

Service-Oriented Analysis

Written by:



13 Stonebriar Road
Columbia, SC 29212
(803) 781-7628
www.evanetics.com

Review Sample

Student Manual:

Copyright © 2007-2009 Evanetics, Inc. All rights reserved. Companies, names and data used in examples and exercises herein are fictitious unless otherwise noted. No part of these materials may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Evanetics, Inc.

Product and company names mentioned herein are the trademarks or registered trademarks of their respective owners.

This course was produced by:

Gary K. Evans
Evanetics, Inc.
13 Stonebriar Road
Columbia, SC 29212
(voice) 803-781-7628
www.evanetics.com

Table of Contents – 1 of 2

1. **Course Introduction (pg 5)**
 - Who is This Course For?
 - What Will You Learn?
2. **Overview of Service Orientation (13)**
 - *Group Discussion: SO in your Company*
 - SO-Architecture & Web Services
 - Service-Orientation Principles
 - Benefits & Challenges of SO
 - What Really Is New with Service-Orientation?
3. **Overview of SO-Analysis (37)**
 - SO-A and OOA, BPMN and Entity Analysis
 - Top-down, or Bottom-up?
 - Our Process Roadmap
 - *Group Discussion: Your experience with SO and SOA*
4. **Gather Business Drivers & Business Requirements (45)**
 - Introduce Course Case Study
 - Example Case Study: Business Requirements
 - *Exercise Case Study: Business Requirements*
5. **Identify Users, Stakeholders and Their Business Goals (51)**
 - Example Case Study: Identify Users & Goals
 - *Exercise Case Study: Identify Users & Goals*
6. **Identify Candidate Services (61)**
 - Challenges of Identifying Services
 - Business Process Modeling
 - UML Activity Diagrams
 - Use Case Modeling
 - Feature Analysis
 - Example Case Study: Identify Candidate Services
 - *Exercise Case Study: Identify Candidate Services*
7. **Identify Service Granularity (75)**
 - Coarse, Medium & Fine-Grained Services
 - Granularity Maps
 - Example Case Study: Granularity Maps
 - *Exercise Case Study: Granularity Maps*

Review Sample

Table of Contents – 2 of 2

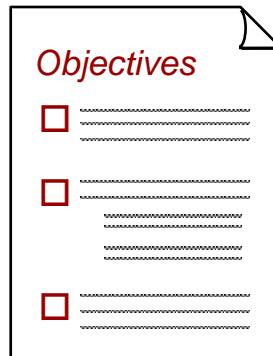
8. **Service Refactoring (85)**
 - The 6 Refactorings
 - Unification, Intersection, Decomposition
 - Subset, Subtraction, Complex
 - Example Case Study: Service Refactorings
 - *Exercise Case Study: Service Refactorings*
9. **Challenge the Candidate Services (101)**
 - The Challenge Questions
 - Building Updated Granularity Maps
 - Example Case Study: Challenging the Services
 - *Exercise Case Study: Challenging the Services*
10. **Identify Service Responsibilities (113)**
 - Responsibility-Driven Analysis
 - Service-Responsibility-Collaborator Specifications
 - Example Case Study: Service Responsibilities
 - *Exercise Case Study: Service Responsibilities*
11. **Identify Service Operations & Interactions (125)**
 - The Service Interaction Diagram (SID)
 - Benefits of the Service Interaction Diagram
 - Example Case Study: Building the SID (#1)
 - Validating the SID against the Service Responsibilities
 - Example Case Study: Building the SID (#2)
 - Updating Service Responsibilities
 - Example Case Study: Service Operations & Interactions
 - *Exercise Case Study: Building the SID, Updating Service Responsibilities*
12. **Wrapup (149)**
13. **References (155)**

Review Sample

2 Overview of Service-Orientation

Review Sample

Section Objectives



Get a firm appreciation for “What are the problems Service-Oriented is trying to solve?”

