

SOA

Arquitectura Orientada a Servicios

II

Profesor: Alejandro Sirvent Llamas

Curso: 2017-2018



Análisis



- Análisis
 - Qué servicios necesito crear?
 - Qué lógica debería encapsular cada servicio?
 - Objetivos
 - Definir un **conjunto previo** de operaciones de **servicios candidatas**
 - Agrupar operaciones en un contexto lógico → **servicios candidatos**
 - Definir límites de los servicios para evitar solapamientos.
 - **Identificar** la **reusabilidad** potencial de la lógica encapsulada en los servicios.
 - Asegurar que el contexto de encapsulamiento de la lógica es apropiado para su objetivo previsto.
 - Definir un **modelo preliminar de composición**.

- "Services" vs "Service Candidates"
 - El **objetivo principal** de la etapa de análisis orientada a servicios es **averiguar** qué es lo **que necesitamos** para **diseñar** y construir más tarde en las fases posteriores del proyecto.
 - **No** estamos en la **etapa** de implementación de un **diseño**.
 - Estamos realizando un **análisis** cuyos resultados, mostraremos como una propuesta de separación de la lógica utilizada como entrada, para su consideración durante la fase service-oriented design.
 - En otras palabras , **estamos produciendo candidatos abstractos que pueden o no ser realizados como parte del diseño de eventual** .

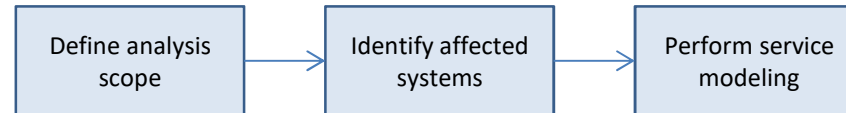
- "Services" vs "Service Candidates"
 - La razón de esta **distinción** es tan **importante** porque , una vez que nuestros **candidatos** se **presentan** al **proceso** de **diseño**, se ven sometidos a las **realidades** de la arquitectura técnica en la que estos, se espera que vivan .
 - Una vez que las **limitaciones** , **requisitos** y limitaciones específicas a los entornos de ejecución se **tengan** en **cuenta**, el **diseño final** de un **servicio** estará determinado por la **elección** del **candidato** original correspondiente .

- "Services" vs "Service Candidates"
 - Por lo tanto, **en esta etapa, no producimos servicios:**
 - Creamos servicios candidatos.
 - Del mismo modo , no definimos operaciones de servicio , se propone candidatos de operación de servicio .
 - Por último, los candidatos de servicios y los candidatos de operación de servicio son el resultado final de un proceso llamado modelado de servicios .



SOA II

- ## Análisis



- Fases → aplicadas a cada proceso de negocio

- ### Recolección de información

- Definición del ámbito del análisis
 - » **Identificar** procesos o partes de los mismos ya analizados en **iteraciones anteriores**
 - » Para estos → saltarse el modelado
 - Identificación de sistemas afectados
 - » **Detectar** que **sistemas** de la empresa se verán **afectados** por el análisis
 - » **Sistemas heredados**
 - » **Restricciones** que pueden determinar la **granularidad** de los servicios candidatos

- ### Modelado de servicios

- Definición de servicios y capacidades desde el punto de vista conceptual (**candidatos**)
 - **Identificación de composición** de servicios candidatos (agrupación de capacidades candidatas)

Modelado

Modelado de servicios

La naturaleza iterativa del análisis del inventario , permite a los candidatos de servicio ser revisados y refinados, repetidamente, antes de la creación de los servicios correspondientes.

Una definición de proceso de negocio se descompone:

Paso 1: En su **representación más detallada**, lo que resulta en una serie de acciones granulares.

Paso 2: Las adecuadas para la encapsulación de servicios se convierten en **candidatos potenciales de capacidad de servicio**.

Step 1

Decompose
business
processes

Step 2

Identify
capacity
candidates

Step 3

Identify
agnostic
service
candidate

Step 4

Identify
process specific
logic

Step 5

Apply service
orientation

Step 6

Identify
candidate
service
compositions

Step 7

Analyze
processing
requirements

Step 8

Identify utility
service
capability
candidates

Step 9

Define utility
service
candidates

Step 10

Apply service
orientation

Step 11

Revise
candidate
service
composition

Step 12

Revise
capability
candidate
grouping

Modelado de servicios

La lógica de servicio de cada capability candidate se **evalúa** en términos de si es **específico o agnóstico al proceso de negocio actual**.

Los capability candidate Agnostic se agrupan en los servicios candidatos agnósticos generalmente sobre la base de la entidad y los modelos de servicios públicos (**Paso 3**), mientras que los capability candidate no-agnósticos se colocan en una *task-centric business services* (**paso 4**).

