

# Ingeniería de Requisitos

---

## Tema 6. Validación y Gestión de Requisitos

*Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Alicante*

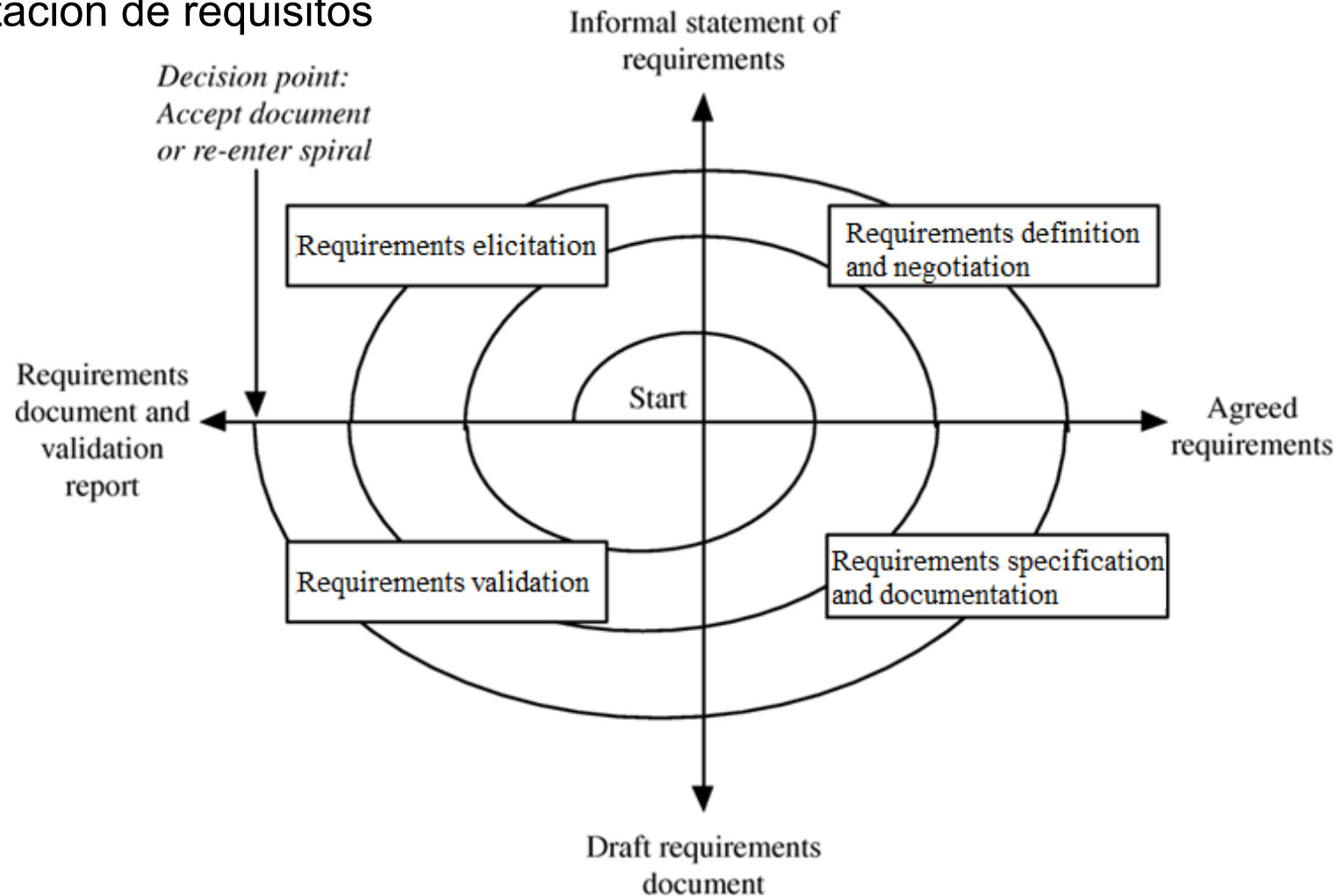
*Materiales cedidos para su uso docente por el prof. Emilio Insfrán Pelozo. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia*

# Contenido

- Introducción
- Validación
  - Métodos
  - Técnicas
    - Revisiones
    - Prototipos
    - Pruebas
- Gestión de requisitos

