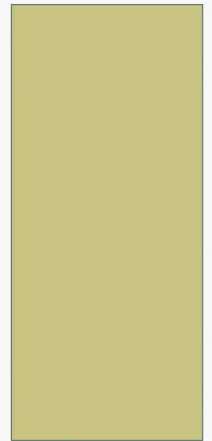


INGENIERÍA DEL SOFTWARE

UNIDAD 6: INGENIERÍA DE REQUERIMIENTOS
CICLO LECTIVO 2013



OBJETIVOS DE LA CLASE

- **Introducción.**
- **Conceptos generales.**
- **Identificación de requerimientos.**
- **Tipos de Requerimientos.**
- **Procesos de la IR.**
- **Técnicas y herramientas.**
- **Aspectos formales**

TEMAS PARA LAS PRÓXIMAS CLASES

- Chaos Report, quienes lo hacen, historia. Estadísticas de los últimos años.
- SRS IEEE 830. (Especificación de requerimientos de Software). Origen de la IEEE.

INTRODUCCIÓN A LA INGENIERÍA DE REQUERIMIENTOS

La Ingeniería de Requerimientos (IR) se relaciona con la búsqueda de la situación futura y el cambio asociado.

- ☐ **Está relacionada con encontrar (capturar) información y considerar posibles opciones, y con la identificación de lo que debería ser diseñado en orden a satisfacer una necesidad futura percibida.**

INGENIERÍA DE REQUERIMIENTOS

“La parte más difícil de construir un sistema de software es decidir precisamente qué construir.

Ninguna otra parte del trabajo conceptual **es tan** **difícil** como establecer los requerimientos técnicos detallados, incluyendo todas las interfases hacia las personas, máquinas y otros sistemas de software.

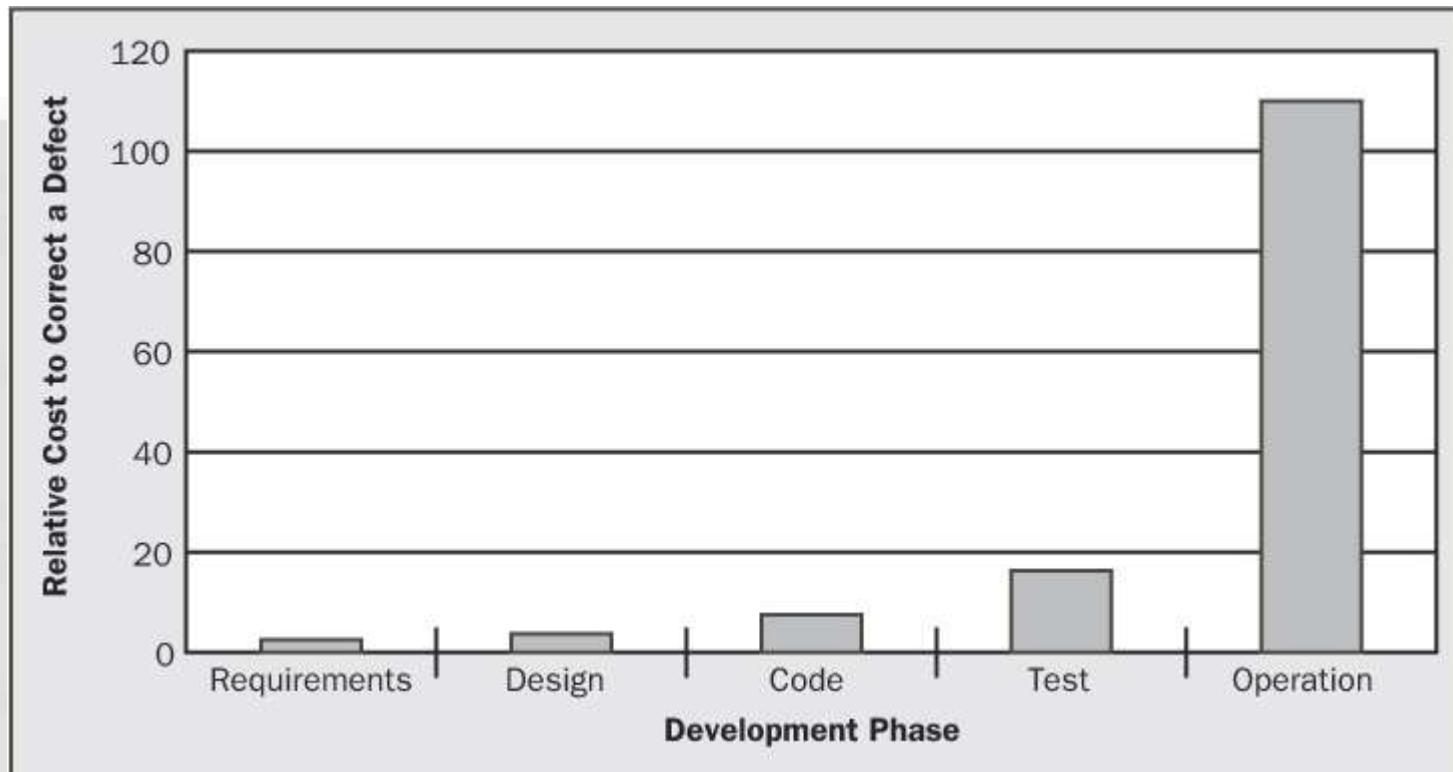
Ninguna otra parte del trabajo paraliza el sistema si es mal hecha. Ninguna otra parte es **tan difícil de rectificar después**”

INGENIERÍA DE REQUERIMIENTOS

Las 3 ideas fundamentales a considerar:

- ☐ La parte más difícil del trabajo: al inicio.
- ☐ El impacto perjudicial del error.
- ☐ La dificultad de rectificar posteriormente.