Análisis

Step 1

Decompose
business
processes

Step 2

Identify
capacity
candidates

Step 3

Identify
agnostic
service
candidate

Step 4

Identify
process specific
logic

Step 5

Apply service
orientation

Step 6

Identify
candidate
service
compositions

Step 7

Analyze
processing
requirements

Step 8

Identify utility
service
capability
candidates

Step 9

Define utility
service
candidates

Step 10

Apply service
orientation

Step 11

Revise
candidate
service
composition

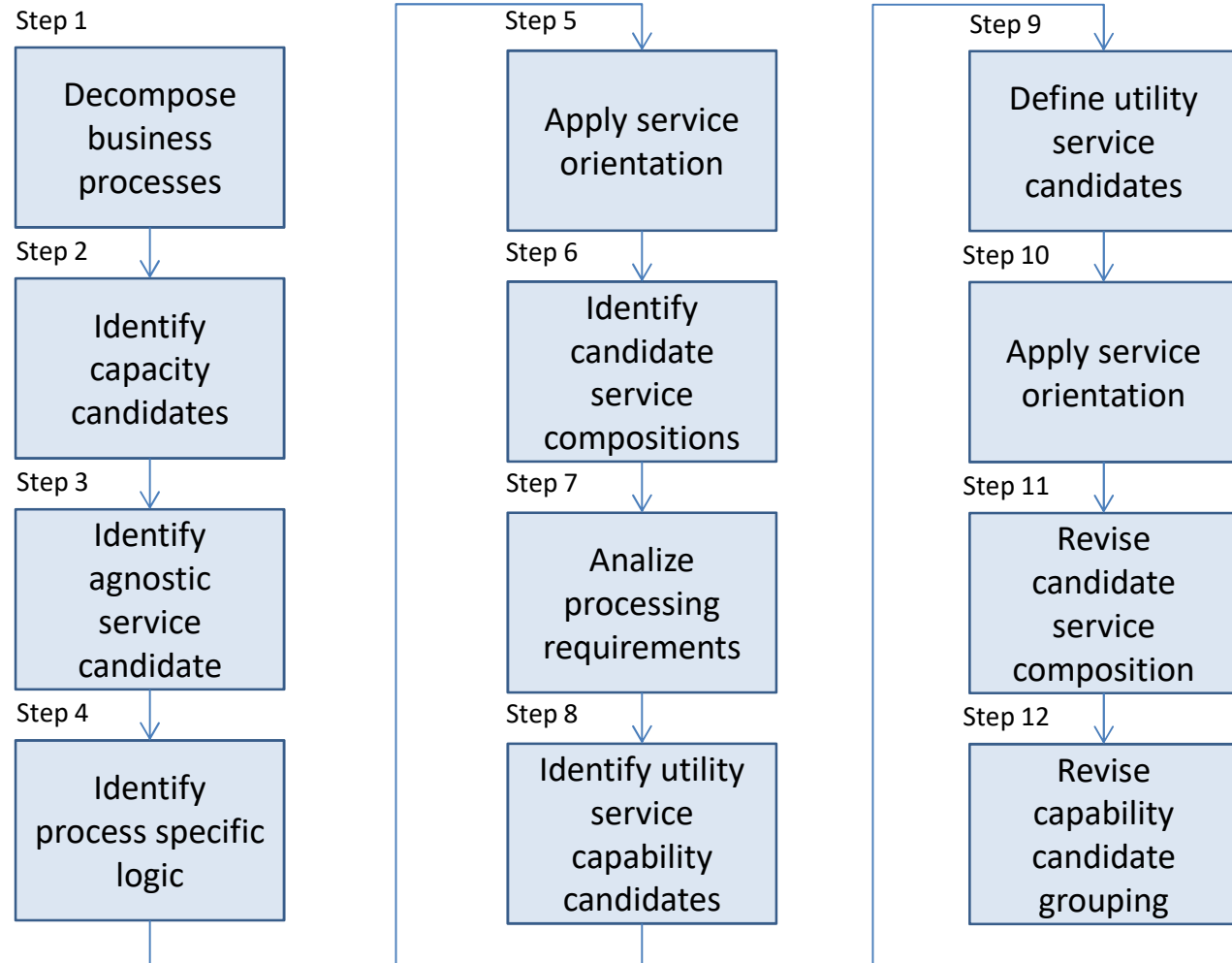
Step 12

Revise
capability
candidate
grouping

Modelado de servicios

Aplicamos los principios de orientación a servicios para dar forma a los servicios capability candidates modelados (Paso 5).

Análisis

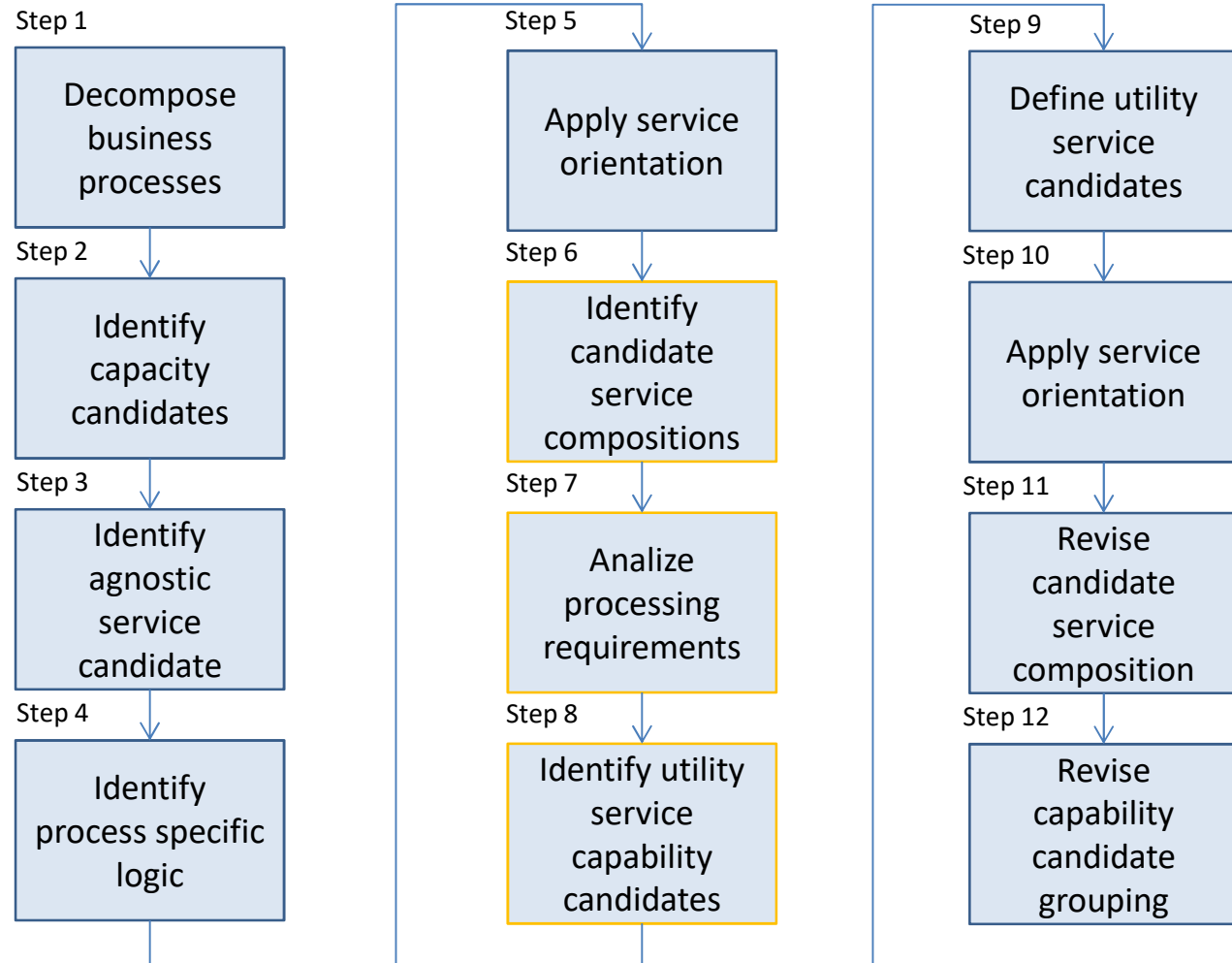


Modelado de servicios

Los siguientes tres principios de orientación a servicios se aplican normalmente durante el proceso de modelado de servicios:

- Reutilización de servicio
- Servicio de Autonomía
- Capacidad de descubrimiento de servicios

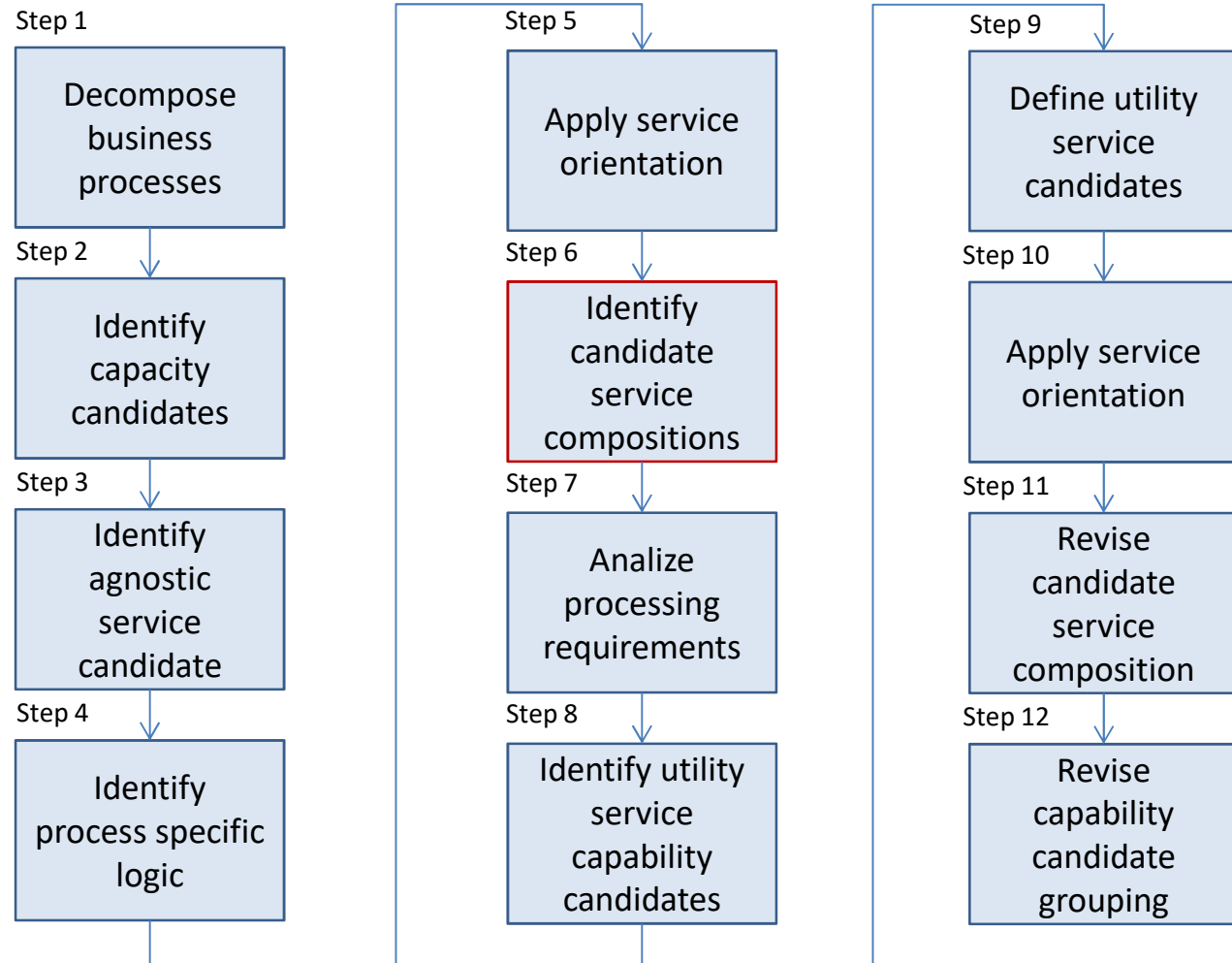
Análisis



Modelado de servicios

Después de establecer el conjunto inicial de candidatos de servicio, una composición de candidatos se **ensambla** y se **somete a posibles escenarios** en tiempo de ejecución (paso 6).