Counteract some of the “hype” about how radical Service-Oriented principles and Service-Oriented architecture are

Explain the concepts of Service-Oriented, the framework within which SO-Analysis is carried out

Review Sample

Group Discussion

Form groups of 3-4 people and write brief answers to the following questions:



What is your definition of Service-Orientation?

What is the status of Service-Orientation (SO), or Service-Oriented Architecture (SOA) in your organization?

In your view, is your company adopting SO from

- A technical initiative, or
- A business initiative?

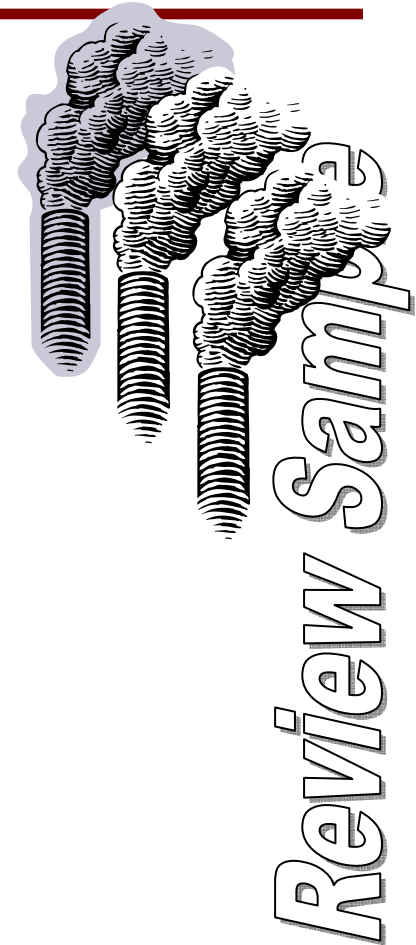
Be prepared to share your answers with the entire class



Estimate: 5 mins.

Review Sample

If Service-Orientation is the Answer...?



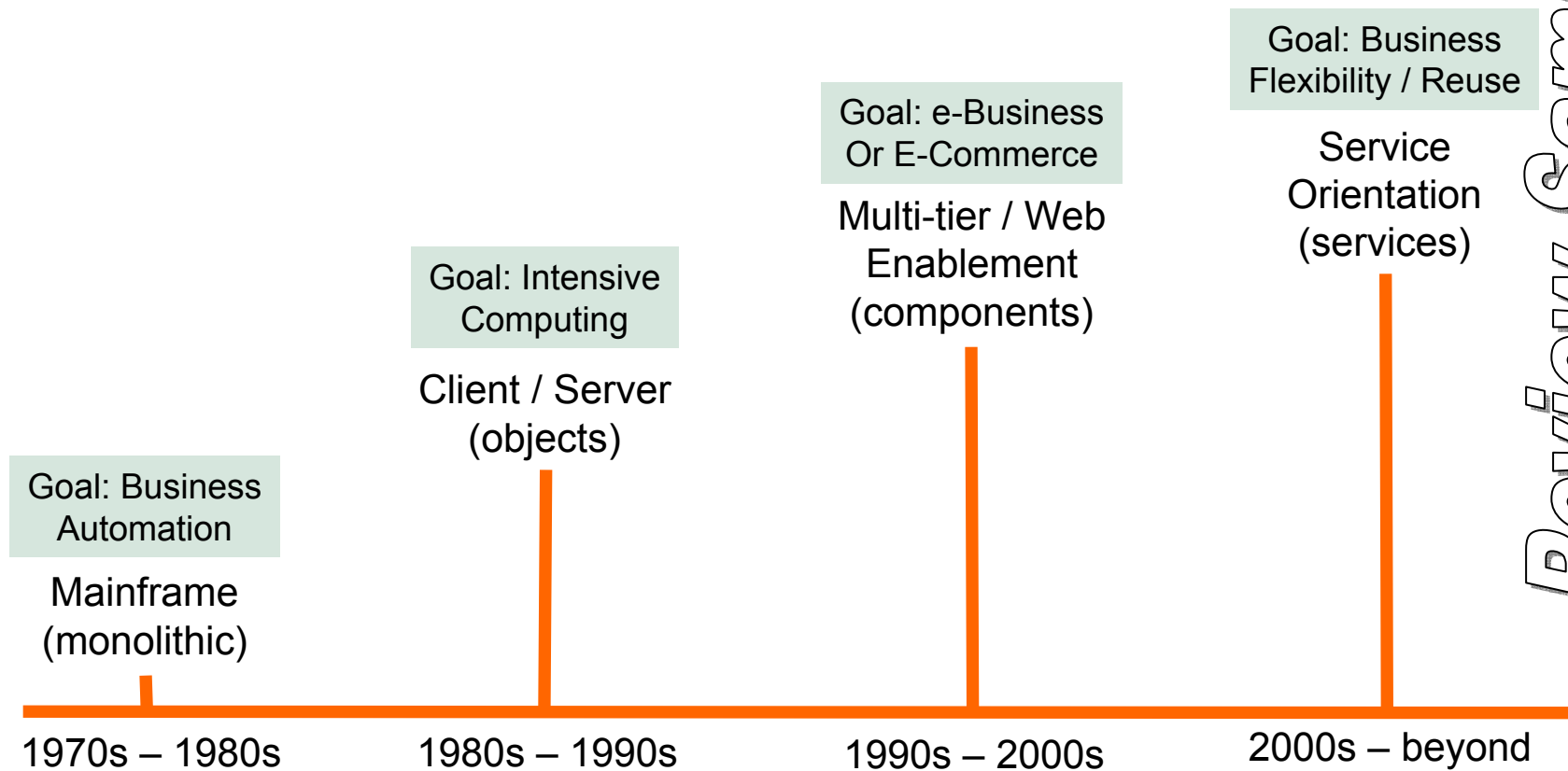
Service-Orientation is an answer to the problems of

