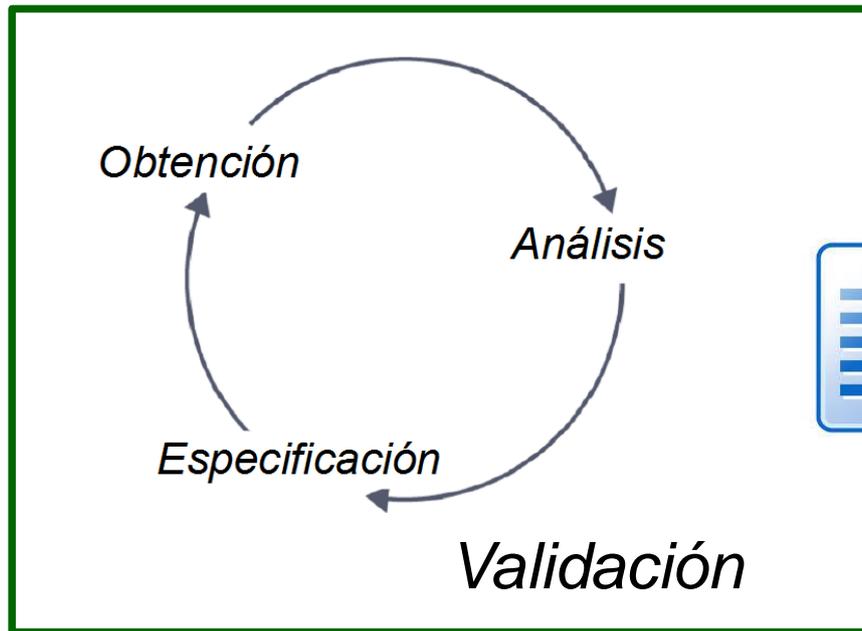


# Fundamentos de Ingeniería de Software

## Ingeniería de requisitos Clase 3

# Actividades de la ingeniería de requisitos

## Desarrollo de requisitos



## Gestión de requisitos

- .Planificación*
- .Gestión de cambios*
- .Trazabilidad*



# Temario

- Documento de requisitos
- Priorización de requisitos
- Validación de requisitos
- Gestión de requisitos

# Documento de requisitos de software

- El documento de requisitos de software (SRS) es una declaración oficial de lo que se requiere del sistema.
- Debe incluir una definición de los requisitos del usuario y una especificación de los requisitos del sistema.
- No es un documento de diseño. Se debe indicar lo que el sistema debe hacer y no la forma en la que lo debe hacer.
- La información del documento de requisitos depende del tipo de sistema y el enfoque de desarrollo usado.
  - Metodologías ágiles prefieren, en lugar de construir un documento formal, recolectar requisitos de forma separada (ejemplo: historias de usuario).
  - Los sistemas desarrollados de forma incremental tienen típicamente menos detalles en el documento de requisitos

# Estructura de documento de requisitos

Capítulo	Descripción
Prefacio	Definir los lectores esperados y describir el historial de versiones, incluyendo una justificación para la creación de una nueva versión y un resumen de los cambios realizados en cada versión.
Introducción	Describir las necesidades y las funciones del sistema y explicar como va a funcionar con otros sistemas. También debe describir como el sistema se ajusta a los objetivos del negocio o a los planes estratégicos de la organización.
Glosario	Definir los términos técnicos usados en el documento. No se deben hacer suposiciones acerca de la experiencia o de los conocimientos del lector.
Definición de requisitos de usuario	Aquí, se describen los servicios prestados para el usuario. En esta sección también deben ser descritos los requisitos no funcionales del sistema. Esta descripción puede utilizar el lenguaje natural, diagramas u otra notación entendible por los clientes. Las normas de producto y proceso que deben seguir deben ser especificados.
Arquitectura del sistema	Este capítulo debe presentar una visión de alto nivel de la arquitectura del sistema, mostrando la distribución de las funciones sobre los módulos del sistema.