# Introducción

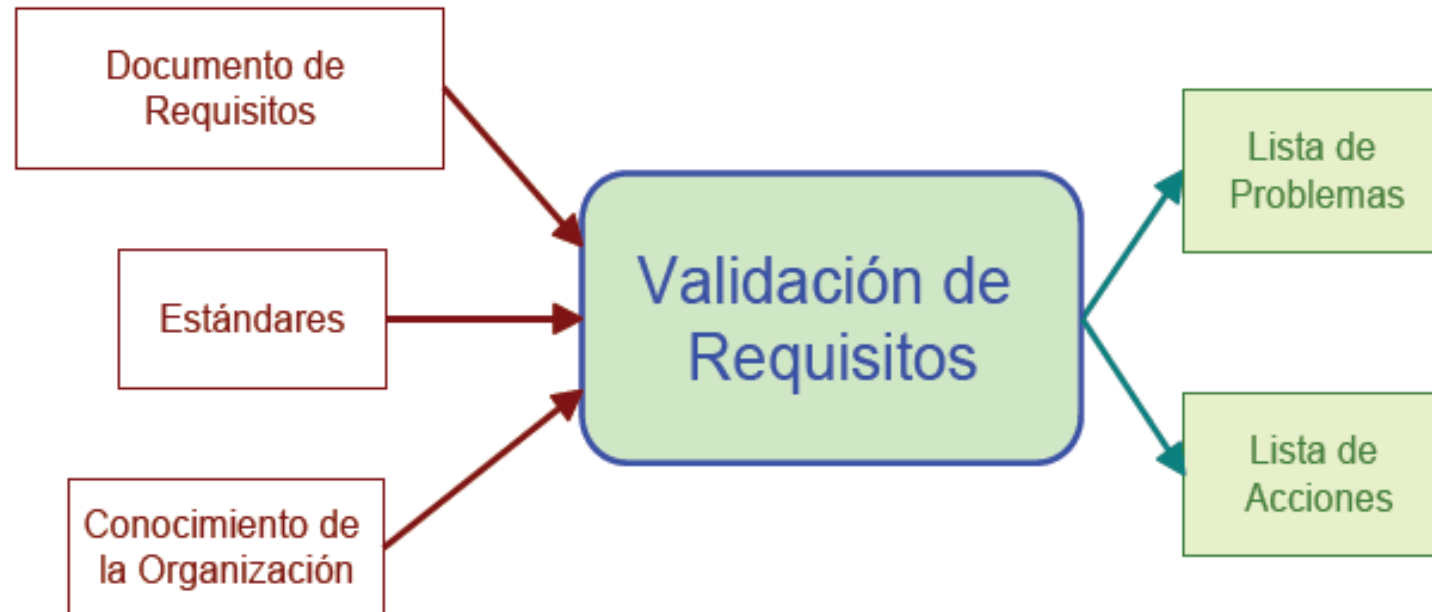
- El proceso de **validación de requisitos** comprende actividades que generalmente se realizan una vez obtenida una primera versión de la documentación de requisitos



# Validación

- Los requisitos deben ser validados para
  - Asegurar que el ingeniero software los ha comprendido
  - El documento de ERS es conforme a los estándares establecidos, comprensible, consistente y completo.
- Se trata de asegurar que el documento ERS define **el software adecuado**, es decir, **el que espera el usuario**.
- Las principales técnicas de validación:
  - Revisiones
  - Prototipado
  - Validación de Modelos [X]
  - Pruebas de Aceptación [X]

# Validación



## **Análisis**

**- ¿Tenemos los requisitos correctos?**

## **Validación**

**- ¿Tenemos los requisitos descritos correctamente?**

# Validación

- Comprobaciones a realizar:
  - **Validez.** El sistema provee las funciones que soportan mejor las necesidades del cliente.
  - **Consistencia.** No existen conflictos entre los requisitos.
  - **Compleitud.** Están incluidas todas las funciones requeridas por el usuario.
  - **Realismo.** Los requisitos pueden ser implementados con el presupuesto y tecnologías disponibles.
  - **Verificable.** Los requisitos pueden ser verificados.

# Validación

- La validación se realiza a través de diversos métodos.
- Los 3 métodos más habituales son:
  - **Revisión de requisitos:** consisten en reuniones donde un equipo de analistas intenta localizar errores en el documento de especificación.
  - **Prototipado:** consiste en construir una maqueta del futuro sistema software a partir de los requisitos recogidos en la especificación. Esta maqueta será evaluada por el cliente y usuarios para comprobar su corrección y completitud.
  - **Generación de casos de prueba (test de requisitos):** tiene como objetivo comprobar la verificabilidad de los requisitos. Consiste en la definición de casos de aceptación que permitan verificar el cumplimiento de los requisitos funcionales.

# Validación

- Además de los tres métodos anteriores, existe un cierto número de otras técnicas alternativas:
  - **Creación de manuales de usuario:** consiste en verificar si la especificación de requisitos contiene el suficiente detalle como para preparar el manual de usuario del sistema. De no ser así, podría ocurrir que la especificación fuera incompleta.
  - **Animación y validación de modelos o especificaciones formales:** sólo puede emplearse cuando los requisitos se han utilizado modelos durante el análisis o métodos de especificación formal durante la documentación. Consiste en la animación tanto de modelos como de especificaciones formales, con el objetivo de verificar que el funcionamiento del sistema, tal y como está especificado, es el deseado por cliente y usuarios.



