on the pieces that change with the ETS. These are the process fragments or control points that need to be applied. For example, an important subject in SOA is identifying services. The ETS describes how to manage requirements in a way that yields enterprise requirements to develop services.

Enterprise Solution Designs

Along the right side of the diagram are the Enterprise Solution Designs (ESDs). Enterprise Solution Designs are industry-specific solution perspectives based on ORA. They define the high-level business processes, business functions, and software capabilities in an underlying technology infrastructure that are required to build enterprise-wide industry solutions. ESDs also map the relevant application and technology products against solutions to illustrate how capabilities in Oracle's complete integrated stack can best meet the business, technical, and quality-of-service requirements within a particular industry.

Industry-Specific Reference Architectures

An industry-specific reference architecture provides a high-level representation of the current state and desired state of the reference architecture that is specific to an industry or industry segment. It is based on high-level ORA layering and industry-relevant components. It maps applicable Oracle products to the architecture.

Industry Solutions

An industry solution is more specific than an industry-specific reference architecture. It addresses particular problems in a vertical industry (for example, smart metering for the utilities industry). It describes specific business processes and specific entities.

Technology Patterns

A technology pattern is the overlap between an ETS and an ESD. For example, suppose that a company is creating an industry solution and one of the core tenets of the industry solution is service orientation. The technology pattern then says that this is a key portion of the solution. It is a reference back to the ETS.

Agenda

- SOA in Context
- **Fundamentals of SOA Adoption**
- Roadmap Creation
- Strategy and Planning
- Execution Preparation

ORACLE

Common Barriers to SOA Success: Sample SOA Challenges

- Lack of tools and infrastructure
- Standards proliferation
- Standards compliance
- Business–IT relationship
- Multiple siloed SOAs
- Lack of best practices
- Organization friction
- Confusing priorities
- Enterprise vs. LOB decisions
- Lack of SOA Roadmap
- Funding
- Charge-back models
- Lack of appropriate service
- Engineering approach
- Maturity of web services standards
- Service sprawl
- Registry sprawl
- SOA portfolio management
- “Right-click architecture”
- Culture change
- Change management
- Lack of appropriate operational processes
- Lack of skills and experience
- Hype vs. reality
- Track and communicate progress
- Policy enforcement (automated, manual)
- Fragile SOA implementation

ORACLE

1 - 11

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

The slide lists a sample of the challenges that have been faced and addressed over the many years that Oracle has been developing and executing SOA initiatives. Note that not all of the challenges are specific to SOA.

Service sprawl occurs when, due to a lack of governance or knowledge, an enterprise ends up with hundreds of services that are not being reused—or services that are 90% identical but have only one consumer each. Some companies approach SOA as a technology-only problem and buy a registry. But with services in a registry, you have services that are documented but still not being reused.

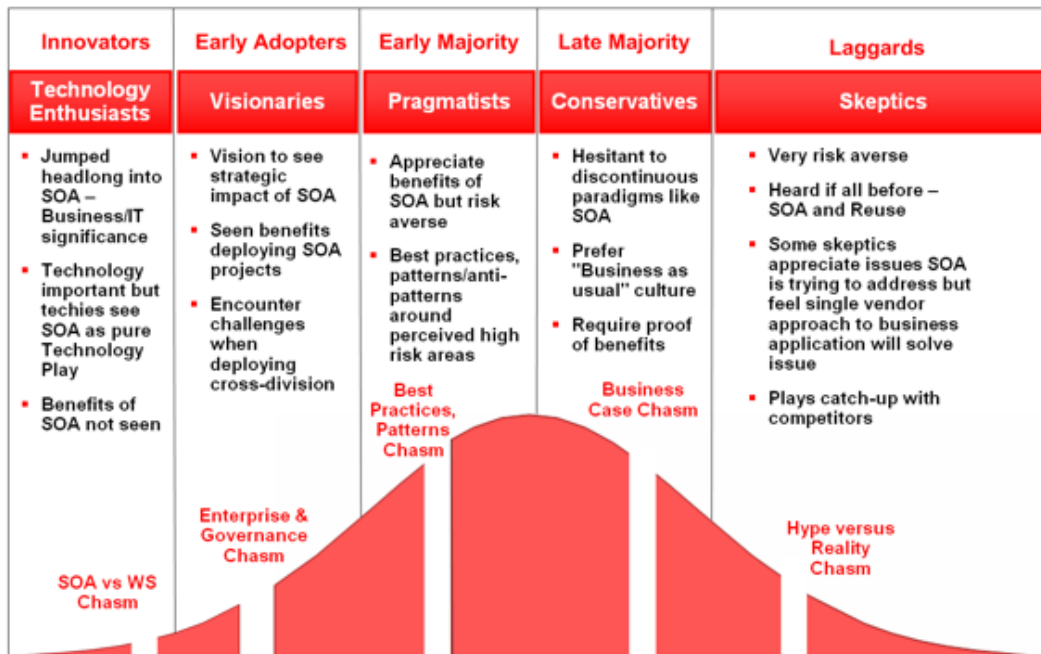
Right-click architecture refers to IDEs that make it easy for users to look at some code and then say, “Here’s an EJB,” right-click to confirm, and produce a web service. It is important to understand that this is not SOA.

Standards compliance highlights the importance of defining standards to support a reference architecture, but there is also the importance of making sure people actually follow their standards in the first place. Companies frequently develop a good reference architecture. But after creating it once, they never maintain it, and after a two-year period, web security standards have moved on and everyone bypasses them.

None of the examples listed in the slide are more critical than the others—these are just examples. In a real-world context, you would identify specific challenges (typically a subset of

a list like this) that are most relevant to your organization.