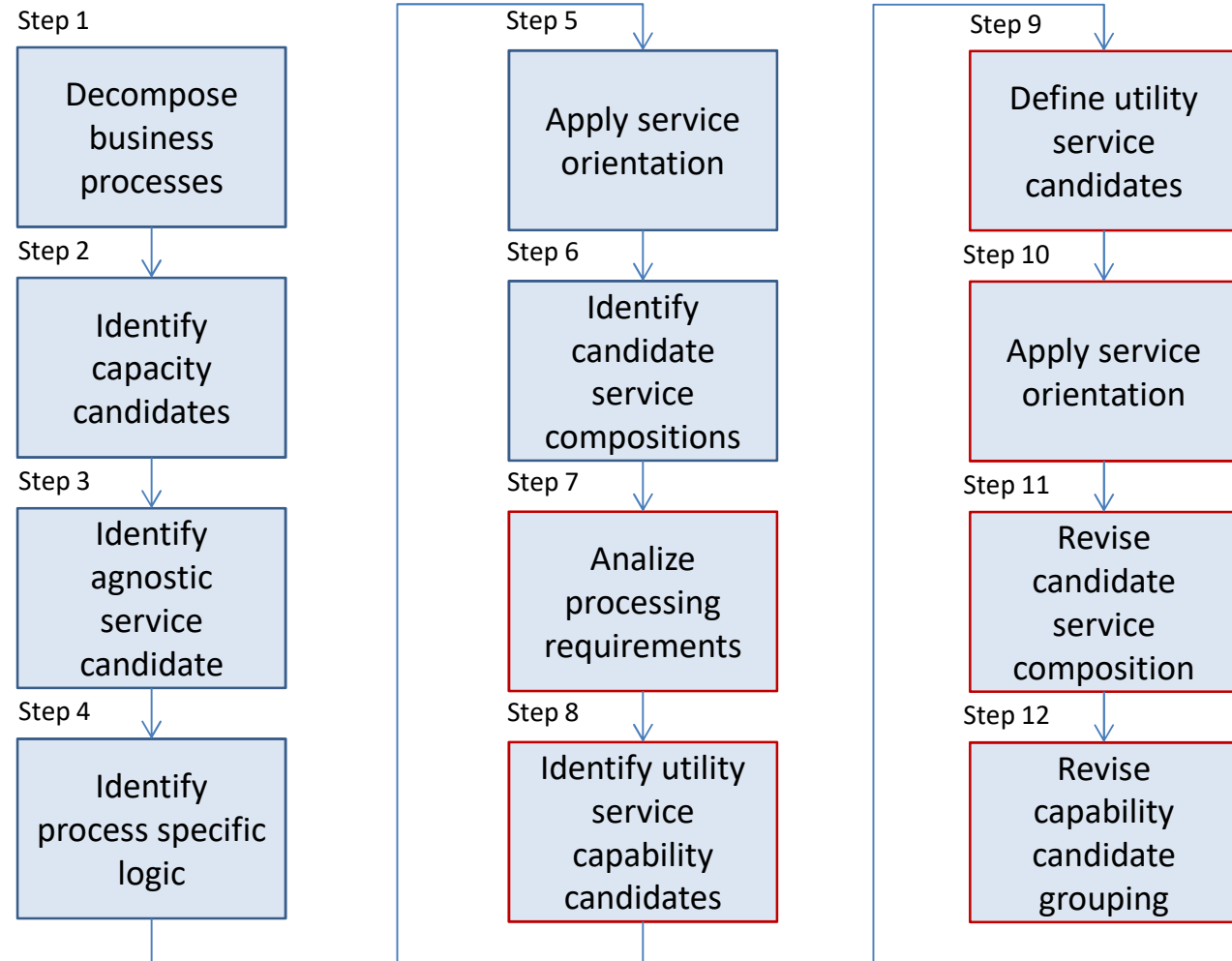
Análisis



Modelado de servicios

Posteriormente, **cada uno** de los **servicios capability candidates identificados**, son **estudiados más** para explorar los requisitos de procesamiento adicionales que puedan ser necesarios para llevar a cabo sus funciones. Esto da inicio a la segunda parte del proceso de modelado de servicios (**pasos 7-12**) durante el cual **los candidatos de capacidad de servicio de utilidad adicionales se definen** en general.

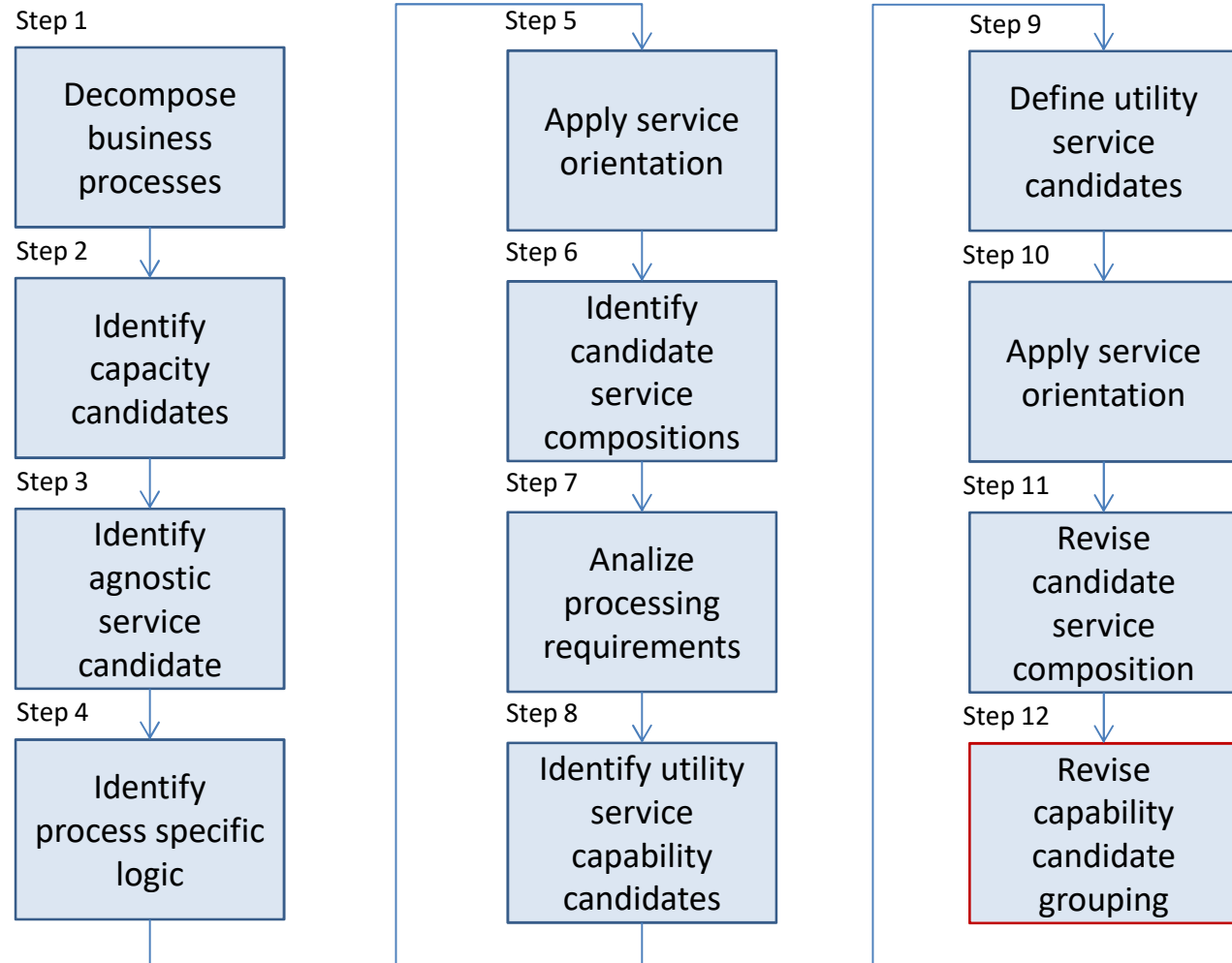
Análisis



Modelado de servicios

El proceso finaliza con una etapa de modelado de candidatos extendida y una revisión final de todas las definiciones de capability candidates y de servicios creados hasta ahora.

