

# Ingeniería de Software



## Ingeniería de Requisitos Parte 1

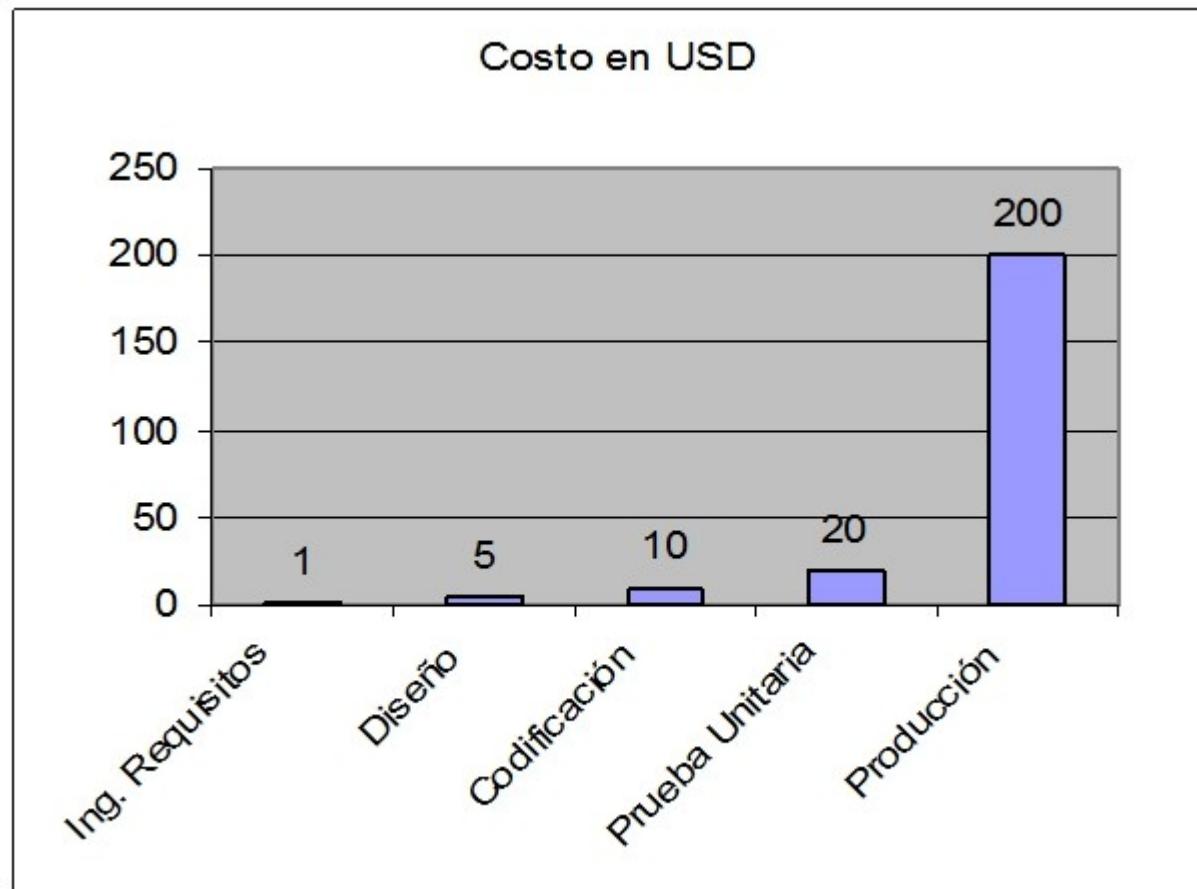
# Requisitos

---

- Los requisitos de un sistema son descripciones de lo qué debería hacer el sistema— los servicios que provee y las restricciones en su operación.
- El proceso de descubrir, analizar, documentar y chequear esos servicios y restricciones es llamado ingeniería de requisitos (RE).

# Costos de errores en los requisitos

- Costo de corregir un error en los requisitos  
(Boehm-Papaccio, 1988)



# Requisitos - Abstracción

---

- La forma de un requisito puede variar desde una oración con un alto nivel de abstracción hasta una especificación funcional matemática.
- Posibles contextos de un requisito:
  - Puede ser la base de una oferta para un contrato— por lo tanto debe estar abierto a la interpretación.
  - Puede ser la base para un contrato— por lo tanto debe ser definido detalladamente.