- Stove-pipe applications that
 - » Do not share data
 - » Cannot communicate with each other
 - » Duplicate functionality in different ways
 - » Are bound to proprietary interfaces, platforms, protocols, etc.
- Businesses that cannot adapt because of their costly investment in software development and infrastructure
- Overhead of integrating existing business systems with new platforms obtained in mergers & acquisitions
- Inability of management to obtain quickly new reports, or views into their business
- IT budgets that are constantly “backward-facing” to maintain existing systems rather than building strategically for the future

SOA – Is This Really New?

SOA is an evolution and not a revolution, that has evolved

- From client server to brokers and
- Lessons learned during this period



Review Sample

Evolution – Not Revolution

Tool vendors market Service-Orientation as the next “silver bullet”

- This is “market-tecture”

Service-Orientation is the latest evolutionary level of software design

Service-Orientation incorporates industry best practices

- Reveal interface, hide implementation
- Standards and agreement on protocols and interfaces
- Decouple requestor from provider
- Location independence
- Loose coupling to increase flexibility and lower costs of maintenance

Service-Orientation is not a shrink-wrapped solution

- It is a “change of lifestyle” in how we develop systems

Review Sample

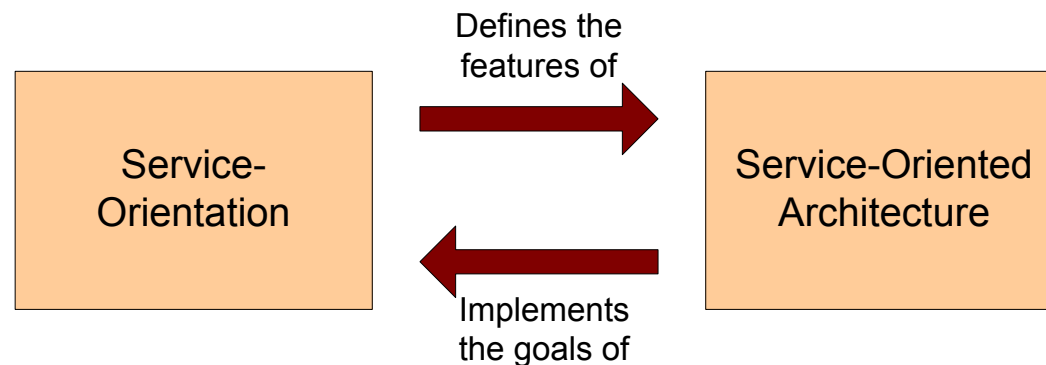


SO and SOA

Service-Oriented Architecture is not the same as Service-Oriented Architecture

– **Service-Oriented Architecture is**

- » A perspective on defining and designing business solutions, irrespective of implementation