SOA Adoption and Organization Characteristics: SOA Chasms



ORACLE

1 - 12

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

The type of organization and its approach to (and adoption of) SOA can determine the type of challenges that it will encounter when attempting to execute an SOA program. Identifying your own organization's characteristics assists in identifying an approach to address these known challenges.

For example, organizations of "technology enthusiasts" tend to jump immediately into SOA without fully understanding its business and IT significance; SOA is seen as pure technology and its benefits are rarely seen. We call this the "SOA Versus Web Services" chasm because these organizations tend to think that SOA is purely about Web Services and its associated specifications, which produces a WSOA (web service-oriented architecture).

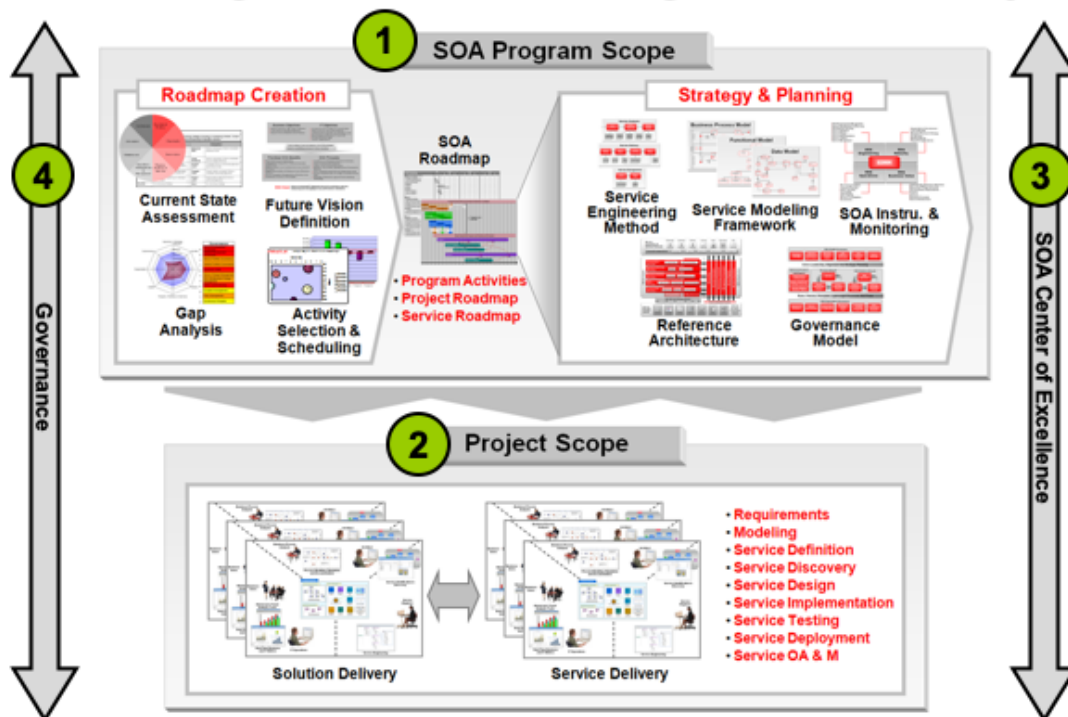
Similarly, other chasms are defined that correspond to other organizational characteristics, such as Governance, Best Practices, Business Case, and Hype vs. Reality chasms.

SOA planning needs to ask and answer the question, “What kind of characteristics does this organization have?” Large organizations with multiple lines of business can reflect all of these chasms. For example, human resources (HR) employees may be laggards and skeptics (“We have PeopleSoft”) and may not be interested in anything else. Another group may comprise early adopters.

As soon as one has described the characteristics of an organization’s employees, one can begin to understand where the organization should start and also what types of problems it might have. For example, it can be said that the early adopters understood SOA, but they did not give a high enough priority to the importance of governance.

The upcoming lessons of this course explain how to fill each of these chasms as part of discussions of SOA fundamentals, governance, and service engineering.

SOA Program and Project Scopes: Creating Focus and Setting Accountability



ORACLE

1 - 14

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Recall that Oracle's SOA ETS topics cover the entire life cycle. When you are looking at ETS, it is useful to consider two primary scopes: the program scope and the project scope. The activities in these scopes are grouped into three major categories:

- Roadmap creation
- Strategy and planning
- Execution

1. Program Scope

The program scope describes the scope of SOA adoption within the enterprise. Within this scope, the primary focus areas are the roadmap creation phase and the strategy and planning phase. As its name suggests, the roadmap creation phase focuses on assessing the current state of the enterprise in respect to its SOA goals and the maturity of the capabilities required to execute SOA successfully. The strategy and planning phase concentrates on defining a number of key frameworks, including an SOA reference architecture, a service engineering method, and an SOA governance model. The main artifact at the end of the program scope is an incremental SOA implementation roadmap that shows the build-out of the infrastructure, the solution roadmap, and the services roadmap.

Many companies that implement SOA will focus primarily on the project scope because funding is at that level. A company may have several projects, each of which is approved and each of which is done in an SOA manner. However, unless the project teams communicate well, when they want to start sharing services a year later, they will encounter problems.

To succeed, there are activities that need to be done at both the program level and the project level. At the program level, there are two primary areas of focus. The first is roadmap creation, which includes the current state, future vision, and gap analysis, as well as activity selection scheduling. The roadmap identifies the kinds of activities one needs to know at the program level. It describes what the project roadmap looks like and what the service roadmap looks like. It is important to focus on the strategic side before starting to execute on the project side.

These program activities include strategy and planning activities applied to engineering and modeling methodologies and standards. It is not necessary to complete every program activity immediately, but it is important to address those that apply to the project pipeline that is in place.

