* **Describe Requirement of SOA in business application? Explain entire implement SOA(6 Marks)**

**Ans :**

What SOA services should I build?" Different IT groups within your organization may be saying different things: Some may want to build technical services, such as a content management service, security (authentication/authorization) related services, or something else. However, the key to an SOA project is your first set of business services

## Service identification

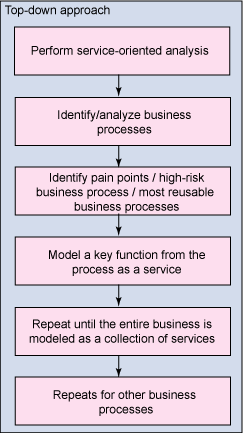
Let's touch on some basic ideas of how to identify the first service or the first few services you're going to build as part of your SOA. These services need to be isolated in terms of impact to the business, and they must be fully scoped out. On the flip side, they should be significant enough to demonstrate the value and vision behind your long-term SOA roadmap.

### Top-down service identification

In a top-down approach, you begin with high-level business use cases or business-process flows that exist within your organization. You can also start with a business strategy or an IT strategic plan presentation (which would include the business strategy). This is just a starting point for you to begin breaking down the processes into functional areas or subsystems. You lay these out for your entire organization and then start to identify any pain points, highly reusable use cases, or functions you can short-list for your candidate SOA services. Remember not to choose the most complex or controversial service.

The top-down approach is business driven: Business documentation exists that can serve as references to identify your SOA services. [Figure 1](http://www.ibm.com/developerworks/library/ar-soareq2/#fig1) shows a simple set of steps that you could use for the top-down approach.

##### Figure 1. Top-down approach

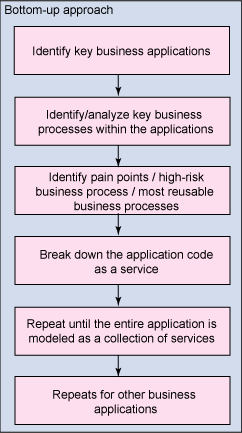


### Bottom-up service identification

In a bottom-up approach, you begin with existing systems or applications. You try to dig out any documentation that exists for these systems and then build your functional areas, subsystem maps, and higher-level business use cases from there. This may be more difficult, but in most organizations it's unavoidable because the high-level documents used in the top-down approach either don't exist or aren't up to date. The bottom-up approach appears to be more IT driven; the business lacks documentation of its strategy, functions, and core competencies.

[Figure 2](http://www.ibm.com/developerworks/library/ar-soareq2/#fig2) shows a sample bottom-up approach for service identification. It's not that much different from the top-down approach -- but the starting point is very different.

##### Figure 2. Bottom-up approach



## Gather the requirements for a single service

This section drills into one service. It discusses the types of requirements and the process you need to follow to gather the requirements for a service in your SOA.

### Types of requirements

You're ready to capture the requirements for your first SOA service. In a broad sense, you need to capture requirements in the following categories. Remember, this article focuses on the business requirements -- the previous article talked about the technical requirements for the service(s):

* **Accessibility**. How does the user of the service find and access this service? This requirement borders being a technical requirement and a business requirement. For now, think of the business process that needs to find and invoke the service you're building.
* **Functionality**. What core business process or function will this service provide? What business problem are you solving? This discussion can become very long. You must determine the appropriate granularity of what becomes a service in your SOA. (See the [Resources](http://www.ibm.com/developerworks/library/ar-soareq2/#resources) section at the end of this article for articles that discuss this in more detail.)
* **Interaction**. How does the service or application that calls this service interact with the service? How does the service handle error conditions?
* **Information**. What data is sent to this service and back from this service?
* **Process**. What are the relationships between the actions and events of this service?

**Web service properties for SOA**

->**Logical view:** The service is an abstracted, logical view of actual programs, databases, business processes, etc. defined in terms of what it does, typically carrying out a business-level operation.

->**Message orientation:** The service is formally defined in terms of the messages exchanged between provider agents and requester agents.

*->***Description orientation:** The description supports the public nature of the SOA: only those details that are exposed to the public and important for the use of the service should be included in the description. The semantics of a service should be documented, either directly or indirectly, by its description.

->**Granularity:** Services tend to use a small number of operations with relatively large and complex messages.

->**Network orientation:** Services tend to be oriented toward use over a network, though this is not an absolute

requirement.

->**Platform neutral:** Messages are sent in a platform-neutral, standardized format delivered through the interfaces. XML is the most obvious format that meets this constraint.