COSTO DE REPARAR ERRORES



- ❑ El costo de reparar un defecto de software varía de acuerdo al momento en que se esté dentro del ciclo de vida del software.

PROBLEMAS DEL DISEÑO DE SOFTWARE

Entre el 40 y el 60 % de los defectos encontrados pertenecen a la etapa de requerimientos.

Esencialmente la diferencia es “entre lo que los desarrolladores piensan que tienen que construir y lo que los usuarios o clientes piensan que van a obtener”.

PROBLEMÁTICA DE LA IR

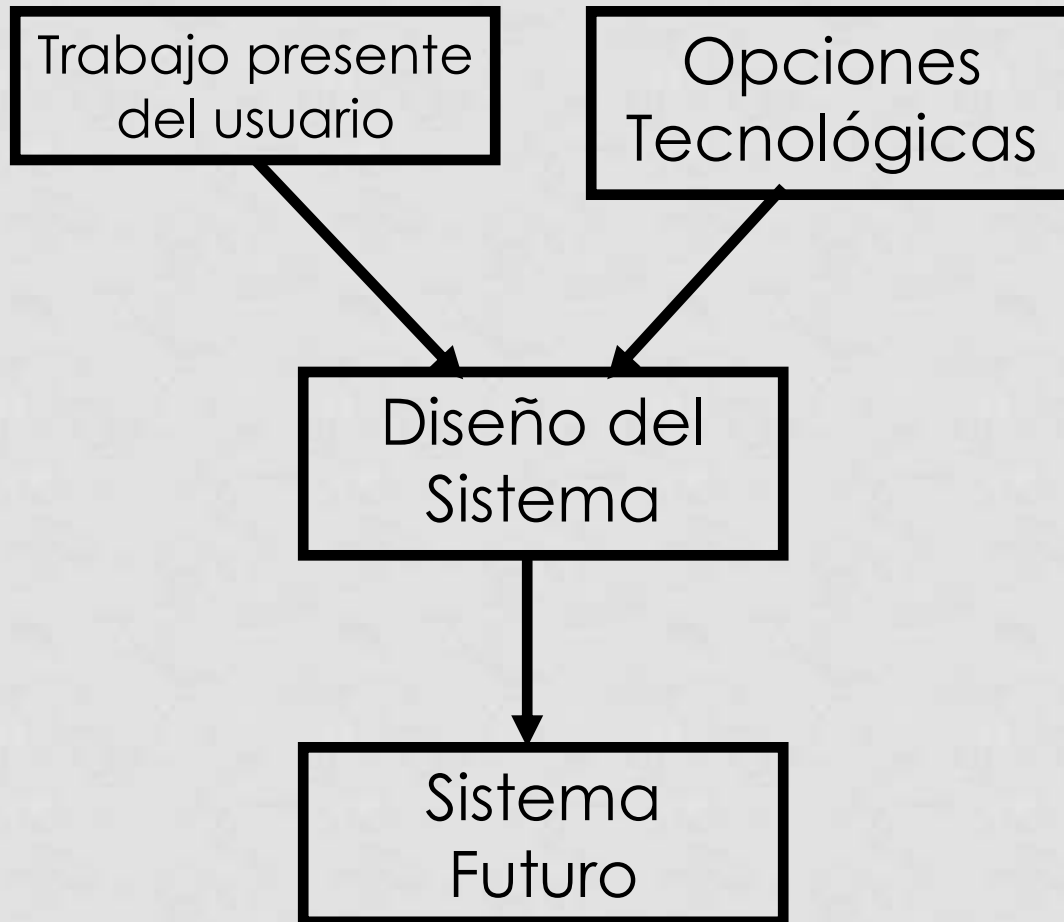
- El desarrollo de software es una cadena de valor.
- Los defectos inyectados acarrearán parte de ese valor.
- Existe la gran falacia de la velocidad:
 - Cuanto menos tarde..... ¡¡Mas tarde!!
- Progresar sobre requerimientos no consolidados es la principal razón de problemas en el desarrollo.

GESTIÓN DE REQUERIMIENTOS

1995, Chaos Report.

- Solo 16% de todos los proyectos se completaron en calendario y presupuesto original.
- 8 Problemas concentran 80% de las fallas (Pareto).
 1. Requerimientos Incompletos (13.1%)
 2. Falta de involucración del Cliente (12.4%)
 3. Falta de recursos (10.6%)
 4. Expectativas poco realistas (9.9%)
 5. Falta de soporte ejecutivo (9.3%)
 6. Requerimientos cambiantes/volátiles (8.7%)
 7. Falta de planeamiento/Planeamiento realista (7.5%).
 8. Carencia/Ausencia de Necesidad (7.5%).

ORIENTACIÓN DE LA IR



IR:
relacionada
a encontrar
la situación
futura y el
cambio
asociado

t ↓

INGENIERÍA DE REQUERIMIENTOS

- La Ingeniería en requerimientos entra como una **subtarea** de la Ingeniería de Software; propone métodos, técnicas y herramientas que faciliten el trabajo de Definición de lo que se quiere de un software.