2. Project Scope

At the project scope level is the execution phase. The execution phase is where enterprises begin to execute their incremental implementation roadmap and start to deliver value to the business. The execution phase covers the different life cycles of delivery of solutions, the delivery of services, and the associated service infrastructure.

3. SOA Center of Excellence

As highlighted in the slide, there are many different roles and needs that are required at both the enterprise level and the project level. A common approach to support these enterprise and project needs has been by the definition and utilization of an SOA Center of Excellence (CoE). There are many different SOA CoE deployment and adoption models to choose from. It is important to select the appropriate model to fit for your organization.

4. Governance

A key pillar for SOA success is having an appropriate SOA Governance model that encompasses not just technology but the people and process aspects of governance to cover both the enterprise and project scopes.

End-to-end governance should not be an afterthought. It should really be there from day 1. It doesn't have to be overbearing, and it doesn't have to comprise committees of people and thousands of control points. It is about having just enough governance for the roadmap that you are creating. A common approach for early adopters is to assemble an SOA COE and have that COE act as the governance body.

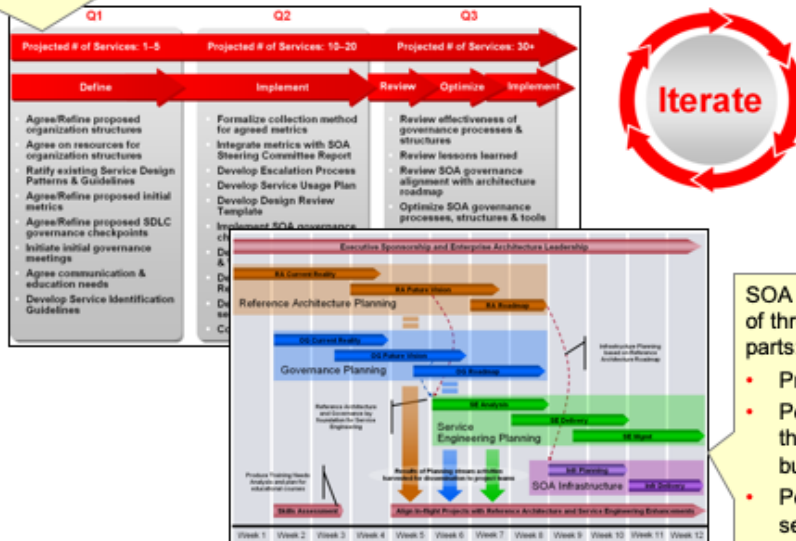
Agenda

- SOA in Context
- Fundamentals of SOA Adoption
- **Roadmap Creation**
- Strategy and Planning
- Execution Preparation

ORACLE

SOA Roadmap: Providing Guidance and Coordination

Multiple projects progress in parallel and remain coordinated.
Typical time horizon of 2–3 years
Regularly reviewed and updated
Incremental improvement and course correction



SOA Roadmap consists of three fundamental parts:

- Program-level efforts
- Portfolio of projects that build specific business solutions
- Portfolio of shared services

The slide describes the typical SOA Roadmap timeline as two or three years, but the actual timeline depends on the enterprise. The preferred timeline will be linked to the project funding and to the major milestones to be attained .

The near-term portion of the roadmap should be detailed. The medium-long-term portion should be a higher level because as projects progress, some correction will be inevitable as a result of knowledge gained and errors corrected. As the mid-term becomes the near term, more detail should be added. Those additions will reflect knowledge gained from the program to that point.

The goal is to have multiple projects running in parallel. For example, a company might employ a double-down pattern. In that scenario, the company begins with one project. After success with one project, the company’s planners would double up to two. When they were happy with two, they would go to four. If they had problems with four, they would go back to two again until they had worked out the problems. Otherwise, there are too many moving parts and too much project management.

There are dependencies among the program-level activity, the individual projects, and the individual services. Suppose that you say, “Here are X number of services that we want to produce.” In a pragmatic approach, you want to make sure that they are going to be consumed at least twice, and you must have the projects to support that. Otherwise, you will build services that may not be consumed for two years. A portfolio approach to services is very important, but it must be done in conjunction with projects, applications, and project management as well.

Agenda

- SOA in Context
- Fundamentals of SOA Adoption
- Roadmap Creation
- **Strategy and Planning**
- Execution Preparation

ORACLE

Planning an SOA Reference Architecture

Establishes an SOA **reference architecture** (an architectural blueprint) to:

- Guide SOA projects toward architectural consistency
- Steer existing siloed architectures on a convergence path toward an enterprise-wide services and infrastructure standard for SOA

Modified 4+1 Approach for Documenting Reference Architecture

- Depict the various layers of the architecture as well as high level components and their relationships

Logical

Process

- Illustrates the way in which components and layers of the architecture interact with each other
- The process views illustrate process flow in the architecture

- Illustrate the various software infrastructures used to implement the architecture
- Provide a sequence of software



Enables enterprises that have:

- **No** clearly articulated **architecture vision** for their SOA
- A reference architecture defined at **too high a level**
- **No roadmap** to implement their reference architecture

ORACLE

Oracle Reference Architecture (ORA) is a *foundational* reference architecture. It is not intended to be duplicated, but rather it enables an enterprise-specific reference architecture to be built in a timely manner—far faster than building a reference architecture from scratch.