# Validación

## Técnicas de Validación de Requisitos:

- **Revisiones [inspecciones]**
  - Un grupo de personas lee, analiza, discute el documento de ERS y las formas de solucionar los problemas detectados.
  - Deben participar representantes del cliente y los usuarios.
  - Norma: IEEE Std 1028-1997, IEEE Standard for Software Reviews
- Las inspecciones pueden ser formales (documento concluido) o informales.

# Validación

## Técnicas de Validación de Requisitos:

- **Revisiones [revisiones]**

Se pueden usar Listas de Comprobación (Checklist):

- **Unicidad**

- Está cada requisito identificado de forma unívoca?

- **Comprensibilidad**

- ¿Se describen en el glosario los términos especializados o técnicos?
- ¿Es cada requisito autocontenido o se tienen que examinar otros requisitos para comprenderlo?
- ¿Hay contradicciones?
- ¿Están agrupados los requisitos relacionados?

- **Trazabilidad**

- ¿Está establecido adecuadamente el origen de cada requisito?
- ¿Está establecida adecuadamente las dependencias entre requisitos?

- **Adaptabilidad**

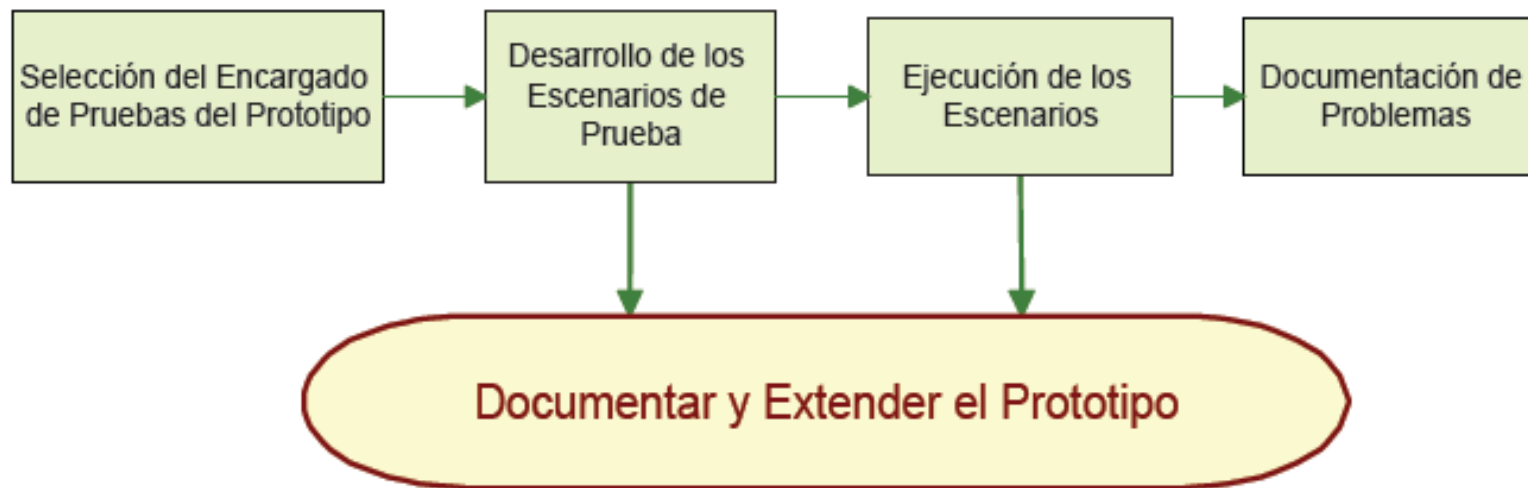
- ¿Puede cambiarse un requisito sin impactar demasiado a los otros?