Ingeniería de requerimientos:
Rol fundamental



- ♣ Definición del Problema
- ♣ Estudio de Factibilidad
- ♣ Análisis
- ♣ Diseño del Sistema
- ♣ Diseño Detallado
- ♣ Implementación
- ♣ Mantenimiento

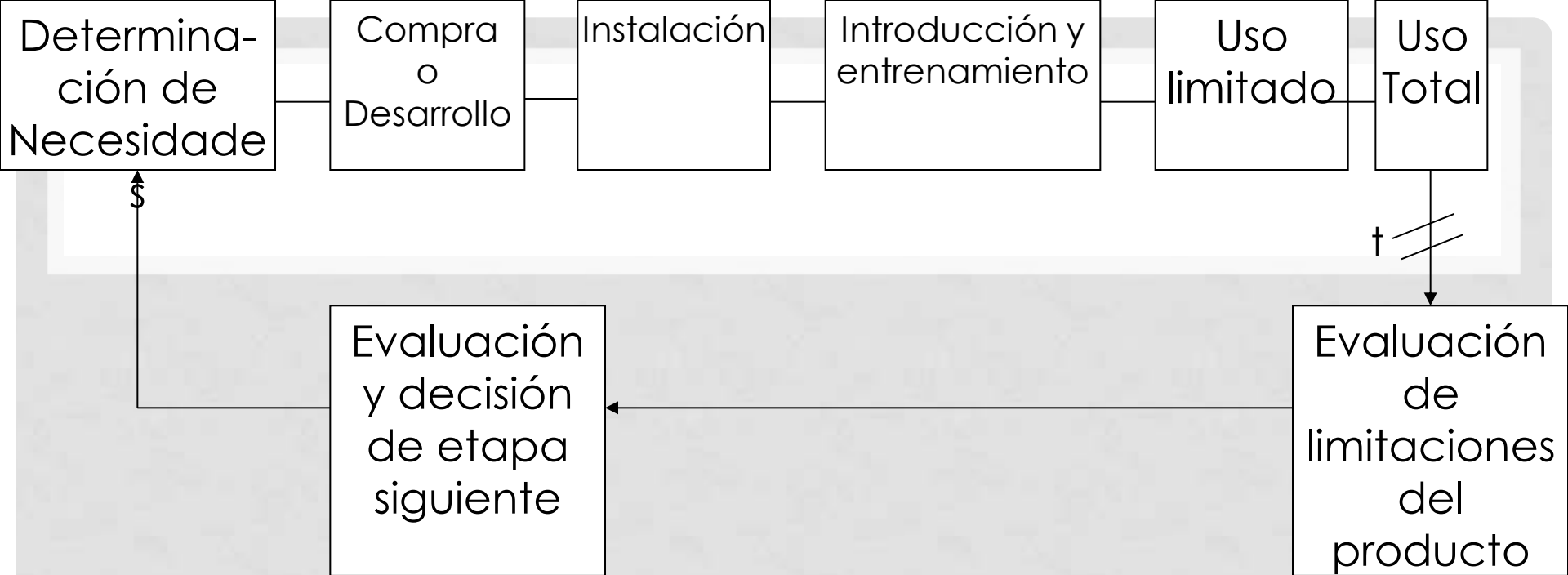
Debido a esta relación se deriva que:

Muy fuerte
interacción

Ing. en requerimientos



Demandantes del Soft



¿Por qué es necesaria una etapa de requerimientos?

Desde el punto de vista del cliente una etapa de requerimientos es necesaria porque le ayuda a entender y expresar las nuevas necesidades y a identificar cómo puede satisfacerlas.

NECESIDAD DE REQUERIMIENTOS

REQUERIMIENTO

- ❑ Simplemente... cualquier cosa que un cliente necesite.
- ❑ Desde el punto de vista del diseñador, cualquier cosa que necesite ser diseñada.

REQUERIMIENTOS

- '94 Jones: “Los requerimientos son las sentencias de necesidades de un usuario que lanzan el desarrollo de un programa o sistema”
- '93 Alan Davis: “una necesidad de un usuario o una facilidad necesaria, función o atributo de un sistema que puede ser sentido desde una posición externa al sistema”
- En general, el énfasis está en:

Lo que irá en el producto, sus características. No cómo el producto se diseñará o construirá.

REQUERIMIENTOS

- ❑ '97 Sommerville and Sawyer: “Requerimientos son... una especificación de lo que debería ser implementado. Son una descripción de cómo el sistema debería comportarse o de una propiedad o atributo del sistema. Puede ser una restricción sobre el proceso de desarrollo del sistema”

Lo que realmente necesitamos es asegurarnos que todos los stakeholders del proyecto lleguen a un entendimiento compartido de los términos usados para describir los requerimientos.

- ❑ Los errores en los requisitos se detectan tarde.

REQUERIMIENTOS

La más notable de las definiciones es la de la **IEEE**:

1. Una condición o capacidad necesaria para un usuario para resolver un problema o alcanzar un objetivo.
2. Una condición o capacidad que debe ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, especificación, u otro documento formalmente impuesto.
3. Una representación documentada de una condición o capacidad dada en 1 o 2.