The reference architecture described in this course is foundational. The first step is to assess your current situation by answering the following questions:

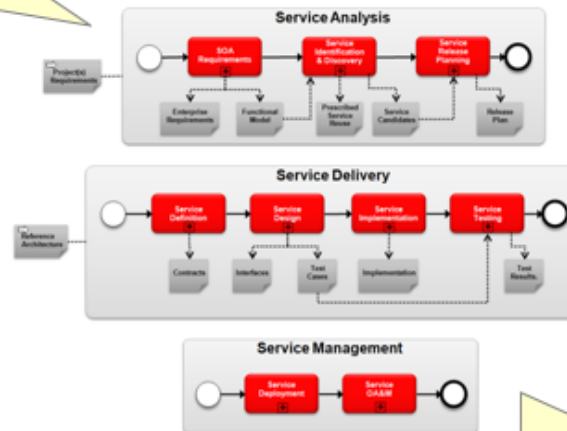
- What do you do today?
- What does your infrastructure look like?
- What are your capabilities?
- Where do you want to be after building and implementing a roadmap?

With the answers to those questions, you then build a roadmap according to your existing infrastructure, the point where you want to go, and ORA. So reference architecture planning is part of the two-to-three-year roadmap, because a reference architecture will not be built out on the very first project. It is too complicated, and there may be capabilities that are not required. However, it is important to know what should be built first. For example, do you need a service bus first or do you need to have some sort of service registry first?

Your organization needs a reference architecture. You need to plan for it and have an associated roadmap that covers two to three years. To accelerate that process, use ORA as a starting point. Then take it to the next level by developing your roadmap and planning the incremental steps to complete your current reference architecture as needed.

Service Engineering Framework: Analysis, Architecture, Delivery, Management

Defines service engineering disciplines that are required to deliver services **consistently** and **effectively** across projects while **complying** with architecture standards to enable agility and reuse



Enables enterprises that:

- Lack a clearly defined **service engineering process**
- Are **struggling to adapt** their software engineering process to align with SOA

ORACLE

1 - 22

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

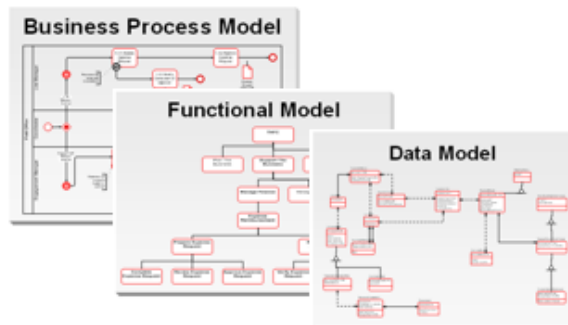
A service engineering framework introduces a discipline to effectively deliver projects and services within an SOA. It is a comprehensive set of processes, associated principles, and best practices that combine to provide a consistent approach to engineering and delivery across the services life cycle.

This framework addresses the service engineering requirements that are not covered by traditional methodologies. It is intended to work with an enterprise's existing methodology, thus minimizing disruption. It separates out the unique challenges put forward by enterprises adopting SOA and provides a framework for not only removing the hurdles they have been struggling with, but helping make their SOA more effective.

The framework is organized into three distinct areas: the analysis, delivery, and management of services. Of the eight areas listed in the slide, upcoming lessons will present four of them: requirements, identification, service definition, and service design. Each of these areas is a specific strategic activity that has a role at the program level.

SOA Modeling Planning

Defines a modeling strategy to **support** SOA initiative by **formalizing** SOA concepts in the form of **meta-models**, as well as the relationships between models and the steps in the delivery process where they are created



Service Modeling



Enables enterprises that:

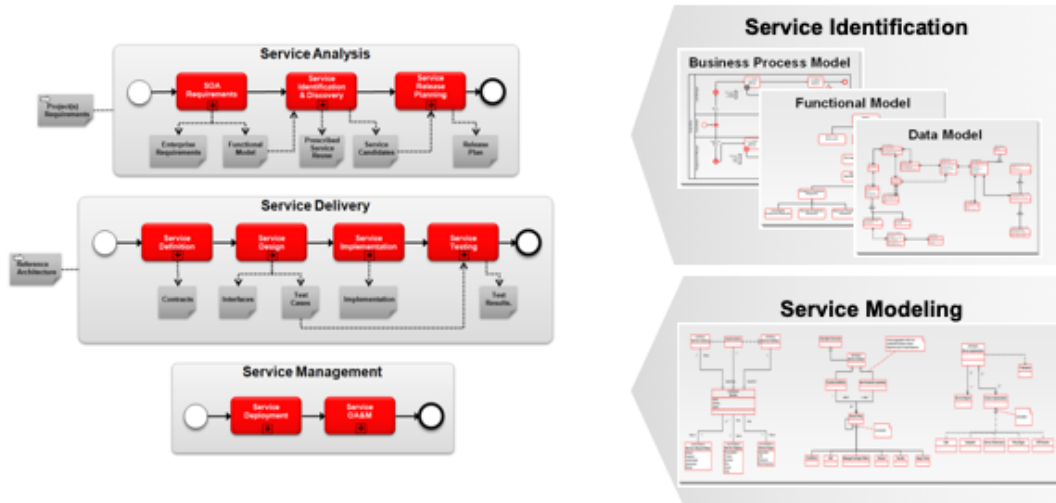
- Encounter **communication** challenges
- **Lack** a **holistic** modeling approach to aid service engineering
- Require a **service meta-model** to achieve the next level of SOA maturity
- Need to serve **various stakeholders** with differing modeling requirements

ORACLE

Service Modeling Planning: Building the Right Services

A service modeling framework:

- Assists in **identifying service candidates**
- Facilitates downstream service engineering