# Validación

## Técnicas de Validación de Requisitos:

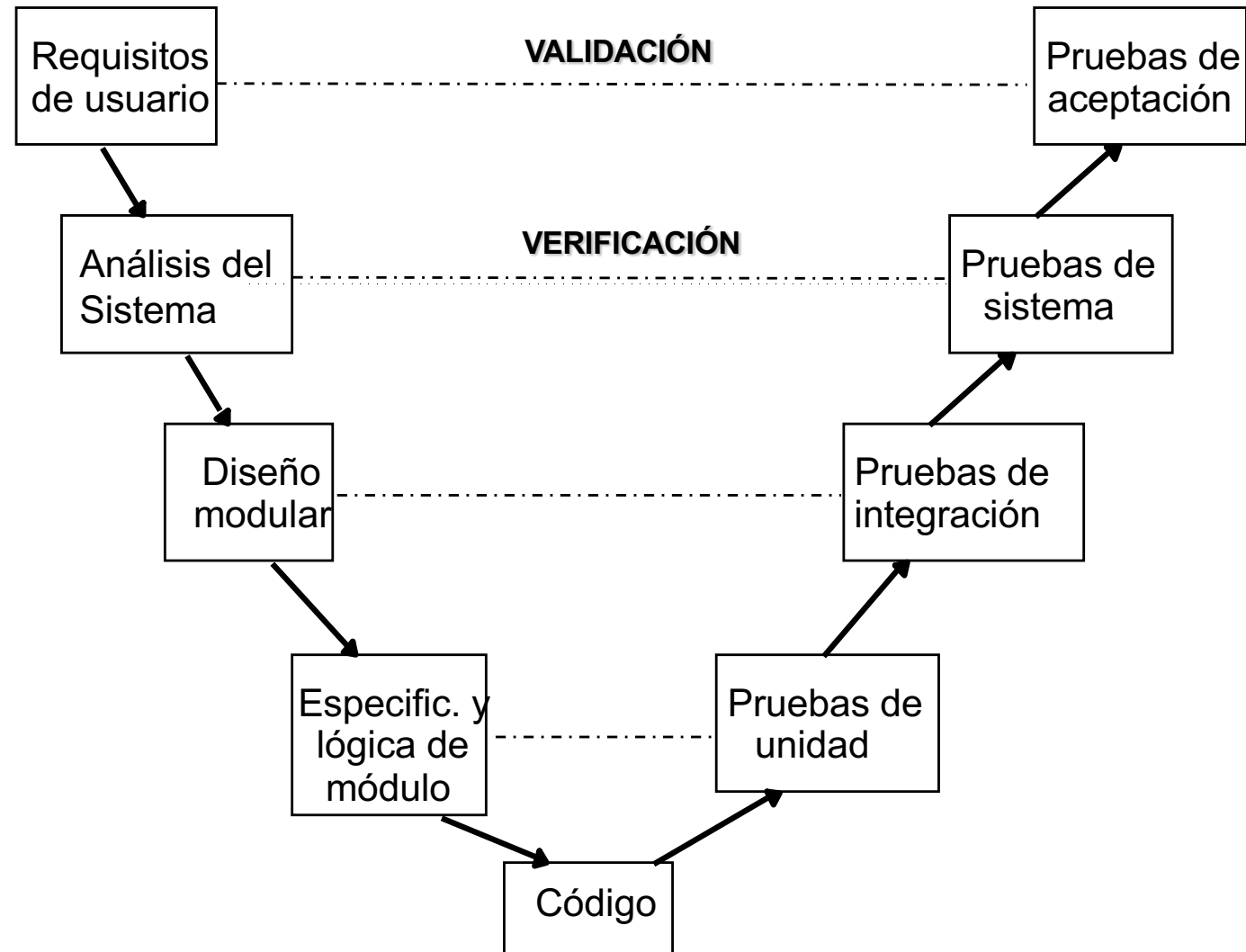
### ■ **Prototipos:**

- Si se desarrolló durante la Elicitación se sigue con su desarrollo en esta fase



- Escribir el **Manual de Usuario** fuerza a un análisis detallado de los requisitos y así revelarse nuevos problemas

# Validación: Pruebas



## Validación: Pruebas

- Pruebas de unidad: se prueba cada módulo individualmente.
- Prueba funcional o de integración: el software totalmente ensamblado se prueba como un conjunto, para comprobar si cumple o no tanto los requisitos funcionales como los requisitos de rendimiento, seguridad, etc.
- Prueba del sistema: el software ya validado se integra con el resto del sistema (por ejemplo, elementos mecánicos, interfaces electrónicas, etc.) para probar su funcionamiento conjunto .
- Prueba de aceptación: el producto final se comprueba por el usuario final en el entorno de desarrollo (alfa) o en el propio entorno de explotación (beta) para determinar si lo acepta o no.

## Validación: Especificación de Pruebas de Aceptación

- En RUP, pruebas de aceptación → lista de *Ideas de Pruebas*
- Este artefacto enumera ideas, a menudo parcialmente formadas, que identifican las pruebas potencialmente útiles de realizar.
- Una idea de prueba (a veces conocida como requisito de prueba) es una sentencia breve sobre una prueba de aceptación a realizar.

# Validación: Especificación de Pruebas de Aceptación

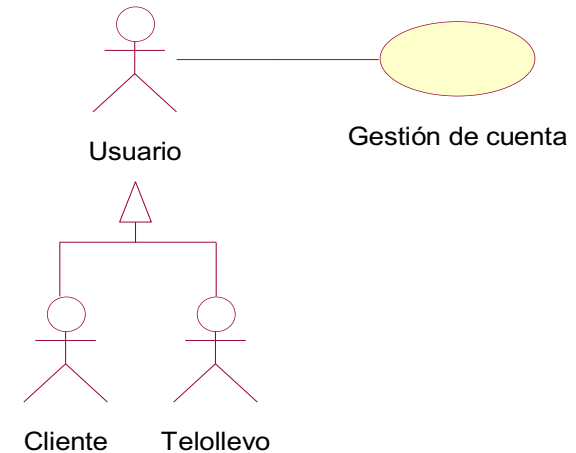
## Ejemplo 1

### CU Gestión de cuenta\*

Pruebas de Aceptación:

1. Alta con nombre de usuario no repetido.
2. No permitir el acceso si el usuario y la contraseña no son válidos.
3. No se permite dar de baja a un usuario si tiene compras asignadas.
4. No se permite modificar los datos de un usuario si tiene compras asignadas.