- » The unit of modularity is not the class or object, or the procedure, rather it is the *service*



– **Service-Oriented Architecture is**

- » An implementation of business functionality, interfaces and interactions, that can be consumed over a distributed, network environment

– **Web Services is currently the leading approach for implementing a Service-Oriented Architecture**

Review Sample

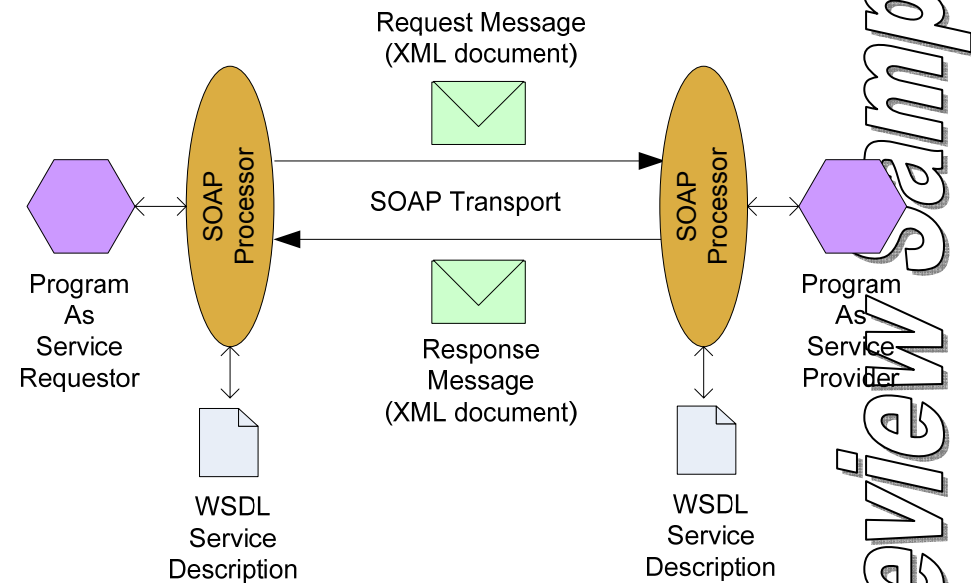
Service-Oriented Architecture & Web Services

Web Services

- Standardized model to support program-to-program communication
- XML data transferred as messages
- Messages transmitted and received rather than remote procedure calls
- SOAP messaging protocol for communication over the Internet

The communicating programs

- Can be written in any language
- Can run on different platforms
- Only requirement is they adhere to the component & communications standards defined for Web Services



WSDL: Web Services Definition Language

XML: eXtensible Markup Language

SOAP: protocol for transmitting data over the network

Service-Oriented Architecture & Web Services

Is Web Services the same as SOA?

- No
- Web services is a technical means of providing program-to-program communication, normally over the Internet (HTTP/TCP-IP)
- Web services defines standardized protocols, components, and messaging interfaces
- You can implement an SOA using Web Services
 - » Web services do not guarantee you are following a Service-Oriented Architecture

You can still build brittle, stovepipe solutions using Web Services

- Just as you can with CORBA, J2EE, COM+ and .NET

WS != SOA

Review Sample

What is Service-Oriented Architecture?