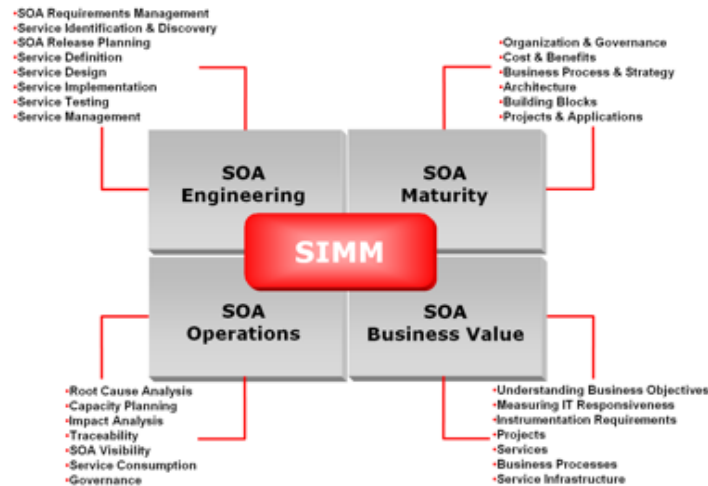


ORACLE

Many enterprises have a rigorous software development life cycle (SDLC), and modeling is a key aspect of this life cycle. For those enterprises, it is useful to have a modeling approach that service identification can use. That modeling approach may include a service meta-model, an end-to-end service model, and an end-to-end method.

SOA Instrumentation, Management, and Monitoring: Essential Measurements to Manage and Improve SOA

- Emphasizes **instrumentation** as a discipline that is required to more **effectively** implement an SOA strategy
- **Collecting, aggregating, collating, and presenting** KPIs to improve engineering, operational, and business aspects of SOA



ORACLE

1 - 25

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

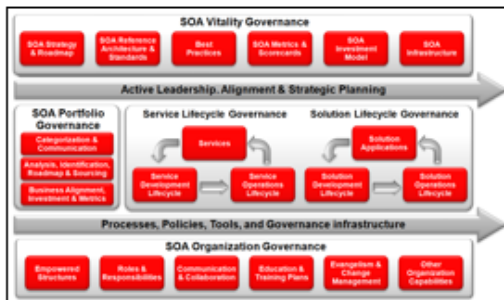
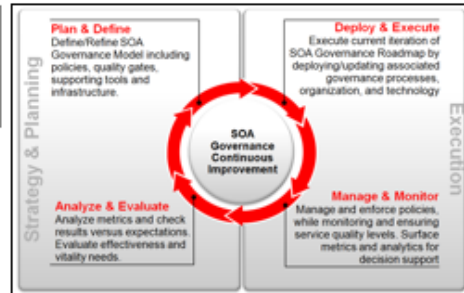
It is important to collect, aggregate, collate, and present key performance indicators to improve the engineering, operational, and business aspects of SOA. These concerns should be considered in the planning, modeling, and engineering stages to ensure that the deployed services and composite applications—and the processes used to create them—can be monitored, managed, and improved.

Metrics can be put into four distinct categories. One is the SOA maturity model. The others are SOA engineering, operations, and business value.

Do not look at metrics individually. Instead, you should identify the benefit that you are trying to achieve or address, and then work your way backward to the key performance indicators (KPIs) and metrics that you need. It is likely that you need to aggregate and collate a number of metrics to get your answer. You should not simply say, “This service has been reused 10000 times, and therefore it has great business value”—although it was merely a connectivity service.

SOA Organization and Governance Planning: Establish an SOA Governance Framework

Defines a **consensus**-based and **pragmatic** SOA Governance model covering organizational **structure**, governance **policies**, operational **processes**, tools, and guiding principles to govern and realize the true benefits of SOA



Enables enterprises that:

- Lack an SOA Governance model to aid in the adoption of SOA
- Encounter employee resistance to adopting SOA
- Are not realizing SOA benefits
- Have **uncontrolled** service creation

ORACLE

Because every enterprise is different, there is no single model of good SOA governance. Oracle has a governance framework that has two aspects:

- The actual baseline model
- A continuous feedback loop

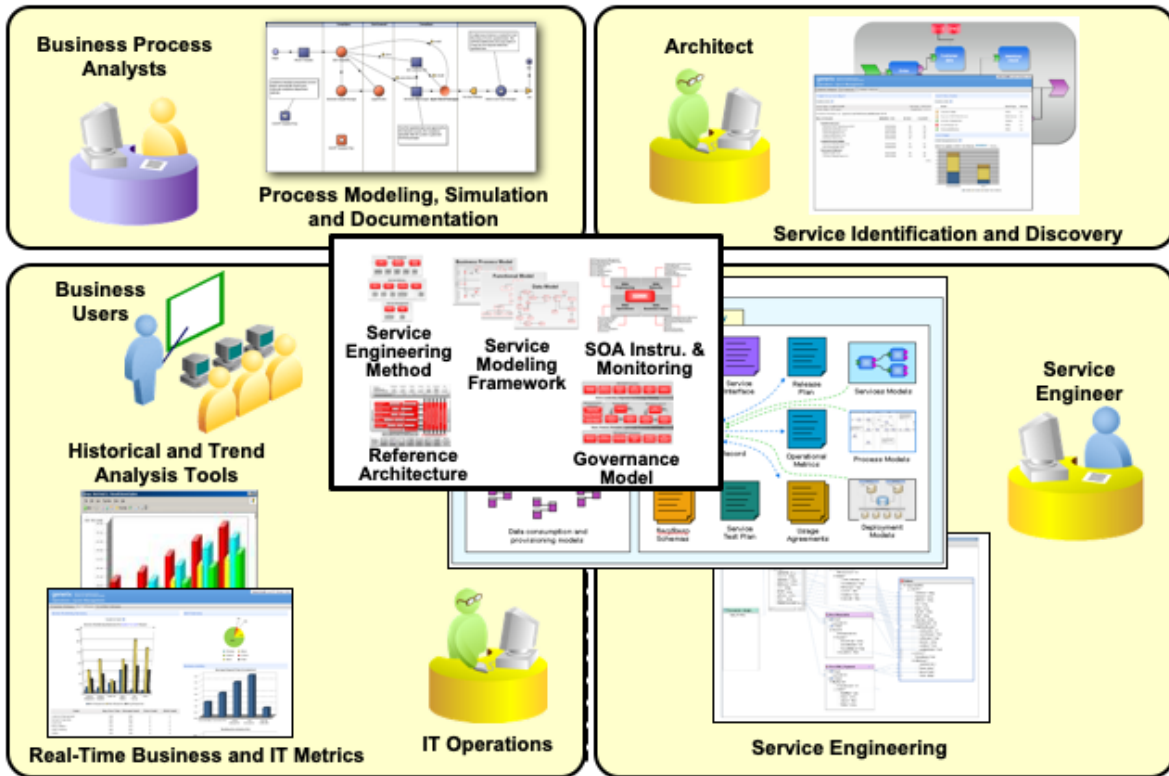
It is important to understand that governance is not a one-off project. It is something that you start, and it continues forever. The SOA Governance Continuous Improvement Method is a definition-improvement feedback process to define a focused and customized SOA Governance model.