**\*Obs:** *CU Gestión de cuenta* agrupa a los casos de uso: Alta, baja, modificación y login



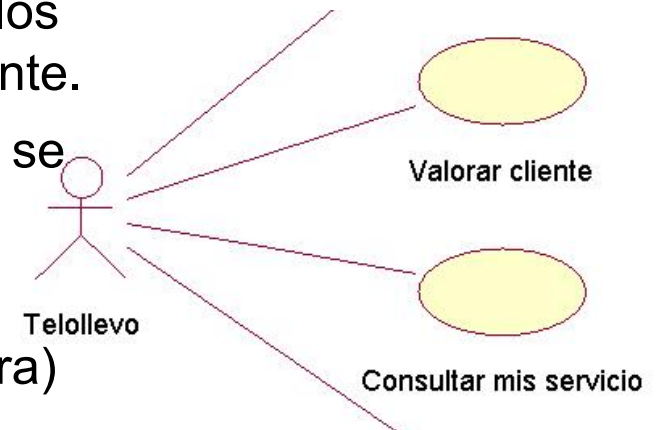
# Validación: Especificación de Pruebas de Aceptación

## Ejemplo 2

### **CU Valorar cliente**

Pruebas de Aceptación:

1. Solo se puede *valorar (calificar)* a los clientes a los cuales se les ha servido una compra anteriormente.
2. Para una compra determinada a un cliente, sólo se le puede valorar 1 vez.
3. La nota sólo admitirá valores entre 0 y 10.
4. Si se vuelve a votar al mismo cliente (otra compra) la nota media se recalcula correctamente.



### **CU Consultar mis servicios**

Pruebas de Aceptación:

1. Si no hay servicios realizados se muestra un mensaje indicándolo.
2. Si hay más de 25 servicios realizados se paginan.

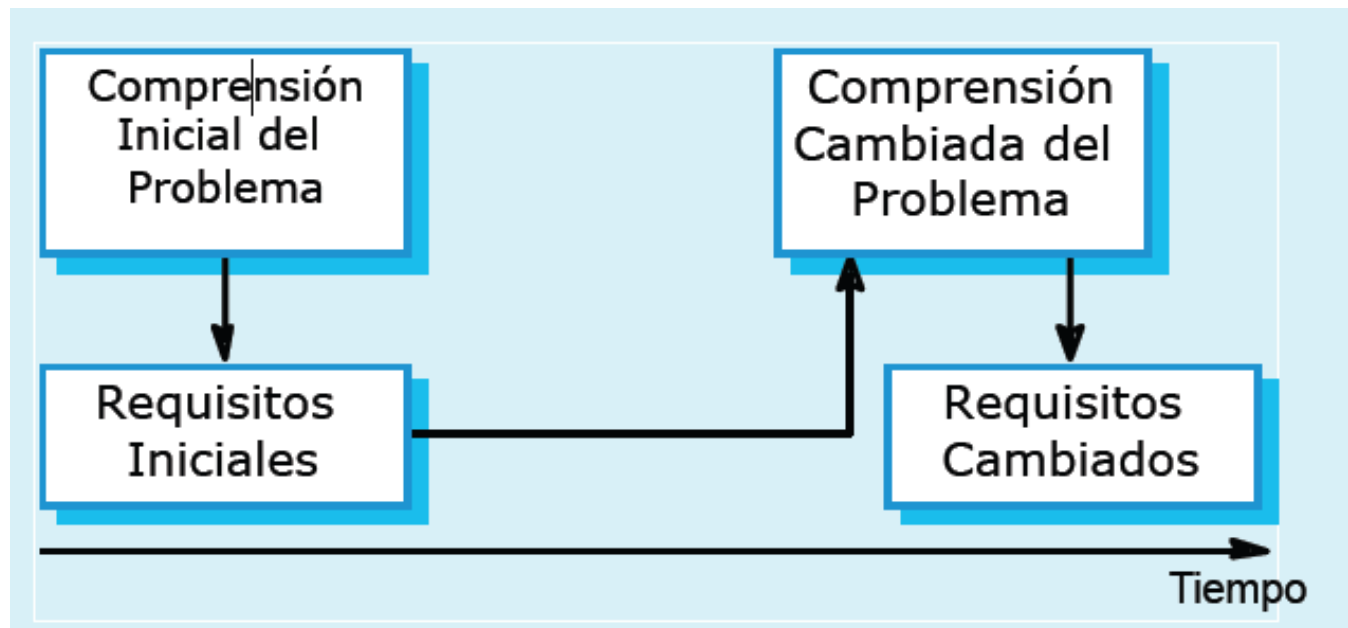


# Gestión de Requisitos

- **Los requisitos cambian:**
  - Evolutivos:
    - Cambio en la estrategia o prioridades del negocio
    - Cambios tecnológicos
    - Cambios en leyes o regulaciones
  - Correctivos
    - Cambios por mejor entendimiento del dominio del problema
- **La Gestión de Requisitos consiste en gestionar**
  - Los cambios en los requisitos acordados
  - Las relaciones entre requisitos
  - Las dependencias entre el documento ERS y otros documentos