# Estructura de documento de requisitos

Capítulo	Descripción
Especificación de requisitos del sistema	Describir con más detalle los requisitos funcionales y no funcionales, así como interfaces con otros sistemas.
Los modelos del sistema	Aquí se puede incluir modelos gráficos del sistema que muestren las relaciones entre los componentes del sistema y la relación con su entorno.
Evolución del sistema	Describir los supuestos fundamentales en los cuales está basado el sistema y cualquier cambio previsto debido a la evolución del hardware, cambios en las necesidades de los usuarios, etc. Esta sección es útil para los diseñadores del sistema ya que puede ayudar a evitar que las decisiones de diseño restrinjan posibles cambios futuros.
Apéndices	Debería proporcionar información detallada y específica relacionada con la aplicación que se está desarrollando, por ejemplo hardware y descripción de la base de datos.
Índice	Se pueden incluir varios índices. Entre ellos puede ser un índice alfabético, índice de diagramas, índice de funciones, etc.

# Ejemplo de un SRS – IEEE 29148:2011

- 1. Introduction**
  - 1.1 Purpose
  - 1.2 Scope
  - 1.3 Product overview
    - 1.3.1 Product perspective
    - 1.3.2 Product functions
    - 1.3.3 User characteristics
    - 1.3.4 Limitations
  - 1.4 Definitions
- 2. References**
- 3. Specific requirements**
  - 3.1 External interfaces
  - 3.2 Functions
  - 3.3 Usability Requirements
  - 3.4 Performance requirements
  - 3.5 Logical database requirements
  - 3.6 Design constraints
  - 3.7 Software system attributes
  - 3.8 Supporting information
- 4. Verification**
  - (parallel to subsections in Section 3)
- 5. Appendices**
  - 5.1 Assumptions and dependencies
  - 5.2 Acronyms and abbreviations

Figure 8 — Example SRS Outline

# Documento de requisitos - Consideraciones

- › Registrar los requisitos en términos del cliente.
- › Características de una buena especificación:
  - › Correcta / Válida
  - › No ambigua
  - › Completa
  - › Consistente internamente
  - › Agrupados y ordenados por importancia y/o estabilidad
  - › Verificable
  - › Modificable
  - › Trazable
  - › Factibles
  - › Entendible

# Priorización de requisitos

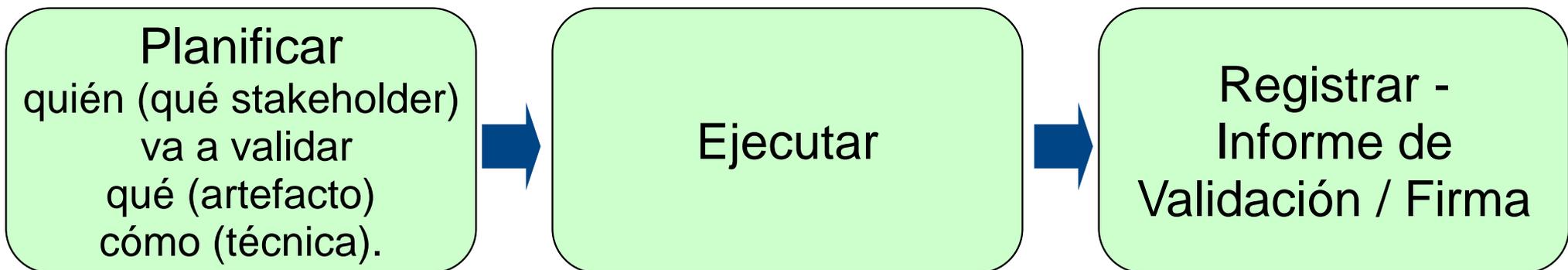
- Todo proyecto que tenga recursos limitados necesita priorizar qué requisitos incluir en el alcance del producto de software.
- La priorización ayuda a entregar el mayor valor de negocio tan rápido como sea posible con las restricciones existentes.
- A veces los clientes no quieren priorizar requisitos pensando que no se hará lo que sea de prioridad baja.
- Los programadores por su lado no suelen priorizar porque eso daría la impresión de que no son capaces de realizar todos los requisitos.
- Un estudio muestra que cerca de dos tercios de las características de los sistemas de software se usan raramente o nunca (*The Standish Group 2009*).
  - ¿Qué sucede si terminamos el proceso de priorización con todos los requisitos con la misma prioridad?