Service-Oriented Architecture (SOA) is

- “...a conceptual business architecture where business functionality, or application logic, is made available to SOA users, or consumers, as shared, reusable services on an IT network. ‘Services’ in an SOA are modules of business or application functionality with exposed interfaces, and are invoked by messages.” [Marks, 1]

The key is being *Business-Service Oriented*



Review Sample

Service-Orientation

Service-Orientation is based on

- The concept of a business *service*, and
- A set of principles to guide our thinking and design of those services

A *service*

- Is the fundamental unit of Service-Orientation
- Services are "reusable modular units of business capabilities, processes, or technical functions that are accessed and delivered in a repeatable fashion to consumers of those services." [Marks, 34]
- Examples:
 - » Reconcile account, Deposit, Withdraw, ...
 - » Add coverage to policy, Renew policy, ...

Service-Orientation is a way of thinking that can be applied to

- Service Oriented Architecture (SOA), and
- Service Oriented Analysis (SO-A)

Review Sample

Approaches to Service-Orientation

Three basic ways to pursue Service-Orientation

Green-field Projects

- New development, not maintenance
- Defined by scope of the project
- Requires a commitment to Service-Orientation principles, Service-Oriented Analysis as well as SOA technology

Legacy Enablement

- Enhancement, or maintenance, project
- Defined by scope of the project
- Requires Service-Oriented Analysis as well as SOA technology

Enterprise SOA

- Defined by infrastructure requirements
- This is the SOA that tool vendors want to sell you
- This is the easy part of Service-Oriented Architecture

Review Sample

Service-Orientation Principles

The principles of Service-Orientation require that services are:

- Business aligned
- Coarse-grained
- Reusable
- Share a formal contract
- Loosely coupled
- Abstract underlying logic
- Composable
- Autonomous
- Stateless
- Discoverable
- Interoperable
- Durable

These may seem tedious,
but they are very important

Review Sample

These affect our analysis goals, so let's look at them in detail

Service-Orientation Principles – 1

Business aligned

- Focus on the business requirements and goals, and then align the technical implementations to those goals
 - » “...service identification and analysis should begin with business imperatives and business requirements, and then cascade into the other services we expect to find in most organizations, such as technical services, data services, infrastructure services, and more.” [Marks, 43]

Coarse-grained

- Start with a top-down analysis of the business needs, not a bottom-up mapping of implementation components.
 - » “...services should represent business functions, processes, or transactions and encapsulate other fine-grained components or services in them.”
 - » “The art and science of services design is finding the right granularity that solves the business problem, can be reused, and can be technically implemented.” [Marks, 39-40]

Composable

- Larger-grain services must be composable from multiple, smaller-grain services. [Erl, 301]
- It must be possible to assemble multiple services into *orchestrated* process flows. [Marks, 42]

Review Sample

Service-Orientation Principles – 2

Reusable

- Reuse of the business logic and business processes (services) are the justification and motivation behind the evolution of SO-A
 - » Defining services to be reusable allows you to accommodate future requirements with less development effort. [Erl, 292]
 - » “You must implement services that have clear and defined reuse across and within business processes and that have multiple consumption patterns in your current and planned business processes.” [Marks, 43]

Share a formal contract

- Services should provide well-defined contracts that separate the functionality of the service from its specific technical implementation. [Marks, 40]
- Service contracts formally specify: service endpoint, each service operation, input and output messages for each operation, rules and characteristics of the service and its operations. [Erl, 295]
 - » In the Web Services world, this contract is specified in the WSDL (Web Services Description Language).

Review Sample

Service-Orientation Principles – 3

Loosely coupled

- Loose coupling exists when a service acquires knowledge of another service while still remaining independent of that service. [Erl, 297]
- Service design should allow specific implementations of services to be replaced, modified, and evolved over time without disrupting the current service consumers and the overall activities of an SOA. [Marks, 41]

Abstract underlying logic

- Remember the software principle: reveal interface, hide implementation.
- Services should strive to be *black boxes*, to hide their details from the outside world.
- It is the service's operations that collectively abstract the underlying service logic.
 - » Services are simply containers for these operations. [Erl, 298-299]

Review Sample

Service-Orientation Principles – 4

Autonomous

- Services should be mature, and able to monitor and direct themselves.
- At execution, the service has exclusive control over the logic it represents.
 - » A service must execute self-governance in its processing. [Erl, 304]
- A service should have well-defined boundary, outside which it has no dependencies.