# Gestión de Requisitos

- Más del 50% de los requisitos suelen cambiar durante la elaboración de la ERS, y además durante el desarrollo e incluso después de la entrega (mantenimiento)



# Gestión de Requisitos

- La **Gestión de Requisitos** implica la **recolección**, almacenamiento y mantenimiento de grandes cantidades de información
- Para ello existen herramientas CASE con:
  - Base de Datos para almacenar requisitos
  - Facilidades de análisis y generación de documentos
  - Facilidades de gestión de cambios
  - Facilidades de Trazabilidad

# Gestión de Requisitos

Algunos requisitos cambian más que otros

- **Requisitos Estables:** esencia del sistema y su Dominio de Aplicación
- **Requisitos Volátiles:** específicos del sistema en un entorno particular para un cliente particular
- **Las causas del cambio son variadas:**
  - Corrección de errores y problemas en los requisitos
  - Conocimiento creciente del cliente/usuario
  - Problemas técnicos, de calendario o costes
  - Cambio en las prioridades del cliente
  - Cambios en el entorno
  - Cambios organizacionales
  - Requisitos emergentes durante el diseño / implementación
  - Asunciones erróneas
  - Dependencias de otros equipos y procesos

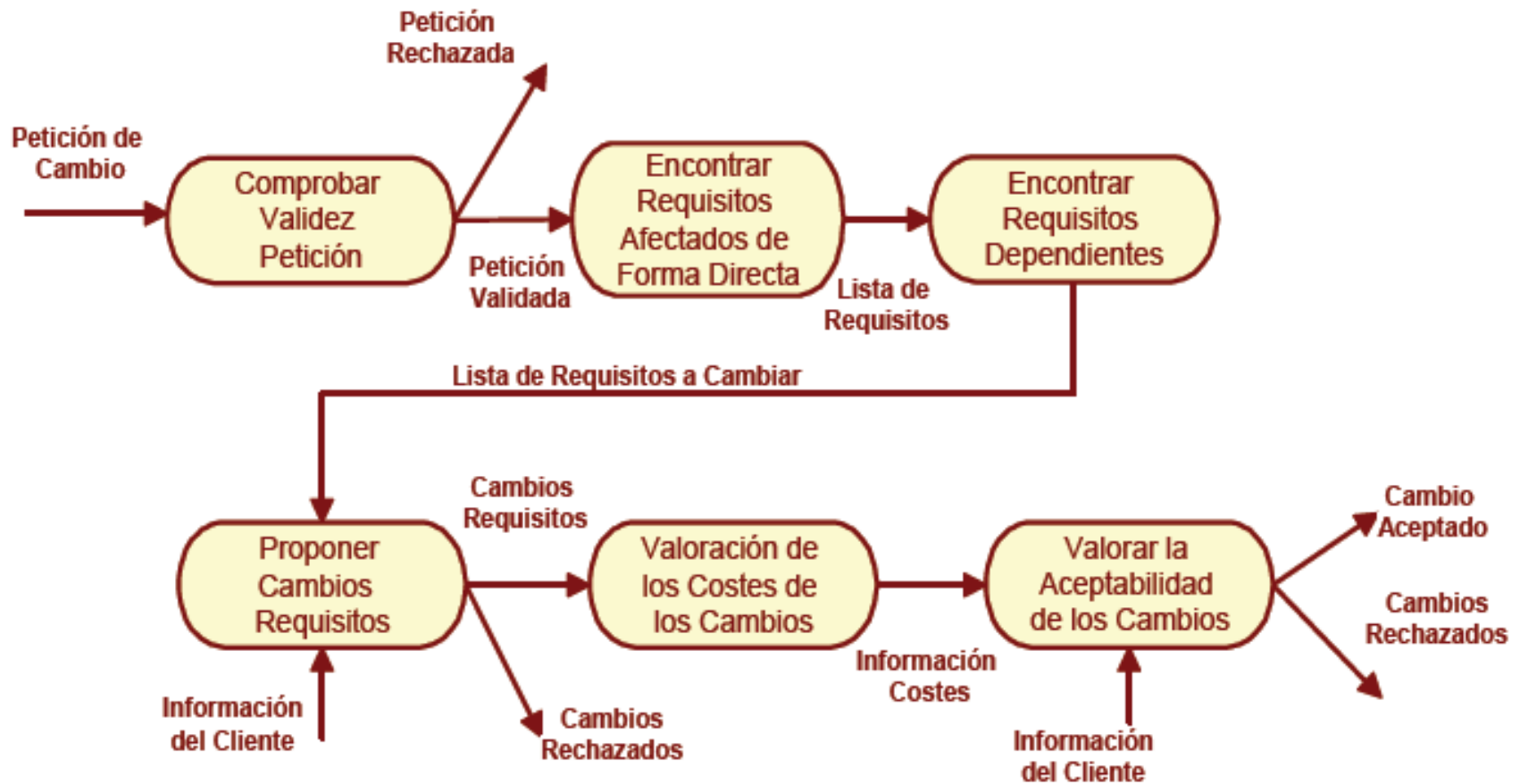
# Gestión de Requisitos

Es importante poder identificar y almacenar los requisitos

- **Identificación de Requisitos**