Análisis



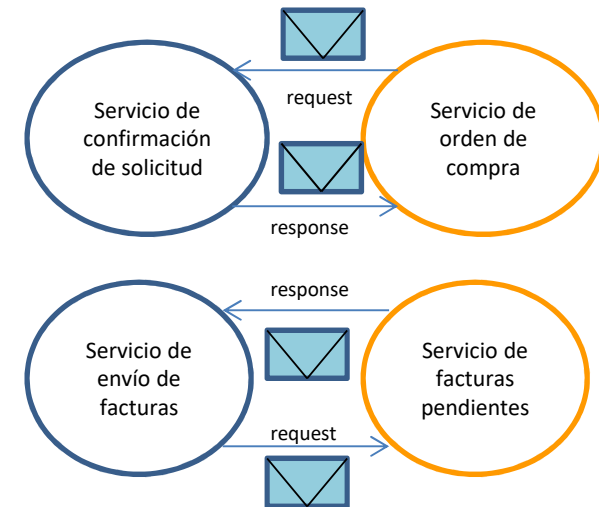
- Caso de estudio

- RailCo.

- Proveedor de piezas para ferrocarriles especializado en
 - Frenos de aire
 - Herramientas para instalación
 - Servicio de técnicos especializados para instalaciones y reparaciones
 - Tecnologías
 - Aplicación cliente-servidor de dos niveles para la gestión de cuentas y control de transacciones de inventario
 - » Dos administrativos introducen los datos desde documentos estándar
 - » Órdenes de compras de entrada y salida y facturas
 - » La recepción y envío de estos documentos
 - Inicia la recepción del inventario correspondiente
 - Inicia proceso de pedido
 - CRM para la gestión de información de clientes y partners
 - » Base de datos + Front-end Web
 - » Interfaz de usuario para ´visualización de informes

- Problemas
 - Procesos muy pesados → clientes a la competencia
 - » Más eficiente y menos coste
 - Competencia → soluciones B2B de los sistemas de gestión (contabilidad) con los clientes
 - » Reduce costes
 - » Reducir tiempo de respuesta
 - » Principal cliente TLS ha empezado una relación de negocio online
- Objetivos
 - Mejorar sus sistemas tecnológicos para automatizar los procesos
 - Relación online con TLS
 - Nuevos clientes que compensen las perdidas con TLS

- Caso de estudio
- Análisis
 - Definir ámbito del análisis
 - Centrado en SOA
 - Objetivos
 - Resolver problemas actuales de automatización
 - » Tareas administrativas manuales
 - Incrementar clientes vía B2B
- Identificar sistemas existentes
 - Solución Bottom-up
 - Desarrollo de servicios B2B para comunicarse con TLS.
 - Centrados en la interacción tecnológica.
 - Solución SOA 100%
 - Solución SOA B2B de facturación independiente del partner.
 - Solución SOA de integración con sistemas heredados independientes de la lógica subyacente.
- Modelado de servicios candidatos

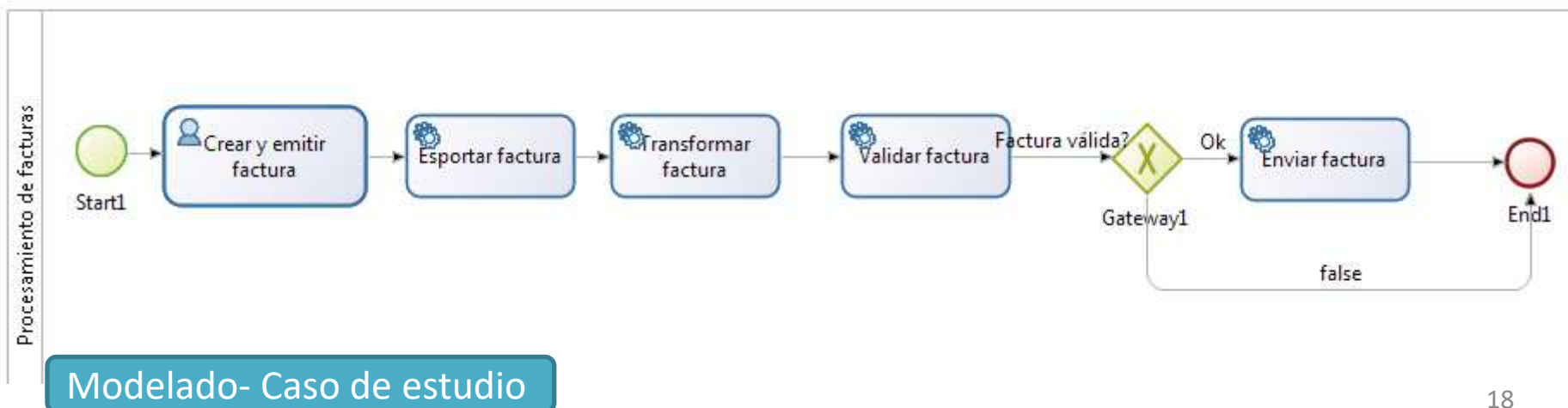


- Caso de estudio

- Modelado del Servicio

- Descomposición del proceso de negocio “Invoice Submission Process” (Step 1)

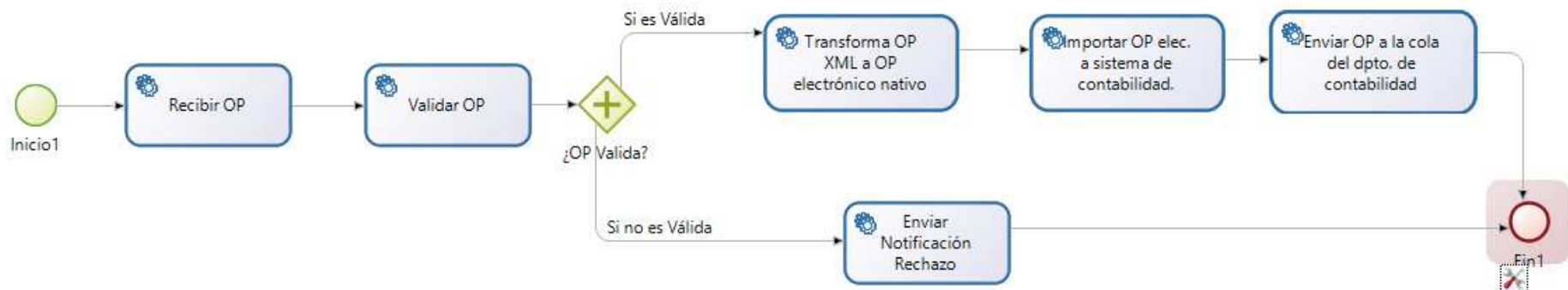
- Crear factura
 - Emitir factura
 - Exportar factura a directorio en red
 - Sondear directorio de red de facturación
 - Recuperar factura
 - Transformar factura a documento XML
 - Comprobar validez del documento
 - Enviar factura



- Caso de estudio

- Modelado del Servicio

- Descomposición del proceso de negocio “Order Fulfillment Process” (Step 1)
 - Recibir documento OP.
 - Validar documento OP.
 - Si documento OP no es válido, enviar notificación de rechazo y finalizar proceso.
 - Transforma documento OP XML a documento OP electrónico nativo.
 - Importar OP electrónico al sistema de contabilidad.
 - Enviar OP a la cola de trabajo del dpto. de contabilidad.