# Validación de requisitos

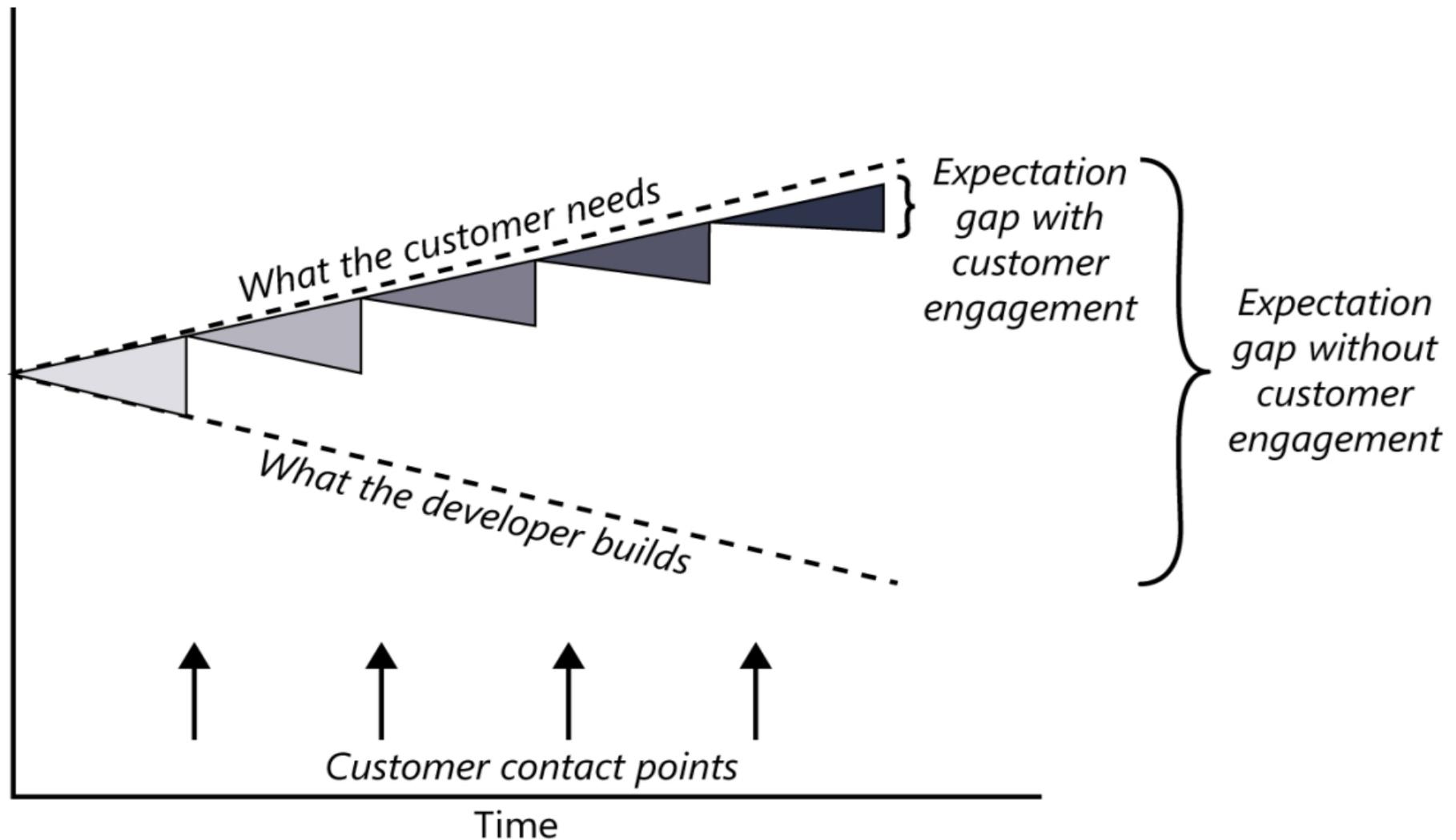
Proceso por el cual se determina si los requisitos relevados son consistentes con las necesidades del cliente.

- **Verificación** — determina si el producto de alguna actividad de desarrollo cumple los requisitos (hacer las cosas bien).
- **Validación** — evalúa si un producto satisface las necesidades del cliente (hacer la cosa correcta).

➤ Proceso:



# Expectativas del cliente



# Validación de requisitos no funcionales

- Son difíciles de validar.
- Se deben expresar de manera cuantitativa utilizando métricas que se puedan probar de forma objetiva (esto es ideal).

Propiedad	Medida
Rapidez	Transacciones por seg
Tamaño	KB
Fiabilidad	Tiempo promedio entre fallas
Portabilidad	Número de sistemas, especificar
Facilidad de uso	Tiempo de capacitación

- Para los usuarios es difícil especificarlos en forma cuantitativa.