  - Asignación no ambigua de identificadores
  - Renumeración dinámica, Identificación de Registro de BD, Identificación simbólica

# Gestión de Requisitos

- Gestión de los cambios:
  - Identificación del cambio / Análisis del cambio / Implementación del cambio



# Ingeniería de Requisitos

## Proceso de IDR. Gestión de Requisitos. Trazabilidad (1/3)

---

- Trazabilidad de Requisitos significa que requisitos relacionados están *conectados* de alguna forma
- La trazabilidad es una propiedad de la especificación de requisitos que refleja la facilidad de identificar requisitos relacionados
  - Los requisitos deben tener asignados *identificadores* únicos
  - Las *referencias cruzadas* relacionan requisitos usando estos identificadores
  - Producen una matriz para cada documento de requisitos mostrando requisitos relacionados. Muchas matrices pueden ser necesarias para representar diferentes tipos de relaciones
- Las herramientas CASE proveen facilidades de *soporte a la trazabilidad*

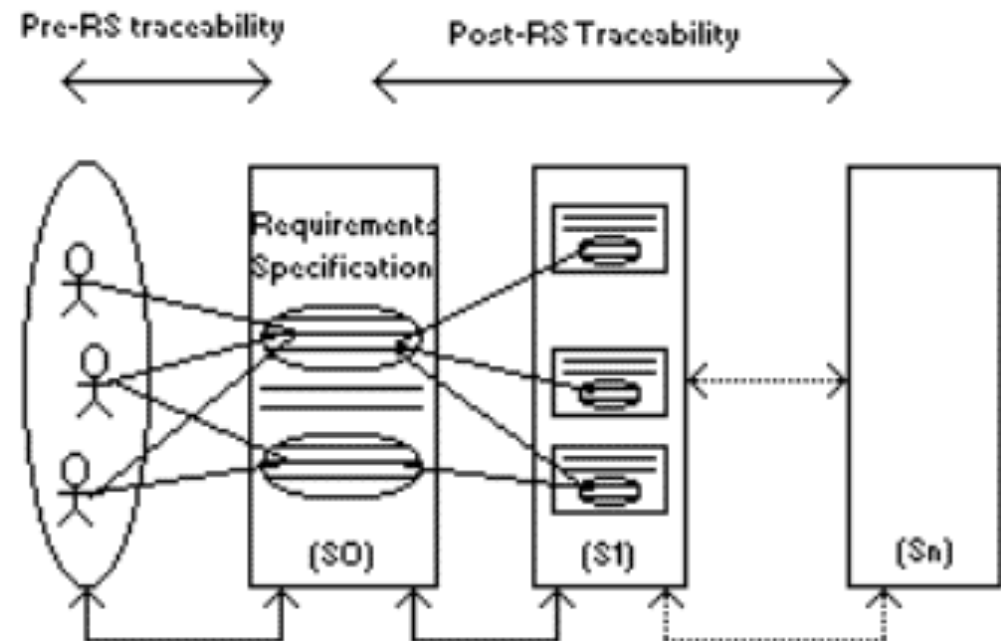


# Ingeniería de Requisitos

## Proceso de IDR. Gestión de Requisitos. Trazabilidad (2/3)

IEEE Std. 830:

- **Backward traceability (Pre Trazabilidad):** conecta los requisitos con su origen en otros documentos
- **Forward traceability (Post Trazabilidad):** conecta los requisitos con componentes de diseño o implementación

