Service Oriented Analysis

UNIT-V

Service Oriented Analysis

- The process of determining how business automation requirements can be represented through service orientation

Objectives of Service Oriented Analysis

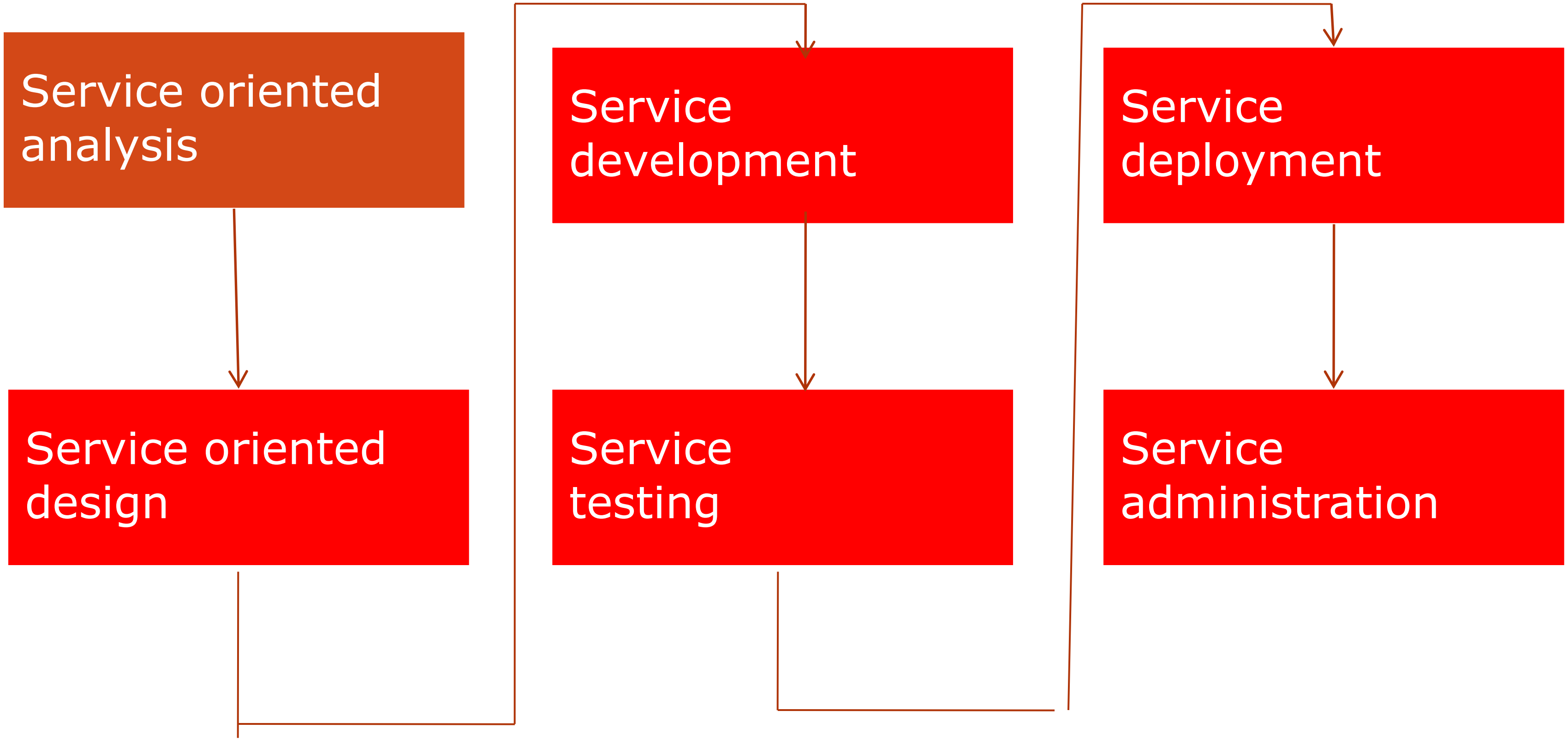
- Primary questions to be addressed during this Phase:

- What services need to be built?
- What logic should be encapsulated by each service?