- Caso de estudio
 - Modelado del Servicio
 - Descomposición del proceso de negocio:
 - Algunos pasos dentro de un proceso de negocio pueden ser fácilmente identificados como no pertenecientes a la lógica de potencial que debe ser encapsulada por un candidato de servicio.
 - Al filtrar estas piezas nos quedamos con las etapas de procesamiento más relevantes a nuestro servicio de modelado de proceso.
 - Identificar capacidades candidatas (step 2)
 - Eliminar
 - » Tareas que deben seguir siendo manuales
 - Pasos del proceso manual que no pueden o no deberían automatizarse.
 - » Lógica llevada a cabo por sistemas heredados
 - Pasos de proceso realizados por lógica obsoleta existente para los que el servicio de encapsulación del candidato no es una opción.
 - Proceso de facturación
 - Proceso de Order Fulfillment Process (Orden de Trabajo)

- Proceso de facturación

- Crear factura electrónica (Tarea manual realizada por administrativo)
- Modificar factura electrónica (Tarea manual realizada por administrativo)
- Exportar factura electrónica a un directorio en red (desarrollo propio extensión de un sistema heredado) → podría ser parte de un servicio candidato genérico
- Sondear directorio de red de facturación (realizado por un componente propio) → podría ser parte de un servicio candidato genérico
- Recuperar factura electrónica (realizado por un componente propio) → podría ser parte de un servicio candidato genérico
- Transformar factura electrónica a documento XML (realizado por un componente propio) → podría ser parte de un servicio candidato genérico
- Comprobar validez del documento (parte del actual servicio de envío de facturas) → se mantiene
- Enviar factura (parte del actual servicio de envío de facturas) → se mantiene

- **Proceso de Order Fulfillment Process**
 - Recepción de la OP. (Actualmente, esta siendo realizado por el Order Fulfillment Service. No es necesario cambiar esto.)
 - Validar un documento OP. (Actualmente, esta siendo realizado por el Order Fulfillment Service. No es necesario cambiar esto.)
 - Si el documento OP no es correcto, enviar notificación de rechazo y finalizar el proceso. (Actualmente, esta siendo realizado por el Order Fulfillment Service. No es necesario cambiar esto.)
 - Transforma un documento OP XML en un documento electrónico nativo OP. (realizado por un componente propio) → podría ser parte de un servicio candidato genérico
 - Importar documento electronico OP al sistema de contabilidad. (desarrollo propio extensión de un sistema heredado) → podría ser parte de un servicio candidato genérico
 - Enviar OP a cola de trabajo del dpto. de contabilidad. (desarrollo propio extensión de un sistema heredado) → podría ser parte de un servicio candidato genérico

- Caso de estudio
 - Modelado del Servicio
 - Identificar servicios agnósticos candidatos (step 3)
 - Comprobar validez del documento ?
 - Transformar factura electrónica a documento XML ?

- Caso de estudio

- Modelado del Servicio

- Identificar lógica específica de proceso (step 4)

- En el ejemplo no se quería incluir un motor de orquestación, sólo se añadirá la capa de servicios de tareas

- » Coste / complejidad de administración

- Corresponde a la capa de orquestación de servicios

- » Reglas de negocio

- Empresa es TLS realizar 10% descuento
 - Empresa es ADIF realizar 15% descuento

- » Lógica condicional

- Si el documento factura es válida → enviar factura
 - Si el documento factura no es válido → fin de proceso