CLASIFICACIÓN DE LOS REQUERIMIENTOS

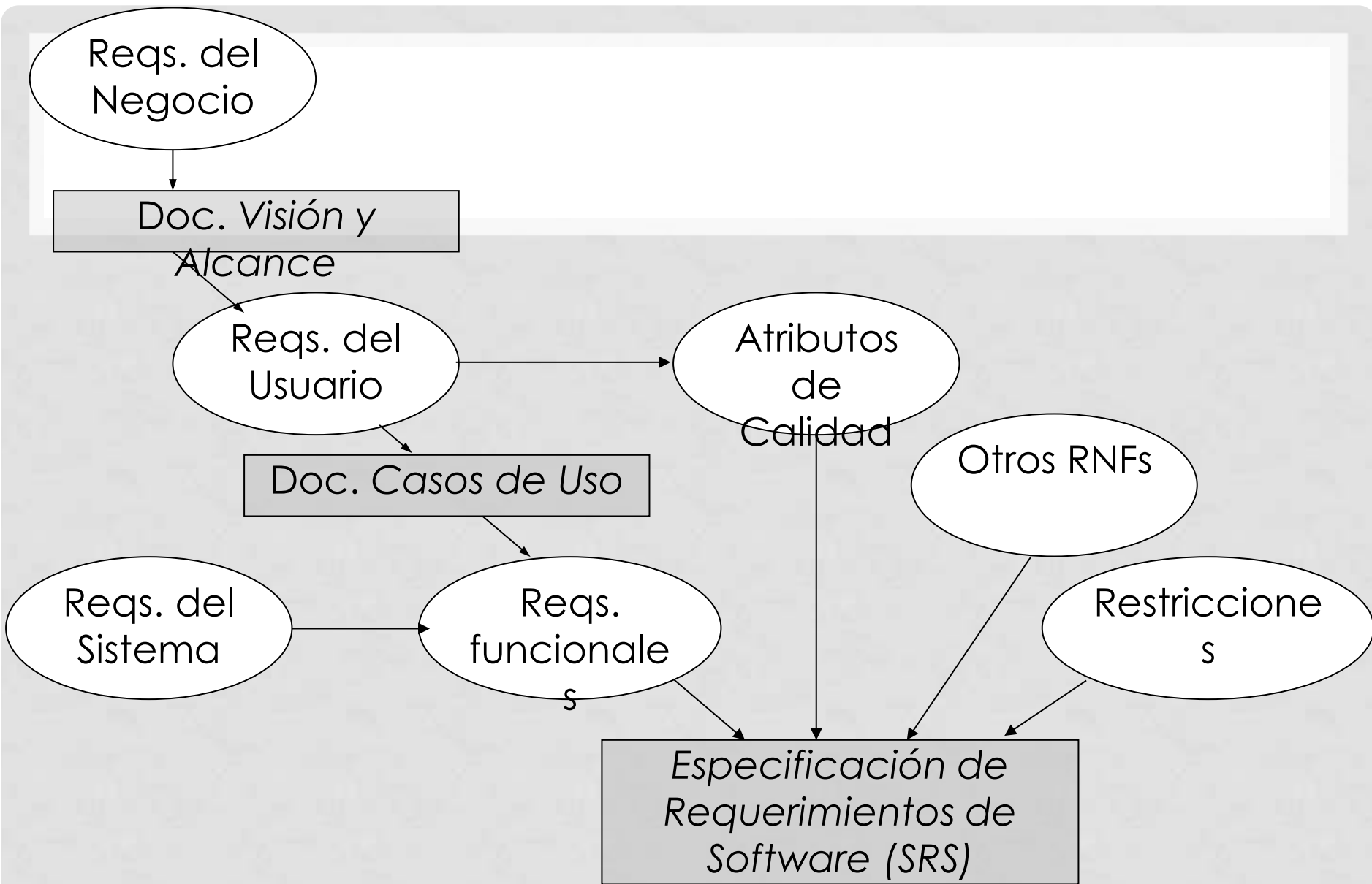
En cuanto al contenido:

- ☐ *Funcionales*: describen las entradas, salidas y funciones de transformación del Sistema;
- ☐ *No Funcionales*: definen atributos como confiabilidad, performance, entre otros;
- ☐ *Inversos*: definen cómo el Sistema de Software nunca se debe comportar.

CLASIFICACIÓN DE LOS REQUERIMIENTOS

En cuanto al origen:

- ☐ **Del negocio:** objetivos de alto nivel de la organización. Documentados en la visión o alcance del proyecto.
- ☐ **Del usuario:** tareas que los usuarios deben hacer con el producto. Documentados en los casos de uso o escenarios.
- ☐ **Funcionales:** las funcionalidades del soft a construir.
- ☐ **No Funcionales:** que responden a estándares, regulaciones, contratos, interfases, performance, restricciones y atributos de calidad.



REQUERIMIENTOS Y CALIDAD

- Los requerimientos están indisolublemente vinculados con la calidad del sistema.
- ¿Calidad?
- Calidad según IEEE:
 - El grado en que un sistema, componente o proceso satisface los requerimientos especificados.
 - El grado en el que un sistema, componente o proceso satisface necesidades y expectativas de un cliente o usuario.

BENEFICIOS

Los beneficios de **buenos requerimientos** incluyen:

- ❑ Acuerdo entre los stakeholders (en relación al comportamiento esperado y al criterio de aceptación).
- ❑ Una base para estimaciones de recursos y tiempos.
- ❑ Mejoras en la usabilidad, mantenibilidad, etc.
- ❑ Reducción del esfuerzo: menos retrabajo y omisiones.

Las ventajas crecen con el tamaño y complejidad del sistema

ROL DE LOS REQUERIMIENTOS

- Acuerdo desarrolladores-clientes-usuarios finales.
 - Aspecto contractual.
 - Multipartes (comunicación – conflictos – cambios de visiones)
- Base para el diseño de software.
 - Minimizar defectos tanto como sea posible.
 - Técnicamente factible.
- Soporte para verificación y validación.
- Soporte a la evolución del sistema.
 - Evolución del sistema (cambio del sistema viejo al sistema nuevo y cambio de requerimientos).