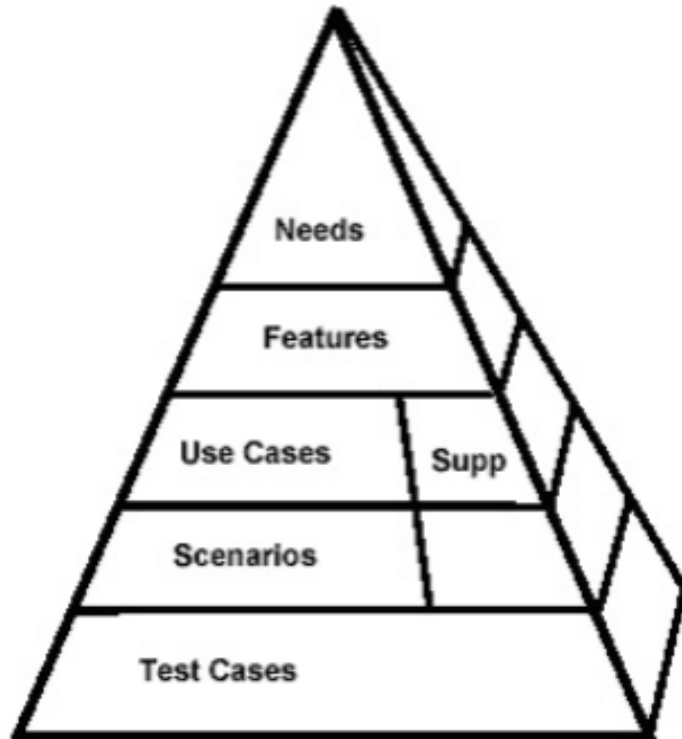




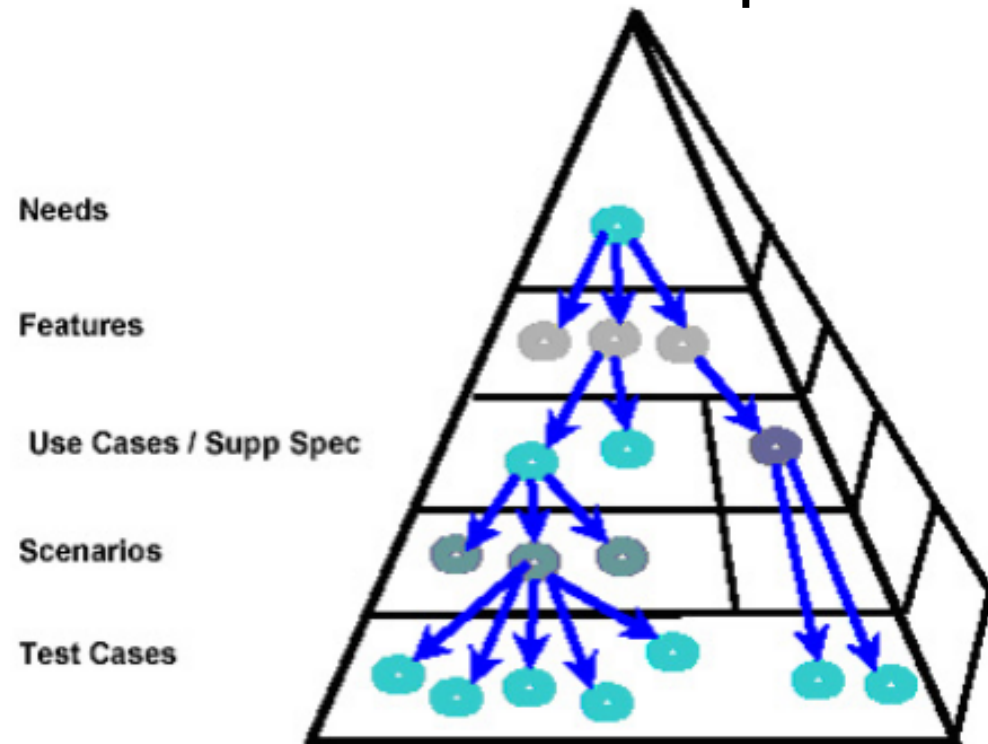
# Ingeniería de Requisitos

## Proceso de IDR. Gestión de Requisitos. Trazabilidad (3/3)

### Pirámide de Requisitos



### Trazabilidad en la Pirámide de Requisitos



**Needs:** Necesidades de usuarios o del negocio (objetivos)

**Features** (características). Funcionalidad de alto nivel. Agrupa funcionalidades más específicas

**Use cases:** Casos de Uso, son los Requisitos Funcionales

**Supp:** Requisitos No Funcionales (RNF) o Requisitos suplementarios

**Scenarios:** descripción de la funcionalidad de RF o casos de uso

**Test cases:** Pruebas que se realizan para comprobar los RF y RNF

# Gestión de Requisitos - Trazabilidad

## Información para **Trazabilidad de Requisitos**

- Backward-from Fuentes  $\leftarrow$  R
  - Enlace de Requisitos con sus fuentes
- Forward-from R  $\rightarrow$  D-I
  - Enlace Requisitos con Diseño e Implementación
- Backward-to R  $\leftarrow$  D-I
  - Enlace Diseño e Implementación con Requisitos
- Forward-to Fuentes  $\rightarrow$  R
  - Enlaza documentos previos con los Requisitos
- Formatos:
  - Matriz de Trazabilidad (arriba)
  - Lista de Trazabilidad (abajo)

Depends-on

	R1	R2	R3	R4	R5	R6
R1			*	*		
R2					*	*
R3				*	*	
R4		*				
R5						*
R6						

Requisito	Depende de
R1	R3, R4
R2	R5, R6
R3	R4, R5
R4	R2
R5	R6