Stateless

- Stateful entities are entangled with their own past.
 - » Services (i.e. their operations) should be designed to be stateless, which supports re-entrant reuse.
- Service functionality should depend minimally, or not at all, on the state of execution. [Erl, 307]

Review Sample

Service-Orientation Principles – 5

Discoverable

- A service cannot be used, or reused, if it cannot be found.
- All services must be discoverable by potential consumers.
- Discoverable services mitigate against creation of redundant services that do what another services already does. [Erl, 309]
- Service contracts must be published for consumers [Marks, 41]
 - » WSDL, UDDI are technical means of service discovery

Interoperable

- Services must interact — i.e., *interoperate*.
- Interoperability will fail if policies, standards and other design constraints are not enforced. [Marks, 43]

Durable

- A durable service is one that supports, or is part of, a lasting business or process theme.
 - » For example, a bank account reconciliation process is a lasting process.
- Your services must support your long-lasting processes. [Marks, 42]

Review Sample

Benefits of Service-Orientation

Improved integration with intrinsic interoperability

- Opportunity to implement and enforce highly standardized service descriptions and message structures

Inherent reuse

- Application-specific services are inherently reusable because of common, shared interface requirements

Streamlined architectures & solutions

- Service composability means only technologies actually required become part of solution

Leverages legacy investment

- Large adapter/connector after-market exists to link to existing vendor solutions as services, not APIs

Standardized data in XML

- Reduces complexity of data processing and data interfaces between applications

Common communications framework

- Infrastructure is reduced

Best of Breed alternatives

- Vendor neutral

Organizational agility

- Supports evolution and change on common infrastructure and loose coupling [Erl, 60ff]

Review Sample

Challenges of Service-Orientation

It is still possible to build bad SOAs by:

- Building SOAs like tradition distributed architectures
- Not standardizing SOA
- Not creating a transition plan
- Not starting with an XML foundation architecture
- Not understanding SOA performance requirements
- Not understanding security issues
- Not following vendor/platform development & evolution [Erl, 64ff]

Focusing on the technical issues:

- “Some IT organizations spend too much time building reusable technical services (e.g., logging, auditing services) rather than delivering line of business services that contribute to the organization’s business goals.” [Newcomer, 63]

You compromise your success if your move to SO is only an IT-initiative without a concurrent commitment through a business-initiative.

Review Sample

Challenges of Service-Orientation

It requires commitment

- "SOA is difficult to implement, manage and control. ...Not because of the technology...but due to the organizational, cultural, and behavioral aspects of SOA."
- "SOA requires governance to achieve and manage."

It requires a long-term perspective

- "SOA is spatially and temporally distributed. ...It is not implemented all at once. Rather, it is achieved through many discrete projects across both space and time." This makes governance critical, to assure project conformance over time to SOA principles.

It requires facing entirely new challenges and uncertainties

- "SOA ...requires new approaches to identifying, modeling and implementing reusable, interoperable services."
- "SOA is an iterative business approach. There is no single correct path to achieving SOA."

Review Sample

[Marks, 29ff]

Challenges of Service-Orientation

Services are the foundation

Identifying and modeling services is challenging

How do we know we have the "right" services?