- » Lógica de excepción

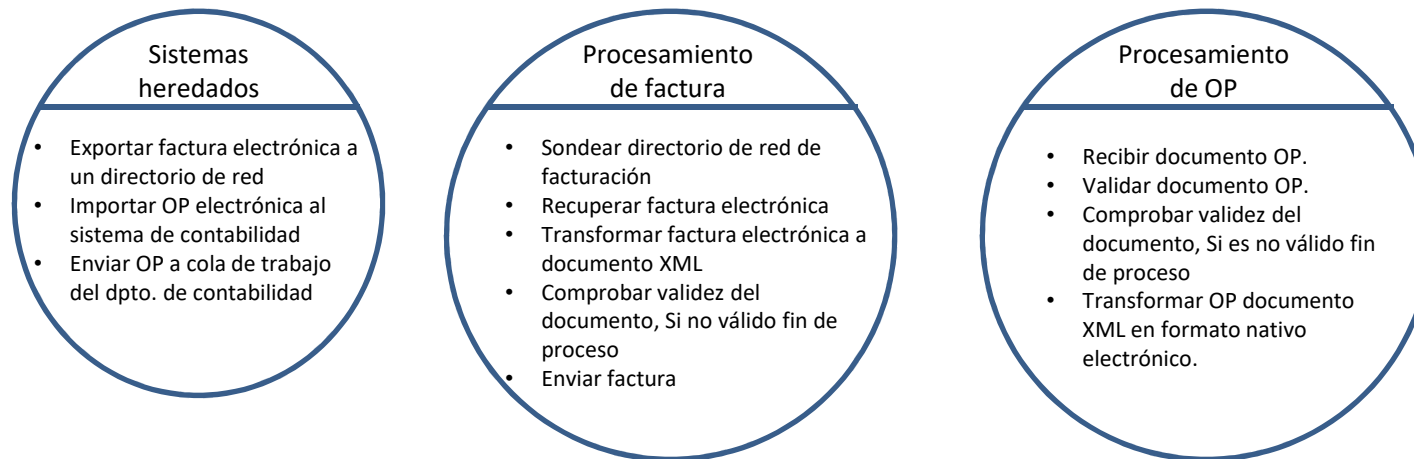
- » Flujo de actividades

- Secuencia de ejecución de las actividades

- Caso de estudio

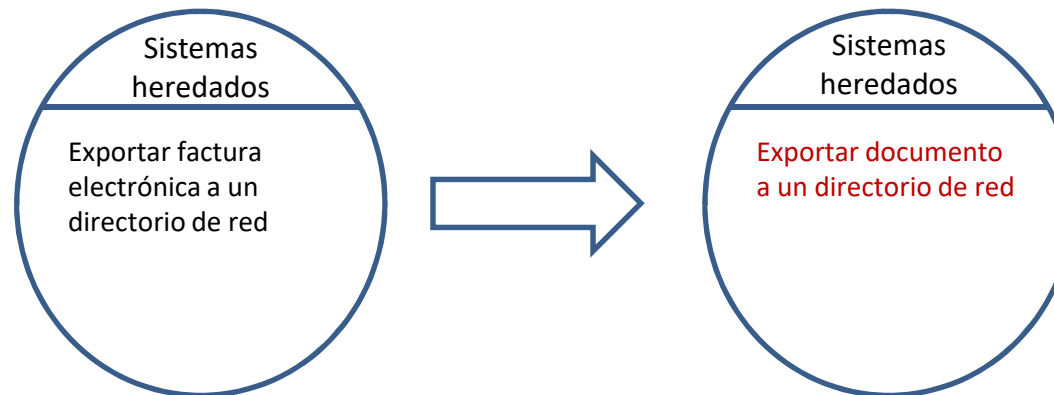
- Modelado del Servicio

- Identificar lógica específica de proceso. (step 4)



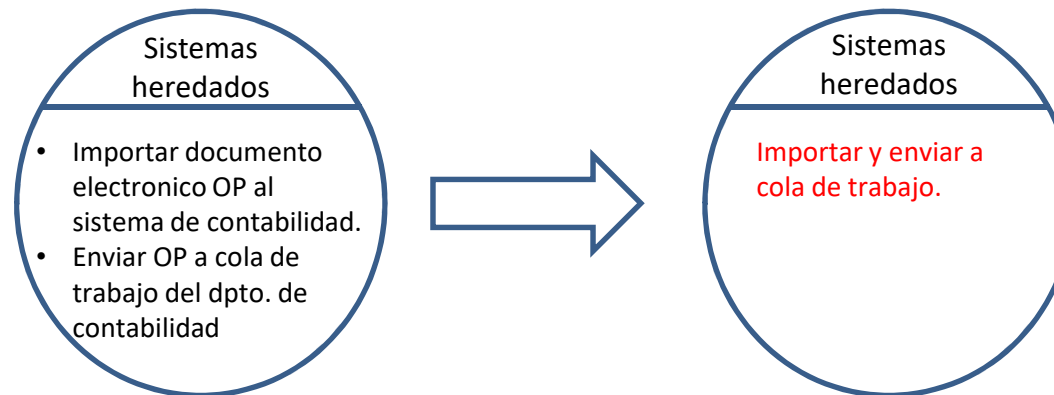
- Caso de estudio

- Modelado del Servicio
 - Aplicar principios de Orientación a Servicios (step 5)
 - Reusabilidad
 - Autonomía
 - Servicio sin estado
 - Descubrimiento
 - Se realizan ajustes dentro de los servicios candidatos propuestos si es necesario
 - Reusabilidad



- Caso de estudio

- Modelado del Servicio
 - Aplicar principios de Orientación a Servicios (step 5)
 - Reusabilidad
 - Autonomía
 - Servicio sin estado
 - Descubrimiento
 - Se realizan ajustes dentro de los servicios candidatos propuestos si es necesario
 - Reusabilidad



- Caso de estudio

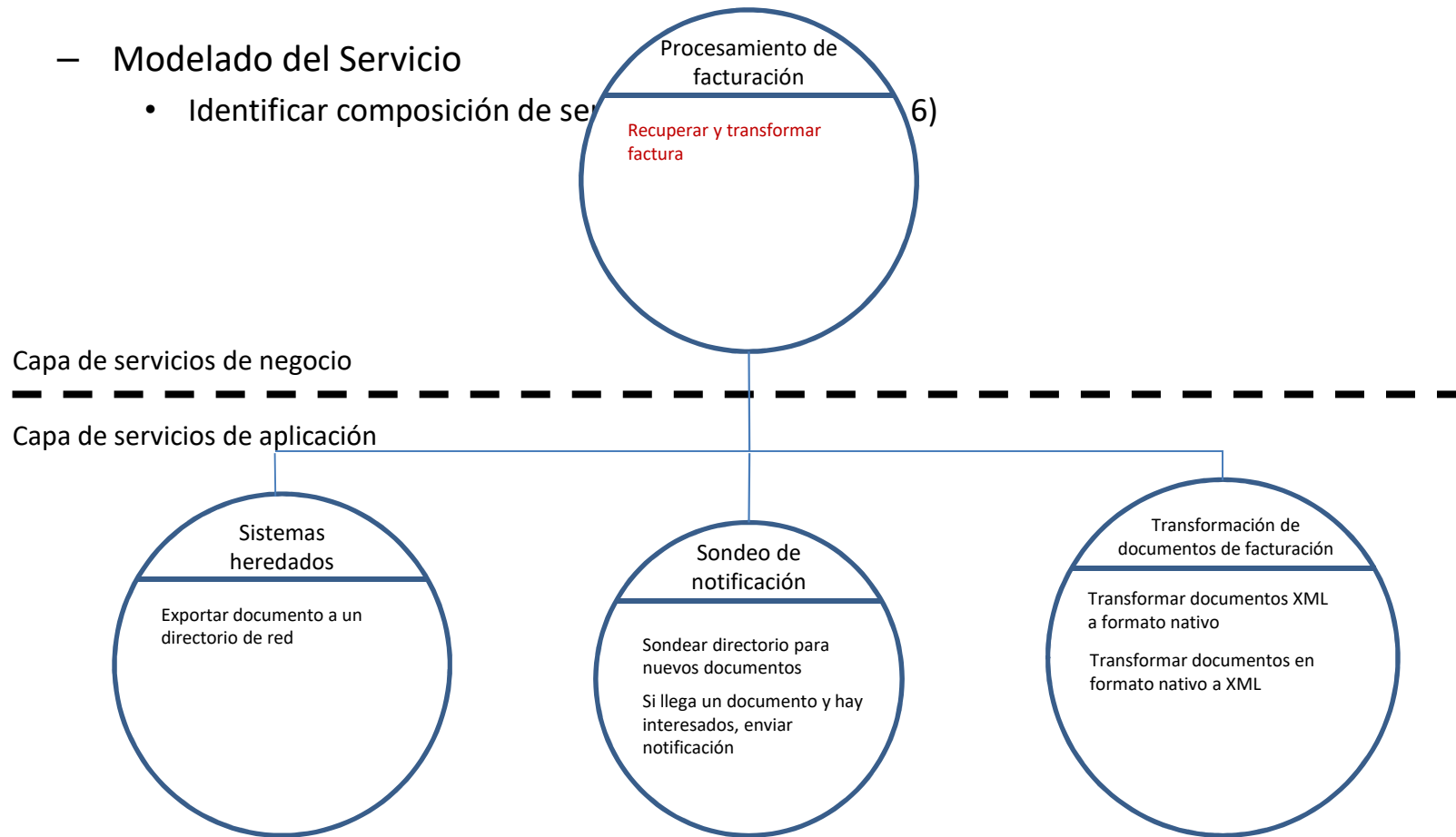
- Modelado del Servicio
 - Aplicar principios de Orientación a Servicios
 - Reusabilidad
 - Autonomía
 - Servicio sin estado
 - Descubrimiento
 - Se realizan ajustes dentro de los servicios candidatos propuestos si es necesario
 - Reusabilidad