Goals of performing a service-oriented analysis

- Define a preliminary set of service operation candidates
- Group service operation candidates into logical contexts - represent service candidates
- Define preliminary service boundaries – to avoid overlap with any existing or planned services.
- Identify reusable encapsulated logic
- Ensure the appropriateness of encapsulated logic
- Define any known preliminary composition models.

A high level Service-Oriented Analysis Process



Service oriented analysis

Service oriented design

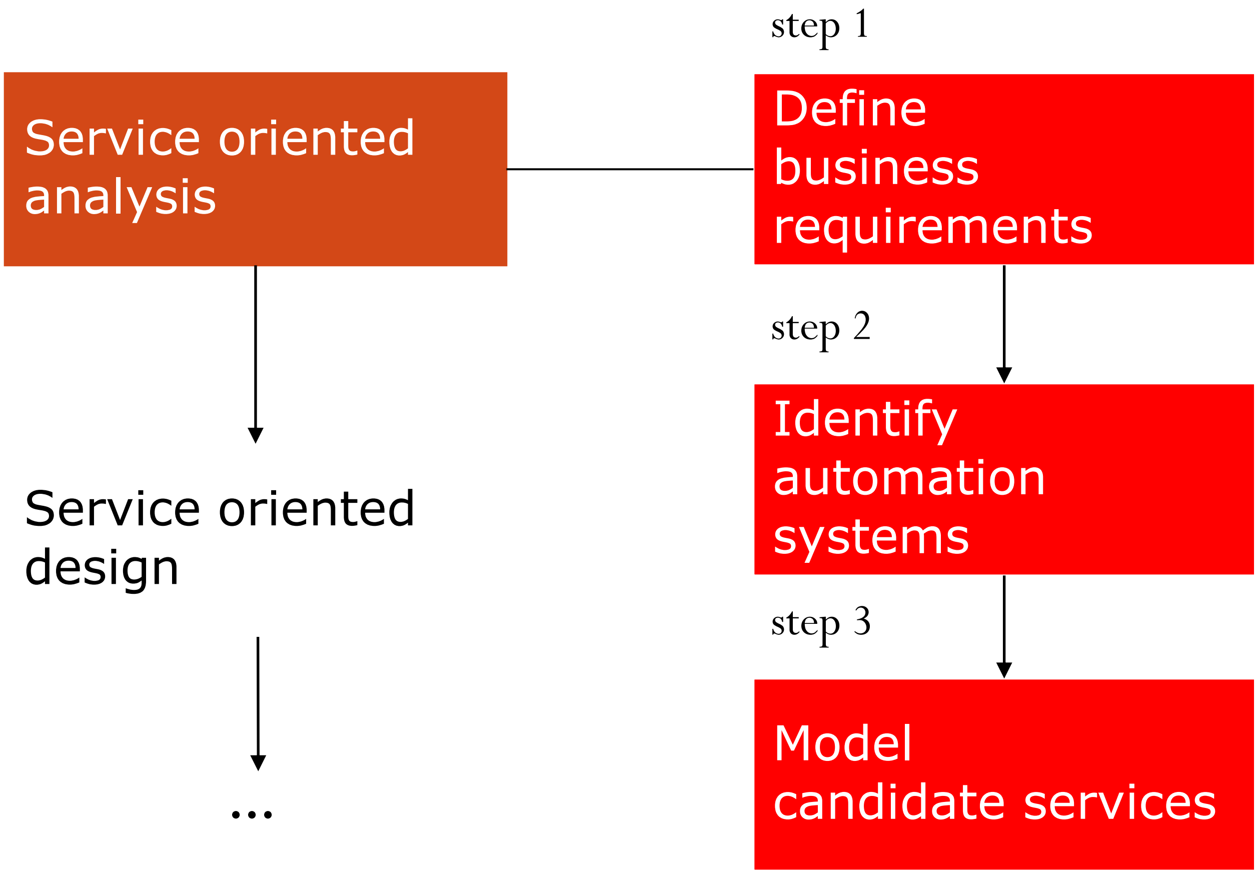
Service development

Service testing

Service deployment

Service administration

Service Oriented Analysis – Sub Steps



Service oriented analysis

Service oriented design

…

step 1

Define business requirements

step 2

Identify automation systems

step 3

Model candidate services

Step1: Define Business automation requirements

- Business requirements collected
- Its documentation required for analysis process to begin
- Scope - creation of services in support of a service-oriented solution
- Business requirements sufficiently mature to define high-level automation process

Step2: Identify existing automation systems

- Identify existing application logic already available for requirements identified in Step 1

- It won’t bother existing logic can encapsulate web service or replace legacy application logic – just to scope potential systems affected

- This is needed to identify application service candidates during the service modeling process described in Step 3.