Agenda

- SOA in Context
- Fundamentals of SOA Adoption
- Roadmap Creation
- Strategy and Planning
- Execution Preparation

ORACLE

SOA Execution: A 360-Degree View



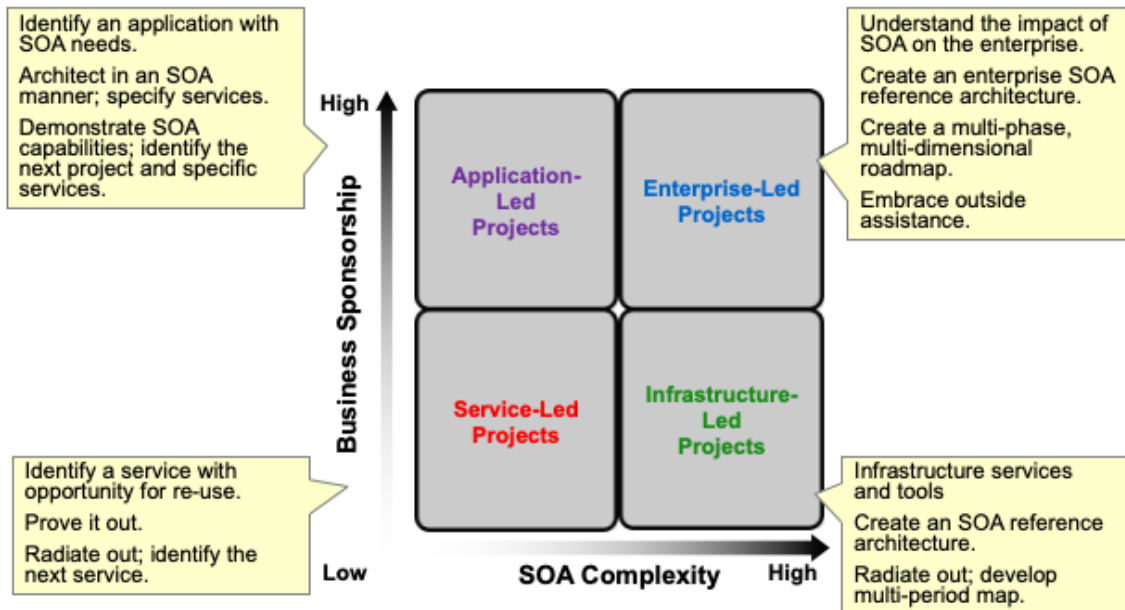
ORACLE

Good communication and access to information are often among the missing pieces of an SOA initiative. When you execute SOA, it is important to have a 360-degree view of the whole software development life cycle (SDLC) of the program-level activities. Rather than just building one big application, an SOA initiative involves building multiple applications and potentially hundreds of services. Each service is made up of a contract, an interface, and a payload. As work is handed from one department to another and from one person to another, a missing part can become a key issue.

The execution phase covers the different life cycles of delivering solutions and services and the associated service infrastructure. An enterprise repository provides a single source of truth for the SOA portfolio and manages SOA assets, projects, and associated metadata. With a repository, you can make rapid informed decisions about dependency tracking, impact analysis, usage tracking, and compliance.

Within the life cycle, analysts capture requirements and perform process, functional, and data modeling. Architects then identify and discover the appropriate services and define the associated service contracts. Service engineers can design, test, and deliver the services. IT Operations then provides facilities to provision, monitor, and manage services. The life cycle is then complete. Business and IT users can utilize tools to visualize historical and trend analysis. That information gets fed into planning for version 2.0 of a service. Version 2.0 of an application is usually released after 12–18 months. But a service might be versioned quite frequently.

SOA Entry Points: What Is the Appropriate Starting Point?



ORACLE

1 - 29

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

The slide illustrates the four options available to an enterprise that is starting SOA from scratch. The entry points can also be used by companies that have already embarked on SOA (or think they have) but may want a more structured approach. This could also include companies that have multiple siloed SOA initiatives or want to reset their existing SOA initiatives. These alternatives are not mutually exclusive. Many organizations will choose a hybrid of them.

In the bottom-left portion of the diagram is the service-led project. In that scenario, the organization focuses on a key service that is going to be used frequently and then reused by multiple projects. After one or two projects that use it and prove it out, the service can be radiated out to multiple lines of business.