INGENIERÍA DE REQUERIMIENTOS

- La IR se define como el proceso sistemático de desarrollo de los requerimientos a través de un proceso cooperativo e iterativo de análisis del problema, documentando las observaciones resultantes en una variedad de representaciones y chequeando la certeza del conocimiento ganado.

IR – PUNTOS A TENER EN CUENTA

- ❑ Es una parte integral del proceso conectado a las otras partes por continuas retroalimentaciones.
- ❑ La naturaleza de los requerimientos es tal que los cambios ocurrirán necesariamente.
- ❑ Con las especificaciones de los requerimientos, el ciclo de desarrollo del software se hará con estos en mente.

IR – PUNTOS A TENER EN CUENTA

- ❑ Hay varias fuentes de requerimientos que deben ser exploradas.
- ❑ Habrá una necesidad continua de relacionar los requerimientos con las fuentes; también de relacionar el diseño y las decisiones de implementación a los requerimientos.
- ❑ Siempre habrá requerimientos conflictivos y situaciones decisivas que deberán acomodarse.

ALCANCE DE LA IR

- ❑ La IR tiene que ver con:
 - ❑ Los procesos que concluyen en la formulación de requerimientos.
 - ❑ La generación de productos relacionados a esos procesos.
 - ❑ La administración de los requerimientos en el desarrollo, operación y mantenimiento de software

IR: EL ROL DE LAS TÉCNICAS Y LAS FALLAS

- La mayoría de los problemas en el desarrollo de sistemas tienen su raíz no tanto en cuestiones técnicas (software) sino en cuestiones sociales, económicas, organizacionales y gerenciales (Bubenko, 1995).

IR: EL ROL DE LAS TÉCNICAS Y LAS FALLAS

- La falla o éxito de un proyecto siempre será definido con relación a un grupo particular con sus objetivos, intereses, roles y expectativas...
- Esos intereses y objetivos están definidos en el contexto global de una organización y su medioambiente político y social y no sólo con relación a la tecnología.

TIPOS DE FALLAS EN EL DESARROLLO DE SOFTWARE

- **Fallas de correspondencia**, cuando los objetivos del proyecto no han sido alcanzados.
- **Fallas de procesos**, que se relacionan a los procesos involucrados en el desarrollo de sistemas de información en los que el presupuesto, tiempo u otro recurso han sobrepasado el punto en donde se esperaba algún beneficio respecto del sistema propuesto, o donde los recursos gastados no resultaron en un sistema implementable.

TIPOS DE FALLAS EN EL DESARROLLO DE SOFTWARE

- **Fallas de interacción:** es el argumento para el cual un bajo nivel de uso de un sistema puede ser interpretado como una falla.
- **Fallas de expectativas:** son simplemente que el sistema ha fallado en dar con las expectativas de al menos un grupo de interés sobre el sistema.

Causa posible					
Tipo de Falla	1. Falta de proceso sistemático	2. Comunicación pobre entre la gente	3. Falta de conocimiento o apropiado o comprensión compartida	4. Documentación inapropiada, incompleta o precisa	5. Gestión pobre de gente o recursos
Del proceso Beneficio negativo, recursos mal estimados	✓	✓			✓
De interacción Bajo nivel de uso del sistema		✓	✓	✓	
De expectativas No alcanzó las expectativas de algún stakeholder		✓	✓	✓	

➔ **La gestión de requerimientos, como proceso, debe minimizar la ocurrencia de fallas**