Challenge is to accomplish the *SOA Three Rights*:

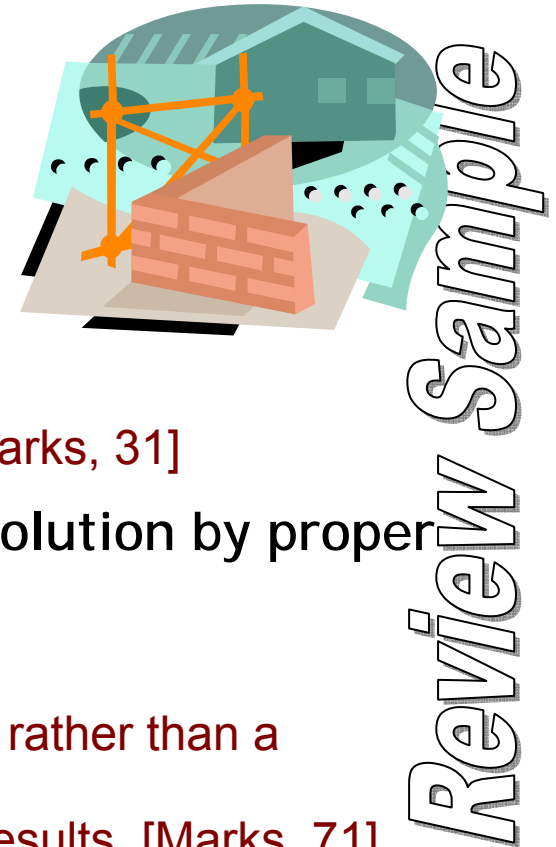
- Identify the "right" services;
- Build those services the "right" way; and
- Run them on the "right" enabling technology stack. [Marks, 31]

You can avoid reducing your SOA to a technology solution by proper SO Analysis

This will:

- Assure your SOA initiative is truly a business initiative rather than a technology initiative.
- Assure that the SOA initiative targets clear business results. [Marks, 71]

And SO Analysis – identifying the "right" services – is the focus of this course



Our Main Concepts for Service-Orientation

Service

- Business-meaningful functionality discoverable and consumable by programs needing that functionality

Operation

- Named functionality within a service. A service exposes one or more operations that can be invoked by a requestor. A service's public operations constitute its public interface.

Request

- A request is a message sent to a service by a requestor program

Response

- A response is a message returned by a service to a requestor program

Orchestration & Choreography

- *Orchestration* refers to assembling services for business processes
- *Choreography* refers to assembling services for peer-to-peer business collaborations.

Review Sample

What Really Is New with Service-Orientation?

For the Business Analyst

- Your role is critical to Service-Orientation
- Your skill in identifying the “right” services will enable, or will constrain, your organization’s adaptability

For the Developer

- IT must become adept at separating business services from technical services
- IT must relinquish strategy to the business community, and hold the business accountable for clarity in the services they specify
- IT must accept that there is much more to Service-Orientation than putting Visual Studio on every developer’s computer

For the Enterprise

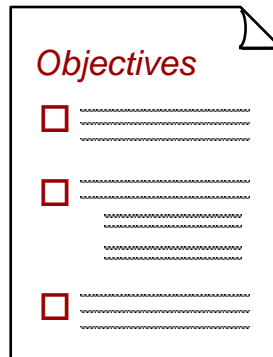
- This is the end of life as you know it now — a new beginning
- Service-Orientation is a “lifestyle change”
 - » It is not an event
 - » It is a goal to be achieved over time, and many projects

Review Sample

3 Overview of SO-Analysis

Review Sample

Section Objectives



Provide a high-level introduction to the goals & activities of Service-Oriented Analysis