For example, each of the lines of business in an enterprise may have their own credit card authorization to different vendors. In a service-led project, you work with one line of business, take their code, service-enable it, test it, operationalize it, and put it into production. When that line of business is satisfied, you go to the next line of business and roll out that service. Slowly the service is rolled out across each of the lines of business. As the slide shows, this approach has low complexity because it is just one service. And this service does not require significant business sponsorship from a funding perspective because it is likely that the code already exists—and more than one project actually needs it.

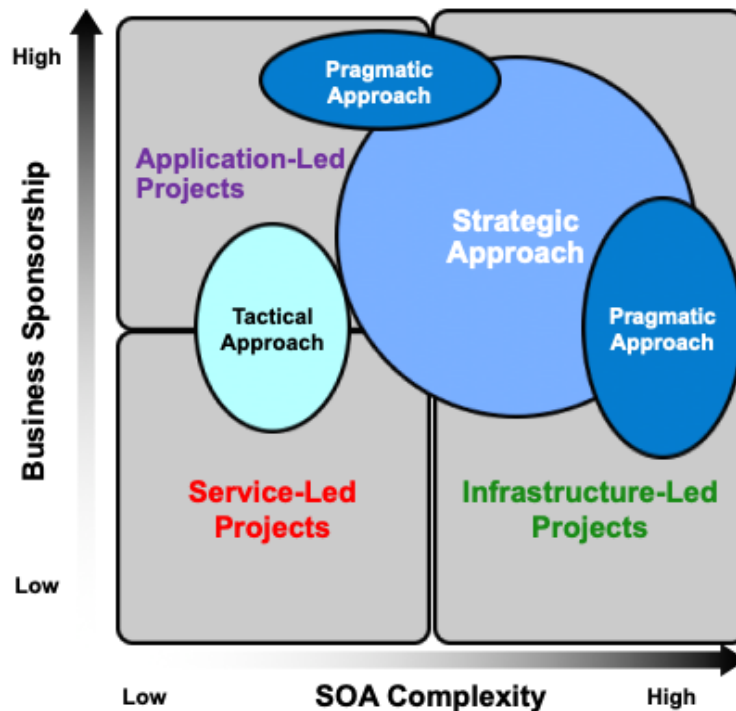
The second option is the infrastructure-led project. This option is for people who are looking at SOA more strategically. It is more complex because it is a far-reaching project, built with the following goal statement: “We need to create an infrastructure that multiple lines of business and multiple sites may potentially need to use.” In this scenario, you need to build a reference architecture, and then you need to actually deploy the infrastructure accordingly. However, in this scenario, more money is involved because there is more up-front planning.

In application-led projects, begin with the project pipeline, where there are projects that have (hopefully) been approved and have an associated funding or budget approved. If possible, you architect the projects in an SOA way, specifying SOA and its benefits. Those benefits are used to convince other lines of business: “These are the benefits we saw for SOA.” That approach is usually well received because the business sponsorship (the funding) has already been approved.

Finally, there is the enterprise-led project. In a way, the three other project types should be moving toward this over time. However, it is a difficult starting point even for companies that truly understand the business environment and the IT benefits of SOA. There is a temptation to try to achieve everything at once—planning every service and every organization, as well as all governance, engineering, modeling, and infrastructure. The result is a large, complex three-year plan with a huge budget. This model is more likely going to be suitable for those companies that are trying to correct or improve what they have today or those that seek to slowly radiate out their SOA initiative in an enterprise- and infrastructure-led project.

It is always important to know what makes sense in your organization and where to start. For example, if yours is an infrastructure-led project, obviously you need a reference architecture. If your company is going to do a service-led project, you may want to look at some service-engineering aspects.

SOA Entry Points: What Is Your Appropriate Starting Point?



ORACLE

1 - 31

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

In the diagram in the slide, we illustrate the engineering method with three approaches: tactical, pragmatic, and strategic. Each approach has its advantages and disadvantages. Each has its challenges.

A tactical approach includes elements of both service-led and application-led projects.

A pragmatic approach is often the best choice. It implies that you think strategically but act tactically. When you do infrastructure-led projects, the bigger picture should be part of the planning as well. The same is true for an organization doing a number of applications.

A purely strategic approach involves a great deal more work. It includes much up-front planning as well as the need to address all three of the other starting points (applications, services, and infrastructure).

SOA Strategy and Planning: Work Streams

Level of		Approach			Work Streams
Scope	Effort	Strategic	Pragmatic	Tactical	
Program	Roadmap Creation	Yes	High Level	No	<ul style="list-style-type: none"> ▪ SOA Roadmap Creation ▪ SOA Reference Architecture Planning ▪ Service Engineering Planning ▪ SOA Modeling Planning ▪ Organization and Governance Planning ▪ SOA Center of Excellence ▪ Program Management ▪ Employee/Partner Enablement
	Strategy and Planning	Yes	Partial	No	
Project	Execution	Yes	Yes	Yes	<ul style="list-style-type: none"> ▪ Service Infrastructure Realization ▪ Service Oriented Project Delivery ▪ Service Realization ▪ Project Management

ORACLE

1 - 32

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

In an SOA implementation, there are program-level activities and project-level activities.

There are three approaches to adopting and executing an SOA initiative. Within the program, there are two main activities: roadmap creation and strategy and planning. Within the project, there is execution.

The table in the slide reveals the work that project managers need to do in each focus area. With a strategic approach, all three efforts are required, including a detailed roadmap, strategy and planning, and execution. A pragmatic approach requires some high-level roadmap creation, as well as partial strategy and planning because of the need to prioritize depending on the pipeline. Once again, execution is required.

With a tactical approach, only execution is required. Although it is tempting to start with this approach, it is better to select one of the other approaches. It is not a good idea to proceed with no program and no planning.