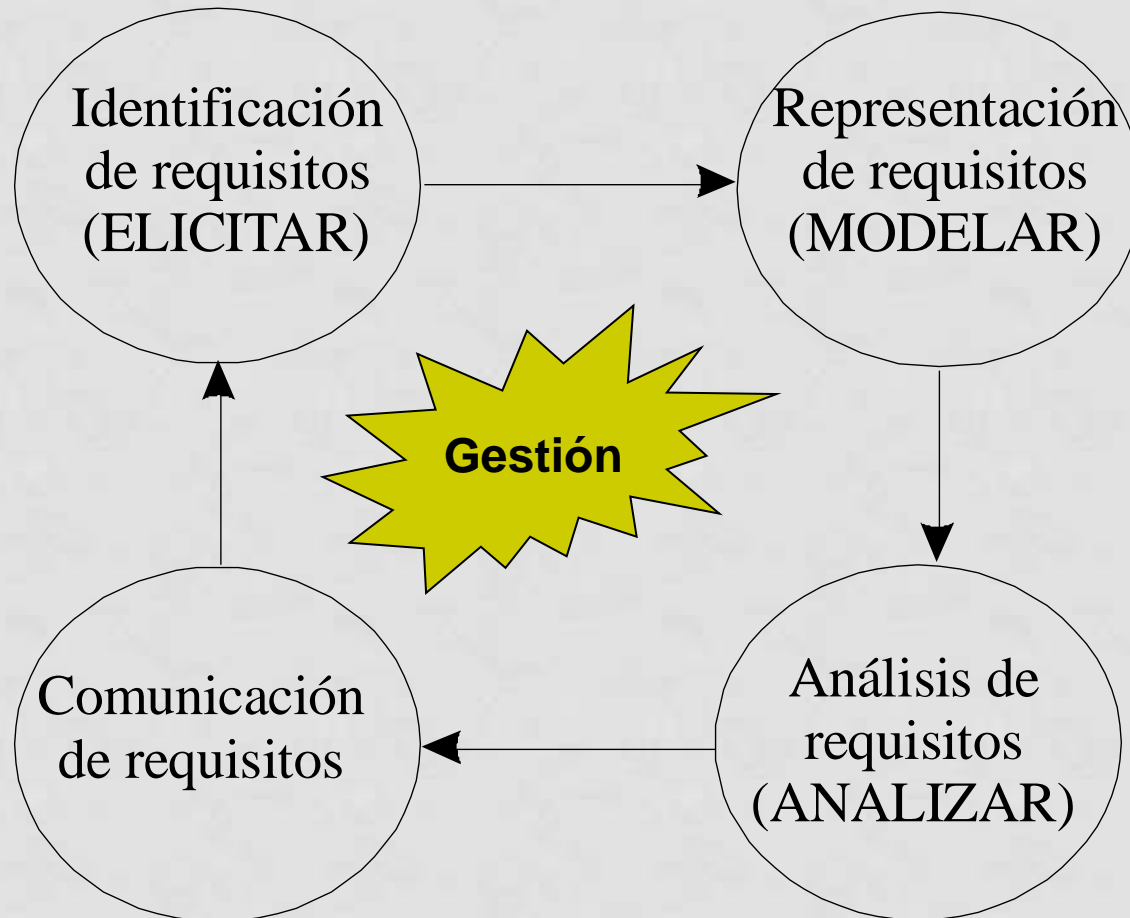
IR COMO PROCESO

El proceso de IR: conjunto de actividades que resultan en el desarrollo de productos intermedios o finales.

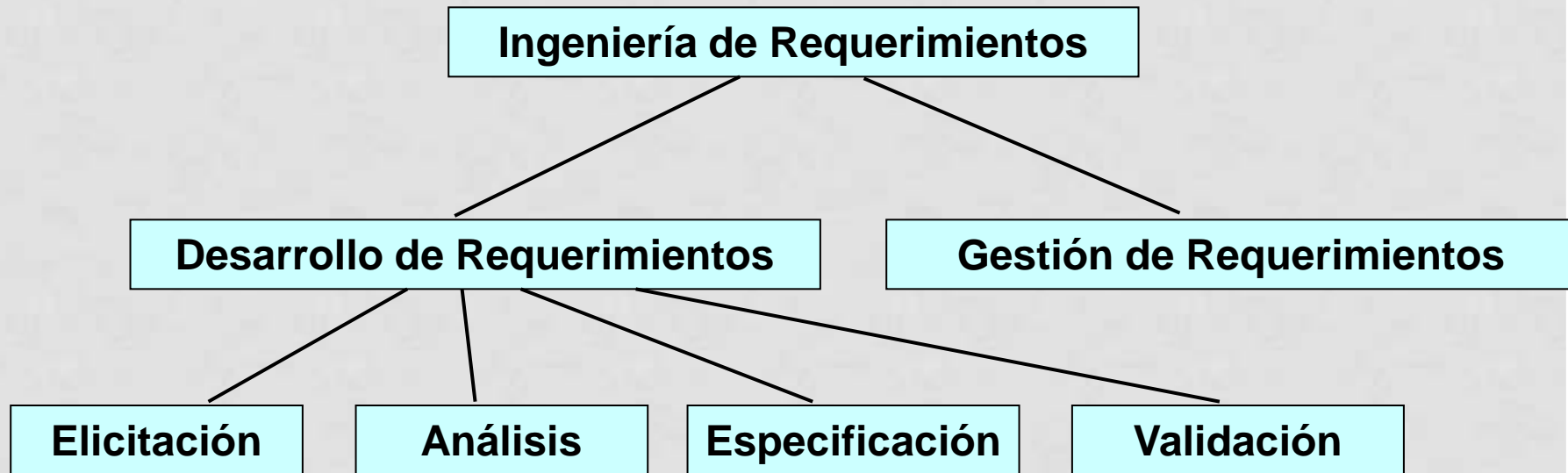
Objetivo: estandarizar el proceso y los productos para:

- ❑ permitir el seguimiento y control de cada actividad.
- ❑ medir la efectividad del procedimiento y formas de mejorarlo a través de la evaluación de los productos parciales.
- ❑ posibilitar el uso de herramientas automáticas → aumentar la productividad y reducir los costos administrativos.

PROCESO DE LA IR



IR: SUBCOMPONENTES



Wiegiers, Karl E. (2003). Software Requirements. 2nd Edition. Microsoft Press.

IR: TAREAS

Desarrollo				Gestión
Elicitación	Modelado	Análisis	Comunicación	
Identificación de Fuentes Inform.	Representación	Verificación	Documento de Requerimientos	Identificación de cambios
Recolección de hechos	Organización	Validación	Técnicas de comunicación	Análisis de cambios
Comunicación	Almacenamiento (registración)	Negociación		Realización de cambios

DOCUMENTO DE REQUERIMIENTOS

- Un Documento de Requerimientos (DR) es una especificación de lo que se requiere que haga un sistema de información y no de cómo hacerlo.
- Un DR puede ser evaluado por su efectividad como un medio de comunicación, por su contenido como una medida de chequeo (checklist), y por la calidad de su contenido.

IEEE: GUÍA DE ESPECIFICACIÓN DE REQUERIMIENTOS

- *Funcionalidad:* Qué se supone que hará el software?
- *Interfaces Externas:* Cómo interactúa el software con las personas, hardware y otras aplicaciones?
- *Performance:*Cuál es la velocidad, disponibilidad, tiempo de respuesta, tiempo de recuperación, etc., de las funciones del software?