- Caso de estudio

- Modelado del Servicio

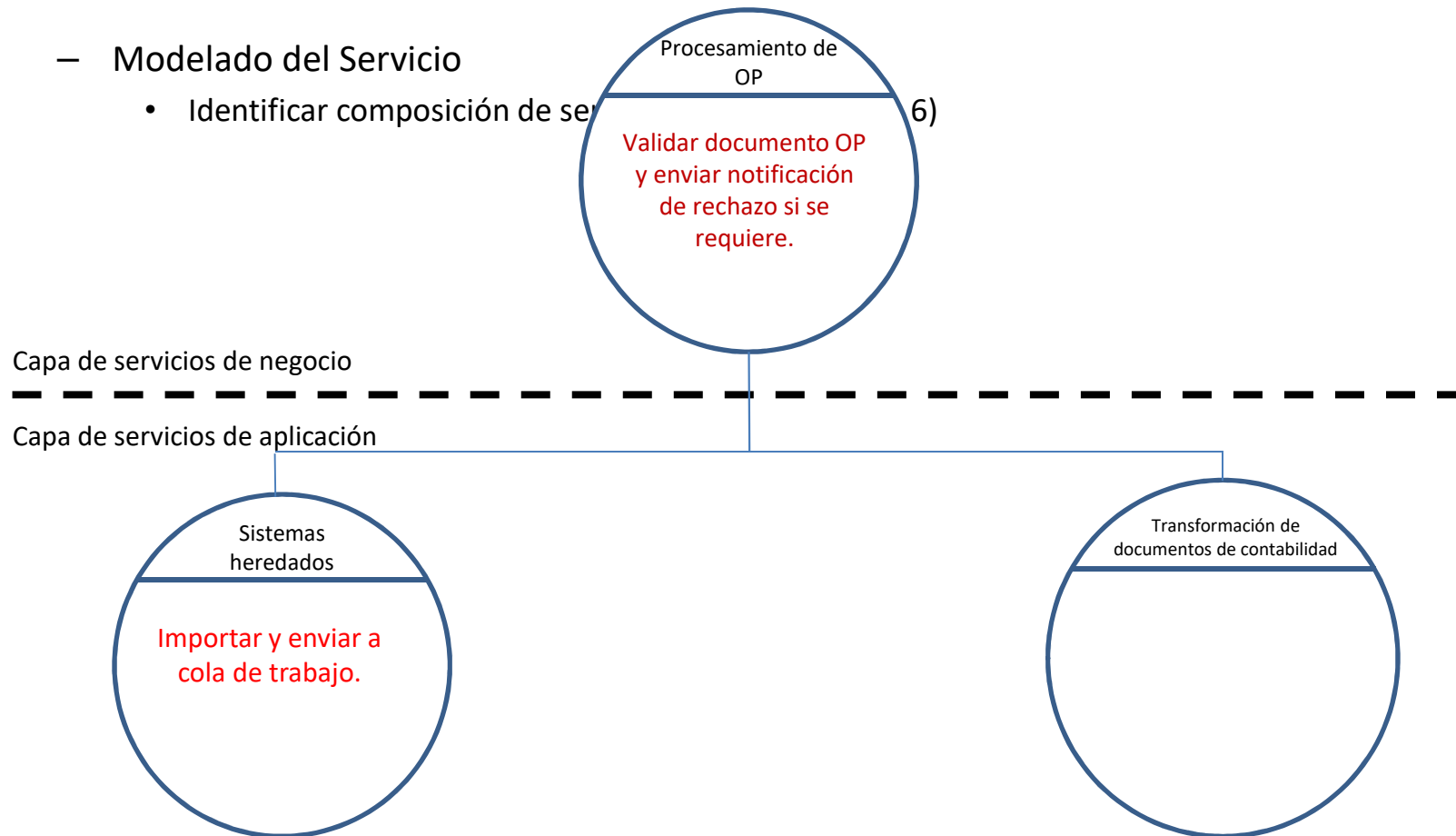
- Identificar composición de ser



- Caso de estudio

- Modelado del Servicio

- Identificar composición de ser



- Caso de estudio
 - Modelado del Servicio
 - Abstract orchestration logic
 - Si se decide construir una capa de orquestación como parte de SOA, entonces se deben identificar las partes de la lógica de procesamiento.
 - Posibles tipos de lógica adecuados para esta capa incluyen:
 - » reglas de negocio
 - » lógica condicional
 - » lógica excepción
 - » secuencia lógica

- Caso de estudio

- Modelado del Servicio

- Abstract orchestration logic

- Sobre la base de nuestros dos descripciones de procesos, la lógica de flujo de trabajo representada por un proceso separado de servicio candidato , incluiría las siguientes:

- » Si el documento de factura no es válido, fin del proceso.
 - » Si el documento de pedido es válido, proceda con el paso de transformar el documento PO.
 - » Si el documento PO no es válido, fin del proceso.

- Caso de estudio
 - Modelado del Servicio
 - Resto de pasos optativos para refinar el modelo
 - Analizar requerimientos de procesamiento
 - Identificar capacidades candidatas de servicios de utilería
 - Definir servicios de utilería candidatos
 - Aplicar principios de orientación a servicio
 - Revisar composición de servicios candidatos
 - Revisar agrupamiento de capacidades candidatas



Service-Oriented Architecture: Concepts, Technology, and Design

By Thomas Erl

.....
Publisher: **Prentice Hall PTR**

Pub Date: **August 04, 2005**

ISBN: **0-13-185858-0**

Pages: **792**