# ¿Cómo validar requisitos?

## › Manuales

- › Lectura por parte del cliente.
- › Recorridas. Útiles con muchos *stakeholders* que no lo leerían de otra manera.
- › Entrevistas.
- › Chequeo manual de referencias cruzadas.
- › Instancias de validación formal:
  - › Revisiones - *Stakeholders* revisan por separado y se reúnen para discutir problemas.
  - › Inspecciones formales – roles y reglas.
- › Listas de comprobación.
- › Generación de casos de prueba.

## › Automatizadas

- › Chequeo automático de referencias cruzadas.
- › Ejecución de modelos para verificar funciones y relaciones.
- › Construcción de prototipos.
- › Simulaciones.

# Revisión de requisitos

- Revisiones periódicas deben mantenerse mientras se formulan los requisitos.
- El cliente y el equipo de desarrollo deben estar involucrados en las revisiones.
- Las revisiones pueden ser formales (con documentos completos) o informales. Una buena comunicación entre desarrolladores, clientes y usuarios puede resolver problemas en fases tempranas.

# Ejemplo de checklist de defectos

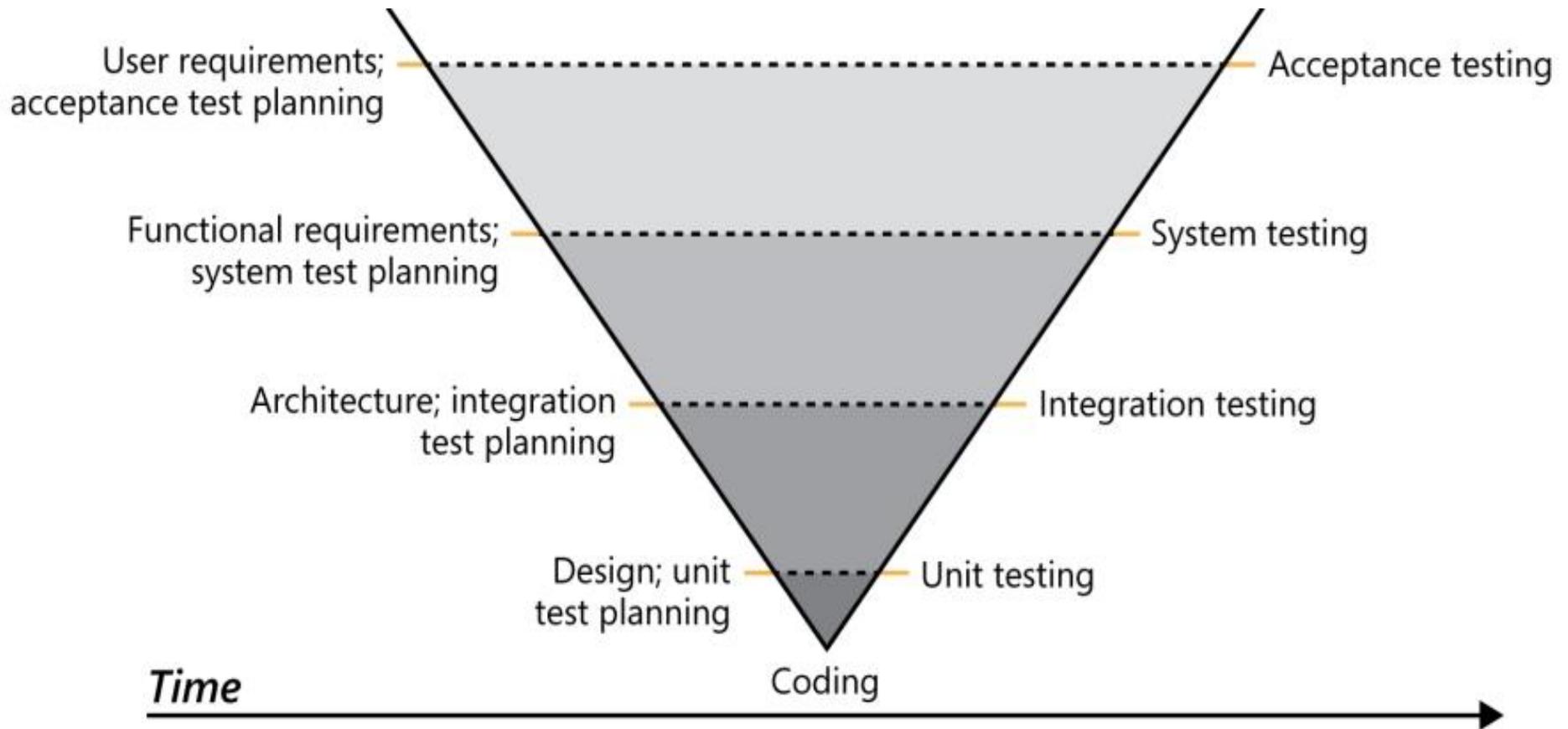
## Completeness

- Do the requirements address all known customer or system needs?
- Is any needed information missing? If so, is it identified as TBD?
- Have algorithms intrinsic to the functional requirements been defined?
- Are all external hardware, software, and communication interfaces defined?
- Is the expected behavior documented for all anticipated error conditions?
- Do the requirements provide an adequate basis for design and test?
- Is the implementation priority of each requirement included?
- Is each requirement in scope for the project, release, or iteration?

## Correctness

- Do any requirements conflict with or duplicate other requirements?
- Is each requirement written in clear, concise, unambiguous, grammatically correct language?
- Is each requirement verifiable by testing, demonstration, review, or analysis?
- Are any specified error messages clear and meaningful?
- Are all requirements actually requirements, not solutions or constraints?
- Are the requirements technically feasible and implementable within known constraints?

# ¿Recuerdan...?



# Validar utilizando criterios de aceptación

- El cliente es el que tendrá la opinión final sobre un producto. La idea es entonces establecer, de forma complementaria a la definición de los requisitos, los criterios con los cuales se aceptarán.
- Los criterios de aceptación no son pruebas funcionales o unitarias; sino condiciones que deben cumplir el sistema.
- Las pruebas son más completas y prueban todos los flujos funcionales, excepciones, límites, etc.