The right column in the table lists work streams. These are the items that a project manager needs to put into a Gantt chart. The red bulleted items are strategic activities described in more detail in ITSO documents. All of those items are needed for full strategic planning. Partial strategic planning includes a subset of those items.

Quiz

In this course, SOA is defined as:

- a. Using web services
- b. Using a program that produces web services
- c. An architectural style
- d. A strategy

ORACLE

1 - 33

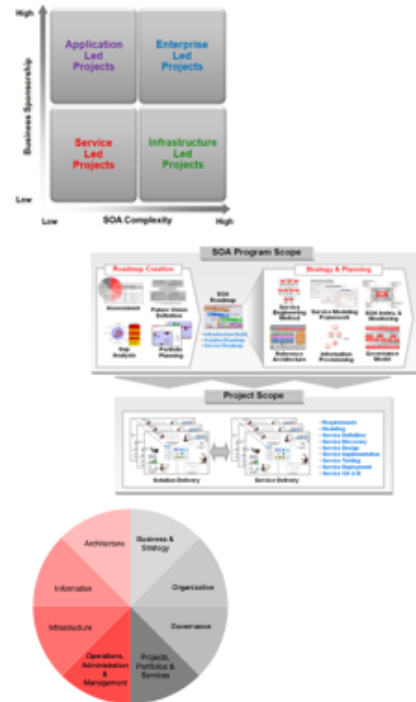
Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Answer: d

SOA is a strategy for constructing business-focused software systems from loosely coupled, interoperable building blocks (called *services*) that can be combined and reused quickly, within and between enterprises, to meet business needs.

Summary: Oracle's Approach to SOA

- 1 Pick your entry point.
- 2 Plan with a vision of 2–3 years.
- 3 Execute project by project.
- 4 Approach SOA on eight dimensions.
- 5 Construct service infrastructure incrementally.



ORACLE

1 - 34

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

The key points for this lesson are the following recommendations:

1. Pick the entry point that makes the most sense for your organization.
2. Plan a vision of 2–3 years: Have a roadmap. Remember that the success of the SOA initiative is measured by the realization of the prioritized SOA benefits identified in the Future Vision Definition phase (this phase is in the Program Scope portion of the diagram in the slide titled “SOA Program and Project Scopes: Creating Focus and Setting Accountability”).
3. Execute project by project. This pragmatic approach is desirable when matching plans to typical funding models.
4. Approach SOA on all eight dimensions. Look at each of these capabilities and best practices. There is no point in having great architecture if infrastructure is lacking. There is a dependency.
5. Construct service infrastructure incrementally.

Practice 1 Overview: Check Your Understanding

This practice covers the following topics:

- Program and project scopes
- SOA entry points

ORACLE

Understanding the Program and Project Scopes

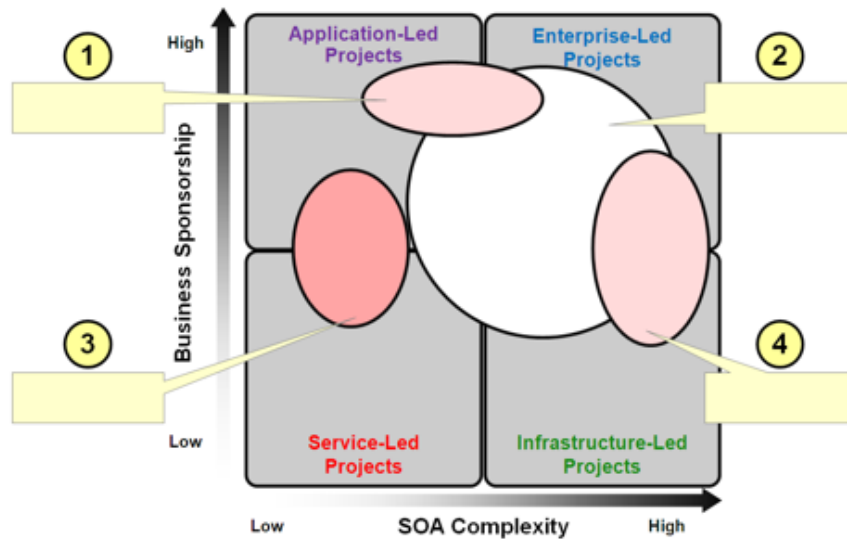
Identify each of the activities on the left as part of either the program or the project scope.

Activity	Program	Project
Describing the scope of SOA adoption in the enterprise		
Execution phase		
Roadmap creation		
Service delivery		
Strategy and planning		
Delivering value to the business		

ORACLE

Identifying SOA Entry Points

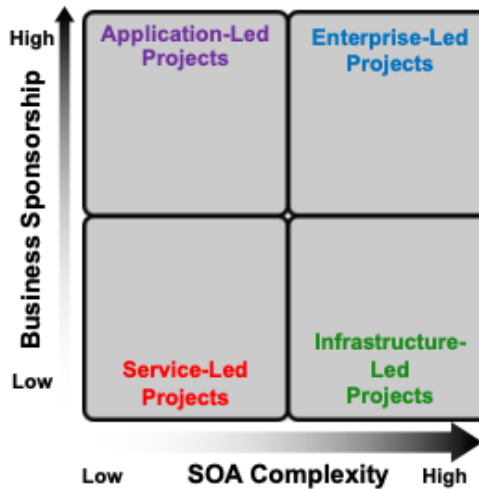
Label each of the numbered SOA entry points as *pragmatic*, *tactical*, or *strategic*.



ORACLE

Case Study: Analyzing SOA Entry Points

In this practice, you read the background of a fictional company and then use that information to identify the company's SOA entry point as *pragmatic*, *tactical*, or *strategic*.



ORACLE