IEEE: GUÍA DE ESPECIFICACIÓN DE REQUERIMIENTOS

- *Atributos:* Cuáles son las consideraciones sobre portabilidad, correctitud, mantenibilidad, seguridad, etc.?
- *Restricciones:* Existe algún estándar vigente, lenguaje de implementación, políticas de integridad de BD, limitaciones de recursos, entornos de operación, etc.?

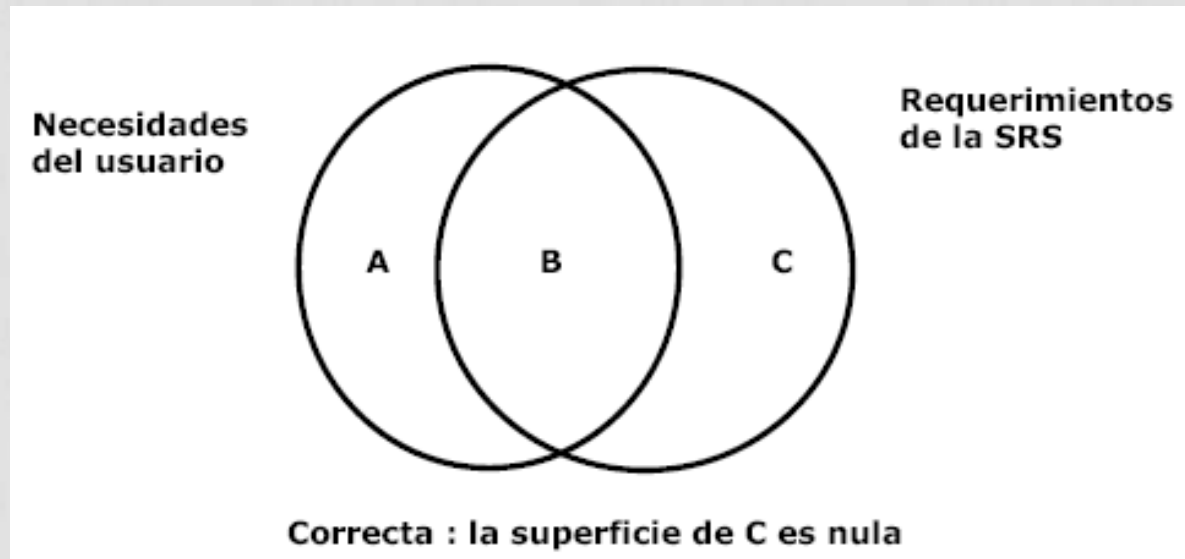
CARACTERÍSTICAS DE UNA BUENA DEFINICIÓN DE REQUERIMIENTOS

Una especificación de requerimientos de software (ERS/SRS) debe ser:

- ☐ Correcta: ?
- ☐ No ambigua: ?
- ☐ Completa: ?
- ☐ Consistente: ?
- ☐ Ranqueable (en importancia y/o estabilidad): ?
- ☐ Verificable: ?
- ☐ Modificable: ?
- ☐ Trazable: ?
- ☐ Usable: ?

CARACTERÍSTICAS DE UNA BUENA DEFINICIÓN DE REQUERIMIENTOS

- **Correcta:**
 - Si y sólo sí todo requerimiento formulado en la ERS debe ser satisfecho por el software.
 - Puede ser correcta pero no abarcar todas las necesidades planteadas.



CARACTERÍSTICAS DE UNA ERS

- **No ambigua:** interpretación única.
 - los requerimientos, comúnmente, están escritos en lenguaje natural, donde las oraciones pueden tener más de un significado.
 - los lenguajes de requerimientos formales ayudan a reducir la ambigüedad, porque los procesadores de lenguaje detectan automáticamente cualquier error semántico, de léxico o sintáctico.

CARACTERÍSTICAS DE UNA ERS

- **Completa:** La ERS es completa si incluye:
 - Todos los requerimientos significativos, donde se relacionan la funcionalidad, performance, restricciones de diseño, atributos o interfaces externas y los estándares de la compañía.
 - Todas las respuestas del software a todas las entradas posibles.
 - Todas las etiquetas y referencias a las figuras, tablas y diagramas de la ERS y las **definiciones de los términos**.
- ➔ **Principio de redacción:** el software hará sólo lo que dice la ERS.

CARACTERÍSTICAS DE UNA ERS

☐ **Consistente:** pueden ocurrir tres tipos de conflictos:

- ☐ **Diferentes términos usados para el mismo objeto:** por ejemplo, 'un P45' y 'un formulario de impuesto' puede ser usado para describir el mismo formulario.
- ☐ **Características de objetos conflictivas:** por ejemplo, en alguna parte del DR se puede exigir que una luz roja indique un error, y en otra que el error sea indicado por una luz azul.
- ☐ **Fallas temporales o lógicas:** por ejemplo, 'A sigue a B' en una parte; y en otra 'A y B ocurren simultáneamente'.

CARACTERÍSTICAS DE UNA ERS

- **Ranqueable:** Cada requerimiento tiene una identificación que indica su importancia y/o estabilidad. Esto ayuda a que:
 - Los clientes consideren mejor los requerimientos.
 - Los desarrolladores toman mejores consideraciones de diseño y asignan esfuerzos más adecuados.