Step3: Model Candidate Services

- *Service Modeling - a process by which service operation candidates are identified*
- Group into a logical context
- Eventually take shape as service candidates
- Further assembled into a tentative composite model
- Composite model represents combined logic of service-oriented application

Service Modeling

- To organize the information gathered in Step1 and Step2 of Service-Oriented Analysis process

- Structure such that it fits your organizational platforms and procedures

Service Candidates and Service Operation Candidates

- Service Candidate / Service Operation Candidates - abstract services or abstract service operation may or may not be realized as part of the eventual concrete design

- Abstract Services or Abstract Service Operations are end result of the process, Service Modeling

- **Service Modeling Process, Steps Contd……**

Step 1: Decompose the business process

- Documented business process is break down into a series of granular process steps
- i.e. Process's workflow logic is decomposed into most granular representation of processing steps

Step 2: Identify business service operation candidates

- Identify steps and filter out parts not belonging to the potential logic or cannot be service operation candidate

- Examples :
- Manual process steps cannot be automated
- Process steps of existing legacy logic

Step 3: Abstract orchestration logic

- If orchestration layer is part of your SOA, identify parts of processing logic that this layer would potentially abstract

- Potential types of logic suitable for this layer include:
- business rules
- conditional logic
- exception logic
- sequence logic

Step 4: Create business service candidates

- Review the processing steps when grouped that forms one or more logical contexts
- Each context represents a service candidate
- Context depends on types of business service chosen

- Example:
- Task-centric business services require a context specific to the process
- Entity-centric business services need to group processing steps according to their relation to previously defined entities

Step 5: Refine and apply principles of service-orientation

- Apply key service-orientation principles to the identified service operation candidate

- This step ensures that each service operation candidate identified is potentially reusable and as autonomous as possible

Step 6: Identify candidate service compositions

- Identify a set of the most common scenarios that can take place within the boundaries of the business process

- Ensure that as part of chosen scenarios includes failure conditions that involve exception handling logic

Step 7: Revise business service operation grouping

- Based on Step 6, revisit the grouping of your business process steps and revise the organization of service operation candidates if necessary

- Consolidate or create new groups (service candidates) at this point

Step 8: Analyze application processing requirements

- This series of steps are optional and suited for complex business processes and larger service-oriented environments

- Study and abstract the technology-centric service candidates to form a preliminary application services layer

Step 9: Identify application service operation candidates

- Break down each application logic processing requirement into a series of steps
- Label these steps so that they reference the function they are performing

Step 10: Create application service candidates

- Group these processing steps according to a predefined context
- The primary context is a logical relationship between operation candidates

- This relationship can be based on any number of factors, including:
- association with a specific legacy system
- association with one or more solution components
- logical grouping according to type of function

Step 11: Revise candidate service compositions

- Revisit the original scenarios you identified in Step 5
- Incorporate the new application service candidates as well
- Keep track of how business service candidates map to underlying application service candidates

Step 12: Revise application service operation grouping

- Step 11 results in changes to the grouping and definition of application service operation candidates
- It notifies omissions in application-level processing steps, resulting addition of new service operation candidates or new service candidates

Service Modeling Guidelines (1)

- Take into account potential cross-process reusability of logic being encapsulated
- Consider potential intra-process reusability of logic being encapsulated
- Factor in process-related dependencies
- Model for cross-application reuse

Service Modeling Guidelines (2)

- Speculate on further decomposition requirements
- Identify logical units of work with explicit boundaries
- Prevent logic boundary creep
- Emulate process services when not using orchestration

Service Modeling Guidelines (3)

- Target a balanced model
- Classify service modeling logic
- Allocate appropriate modeling resources
- Create and publish business service modeling standards