# Criteria de aceptación - Ejemplos

Requerimientos	Criteria de Aceptación
Registrar Oferta Laboral	Permite que una empresa pueda registrar una oferta laboral basándose en las capacidades y competencias que maneja la carrera de computación.
Toma de tests por alumnos de psicología a personas de otras instituciones	Permite convocar a un alumno que, luego de ser evaluado por la empresa, cumple con todos los requisitos que la oferta laboral establece.

# Planificación de la gestión de requisitos

- Establece el detalle del nivel de gestión requisitos que es requerido.
- Decisiones de gestión de requisitos:
  - **Identificación de requisitos**— Cada requisito debe ser identificado de modo que pueda hacerse una referencia cruzada con otros.
  - **Proceso de gestión de cambios**— Es el conjunto de actividades que evalúan el impacto y el costo de los cambios.
  - **Políticas de trazabilidad**— Estas políticas definen cómo registrar las relaciones entre los requisitos y el sistema diseñado.
  - **Soporte de herramientas**— Las herramientas que utilizarán.

# Gestión de cambios en los requisitos (I)

- Es el proceso de gestionar los cambios de los requisitos.
- Es común que nuevos requisitos surjan cuando un sistema está siendo desarrollado y después de haber entrado en uso.
  - El entorno empresarial y técnico del sistema siempre cambia después de la instalación.
  - Generalmente los grandes sistemas tienen una diversa comunidad de usuarios, algunos de los cuales tienen diferentes requisitos y pueden existir conflictos en las prioridades.
- Es necesario mantener un rastreo individual de los requisitos y mantener sus referencias a otros requisitos de forma de facilitar el análisis de impacto ante posibles cambios.
- Se recomienda establecer un proceso formal de control de cambios.

# Gestión de cambios en los requisitos (2)

- Decidir si un cambio en los requisitos debe ser aceptado.
- Problemas de análisis y cambios en la especificación
- El problema o cambio propuesto es analizado para chequear si es válido.
- Análisis de impacto
- El efecto del cambio propuesto se evalúa a través de la información de trazabilidad y el conocimiento general de los requisitos del sistema. Luego se toma la decisión de si se debe proceder o no con el cambio.
- Implementación del cambio
- Se modifica la documentación impactada por el cambio (especificación de requisitos, diseño del sistema) y la aplicación.
- Idealmente, el documento debe estar organizado para que los cambios puedan ser fáciles de implementar.

Próxima clase:  
Ingeniería de requisitos  
(continuación)