CARACTERÍSTICAS DE UNA ERS

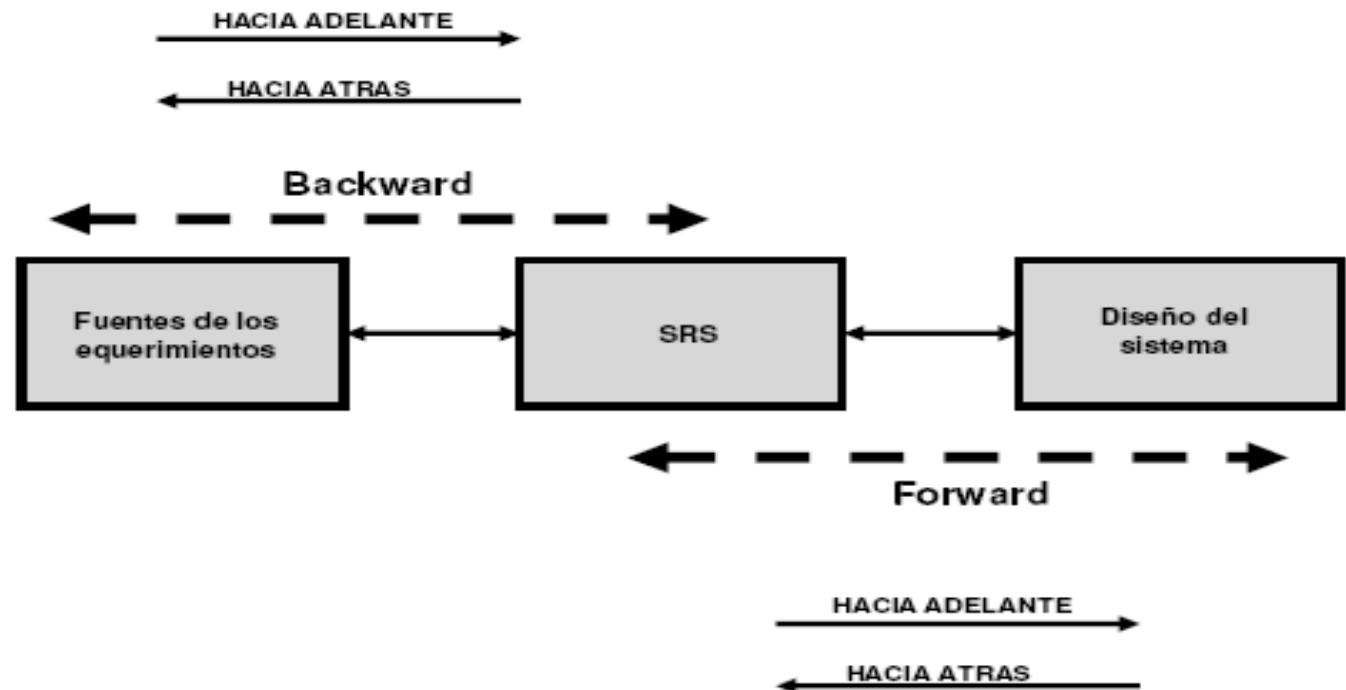
- **Verificable:** Una ERS es verificable si solo si todo requerimiento es verificable.
 - Debe existir un proceso con el que un humano o máquina pueda chequear que el producto de software cumple con el requerimiento.
 - “el producto deberá tener una buena interfase con el hombre”, “el programa debe funcionar bien”, “usualmente debería...”
 - “el sistema responderá a la consulta de un usuario dentro de los 20 segundos después que el usuario presione la tecla enter, el 80% de las veces”.
- ➔ Los requerimientos ambiguos no son verificables

CARACTERÍSTICAS DE UNA ERS

- **Modificable:** La ERS deberá estar organizada en forma coherente y fácil de usar, con una tabla de contenidos, un índice y referencias cruzadas explícitas. Las sentencias de requerimientos no deben ser redundantes, en lo posible.
 - Cualquier cambio debería poder hacerse
 - Fácilmente
 - Manteniendo la completitud.
 - Consistentemente.
 - Requiere:
 - Organización coherente y fácil de usar,
 - No ser redundante.

CARACTERÍSTICAS DE UNA ERS

- **Trazable:** el origen de cada requerimiento debe ser claro, facilitando el rastreo de decisiones anteriores, y la búsqueda de todos los documentos que se referencian en la ERS.



CARACTERÍSTICAS DE UNA ERS

- **Usable:** La ERS debe ser diseñada de modo que sea una referencia y si es necesario se modifique durante la vida del producto.
- Es usado en las fases de operación y mantenimiento.
- Tabla de contenido, índice y apéndice.

BENEFICIOS DE UNA BUENA ERS

- ❑ Según la IEEE (1998), una buena ERS debe brindar diversos beneficios al proceso:
 - ❑ Establecer la base de acuerdo entre los clientes y los proveedores sobre las características del producto de software.
 - ❑ Reducir el esfuerzo de desarrollo.
 - ❑ Provee una base para estimar costos y cronogramas.
 - ❑ Provee una línea base para la validación y verificación.
 - ❑ Facilita la transferencia (nuevos usuarios, máquinas, etc.)
 - ❑ Sirve como base para la mejora posterior del producto.

BIBLIOGRAFÍA OBLIGATORIA

- ❑ [Sommerville, 2011] Sommerville, Ian. “Ingeniería de Software”. **Capítulo 4**. Novena Edición. Pearson Educación, 2011.
- ❑ [Macaulay, 2006] Macaulay, Linda A. “Requirements Engineering”. Springer Verlag.
- ❑ [Wiegers, 2003] Wiegers, Karl E. “Software Requirements”. 2nd Edition. Microsoft Press.

¿Dudas, consultas?