Describe how Service-Oriented Analysis is similar to other analysis techniques

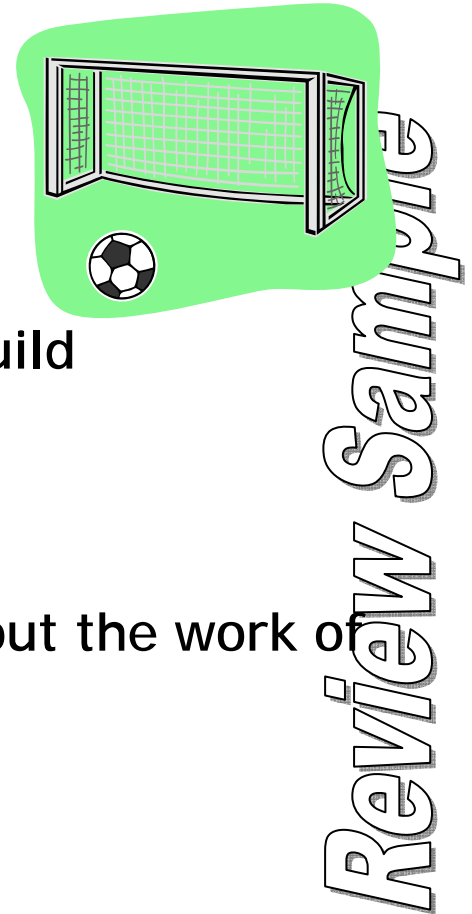
Provide a process context for the remainder of the course

Review Sample

Service-Oriented Analysis

We have four goals in Service-Oriented Analysis

1. Identify, then describe, the services we need to build
2. Determine the logic in each service
3. Identify the interfaces (operations) in each service
4. Identify how these services may interact to carry out the work of the business



Service-Oriented Analysis

Service-Oriented Analysis is similar to other analysis techniques you may already know

Object-Oriented Analysis

- Analysis-level classes (*domain classes*) contain operations (methods) that represent the intent of that class's role in our business domain
- Contrast: SO-analysis strive for coarse-grained service units, while OO analysis strives for fine-grained units of functionality [Erl, 107]

Business Process Modeling

- Business processes are carried out by multiple services and operations in those services
- Every business model is unique, and business process modeling is required to identify services that best fit the organization

Entity (Data) Analysis

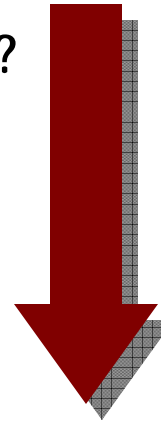
- Modeling business entities, documents, and transactions offers us the major concepts in a business organization
- These concepts are qualified with business rules and policies we will implement in services

Review Sample

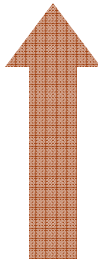
Service-Oriented Analysis

When we look for services, should we think...

Top-down?



Or Bottom-up?



Yes!

You will need both approaches (think “outside, in”)

Note: In your initial analysis, Top-down is the best approach

- Minimizes your risk of being misled by legacy technical or implementation choices

Use a Bottom-up approach when you are ready to implement physical services through Service Oriented Design

Review Sample

SO-Analysis Activities – High-Level

At a high-level perspective, Service-Oriented Analysis requires these activities:

- Identify your business drivers and requirements
 - » What is your business trying to achieve?
- Identify the business values in these requirements
 - » Prioritize and justify
- Identify re-engineering opportunities
 - » Consider the future state desired for your organization
- Identify the services you need to fulfill these requirements
 - » Follow the Service-Orientation Principles in the previous section
- Describe how the services might interact
- Model and characterize these services
 - » Name of the service
 - » Responsibilities supported by the service
 - » Business logic (operations) provided by the service
 - » Interfaces it presents to service's consumers

Review Sample

Our Process Roadmap

In this course we will follow this process in identifying and defining the services in our example case study:

1. Gather business drivers and requirements
2. Identify our users and their business goals
3. Identify candidate services
4. Challenge the candidate services to obtain the “right” services
5. Identify responsibilities of each service
6. Identify operations that support the service’s responsibilities
7. Identify how our services might interact



Review Sample

Group Discussion

Form groups of 3-4 people and discuss the following questions



For those whose organization is actively involved in SO or SOA

- What is driving your organization's move to embrace either SO or SOA?
- What benefits is your organization expecting?
- Are the benefits being realized?
- What obstacles have you experienced, or are you anticipating?
- How has your job role been affected by moving to a Service Orientation?

Be prepared to share your answers with the entire class



Estimate: 5 mins.

Review Sample