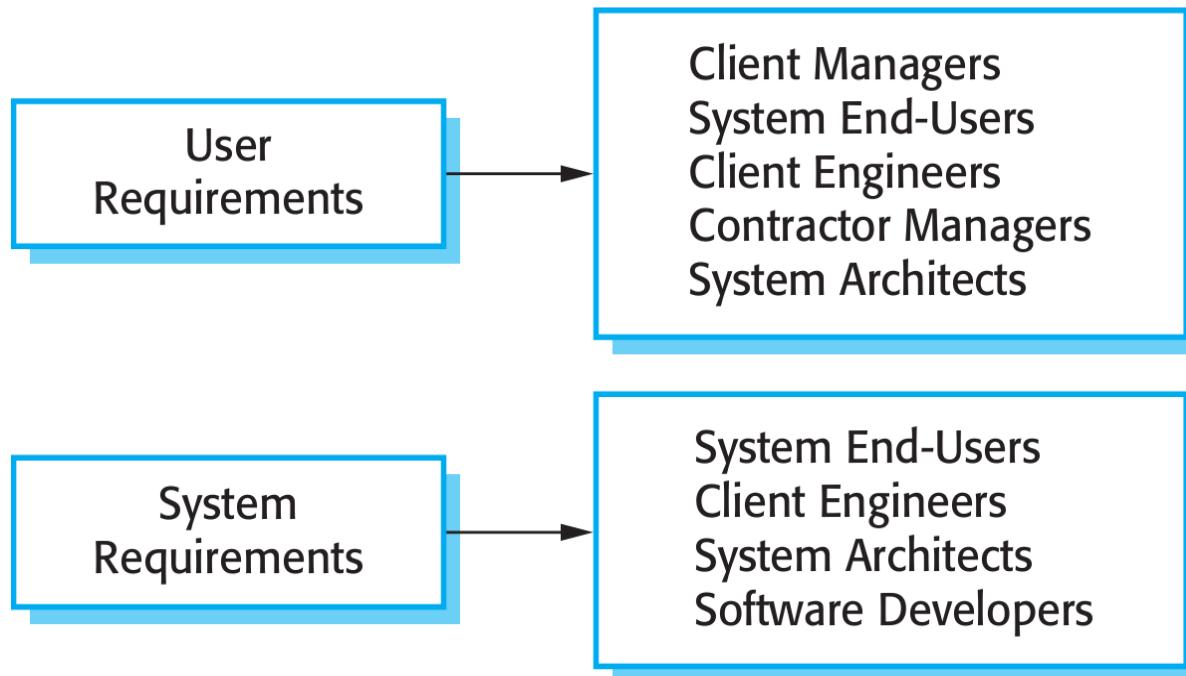
# Tipos de requisitos - Abstracción

---

- **Requisitos de usuario**
  - Son declaraciones en lenguaje natural, además de diagramas de los servicios que proporciona el sistema y sus limitaciones operativas. Escritos para los clientes.
- **Requisitos del sistema**
  - Un documento estructurado que establece descripciones detalladas de las funciones, servicios y restricciones operacionales del sistema. Define lo que se debe implementar, podría ser parte de un contrato entre el cliente y el empresario.

# Tipos de requisitos - Abstracción

- Es necesario escribir los requisitos utilizando diferentes niveles de abstracción porque diferentes lectores los usan de diferente forma.



# Otros tipos de *información de requisitos*

---

- **Requisitos del dominio**
  - Se derivan del dominio de aplicación del sistema y no de las necesidades específicas de los usuarios.
- **Requisitos del negocio**
  - Un objetivo de alto nivel de una organización que desarrolla un producto o de un cliente que lo compra.
- **Regla de negocio**
  - Una política, guía, estándar o regulación que define o restringe algún aspecto del negocio.
- **Requisito de interfaz externa**
  - Una descripción de una conexión entre un sistema de software y un usuario, otro sistema de software o un dispositivo de hardware.

# Otros tipos de *información de requisitos*

---

- **Característica (feature)**
  - Una o más capacidades relacionadas de forma lógicas que proveen valor al usuario y son descriptas como un conjunto de requisitos funcionales. Ejemplo: los favoritos del navegador.
- **Requisito funcional**
  - Una descripción de lo que el sistema debe hacer bajo condiciones específicas.
- **Requisitos no funcionales**
  - Una descripción de una propiedad o característica que un sistema debe poseer o una restricción que debe respetar.
- **Atributo de calidad**
  - Un tipo de requisito no funcional que describe una característica de servicio o desempeño de un producto.

# Requisitos funcionales y no funcionales

- **Requisitos funcionales**

- Son declaraciones de los servicios que el sistema debería proveer, cómo el sistema debería reaccionar a entradas particulares y cómo debería comportarse en situaciones particulares.
- Pueden indicar lo que el sistema no debería hacer.

- **Requisitos no funcionales**

- Son restricciones a los servicios o funciones provistas por el sistema, como restricciones de tiempo, restricciones sobre el proceso de desarrollo, estándares, etc.
- Generalmente son aplicables al sistema entero y no a servicios o funciones en particular.

# Requisitos funcionales

---

- Describen funcionalidades o servicios del sistema.
- Dependen del tipo de software, de los usuarios y del tipo de sistema donde el software es usado.
- Ejemplos:
  - Un usuario podrá buscar en las listas de citas de todas las clínicas.
  - El sistema deberá generar cada día para cada clínica una lista de pacientes que se espera asistan a la cita durante ese día.

# Falta de precisión en los requisitos

---

- La falta de precisión en los requisitos es la causa de muchos problemas en la ingeniería de software.
- Los requisitos ambiguos pueden ser interpretados de diferentes maneras por desarrolladores y usuarios.
- Ejemplo:
  - Un usuario podrá buscar en las listas de citas de todas las clínicas.
    - Búsqueda dentro de todas las clínicas a la vez
    - Búsqueda dentro de cada clínica por vez
  - Búsqueda por nombre y apellido
    - Búsqueda por hora de la cita

# Completitud y consistencia de los requisitos

---

- Los requisitos deben ser lo más completos y consistentes posibles.
- **Completos**
  - Todos los requisitos requeridos por el usuario están definidos.
- **Consistentes**
  - En los requisitos no hay conflictos o definiciones contradictorias.

# Requisitos no funcionales

---

- Definen las propiedades y las restricciones del sistema.
- Ejemplos:
  - Propiedades— confiabilidad, tiempo de respuesta y requisitos de almacenamiento.
  - Restricciones— capacidad de dispositivos de entrada/salida, las representaciones del sistema, etc.
- Los requisitos no funcionales pueden ser más críticos que los requisitos funcionales. Si no se cumplen el sistema puede resultar inútil.

# Implementación de los requisitos no funcionales

---

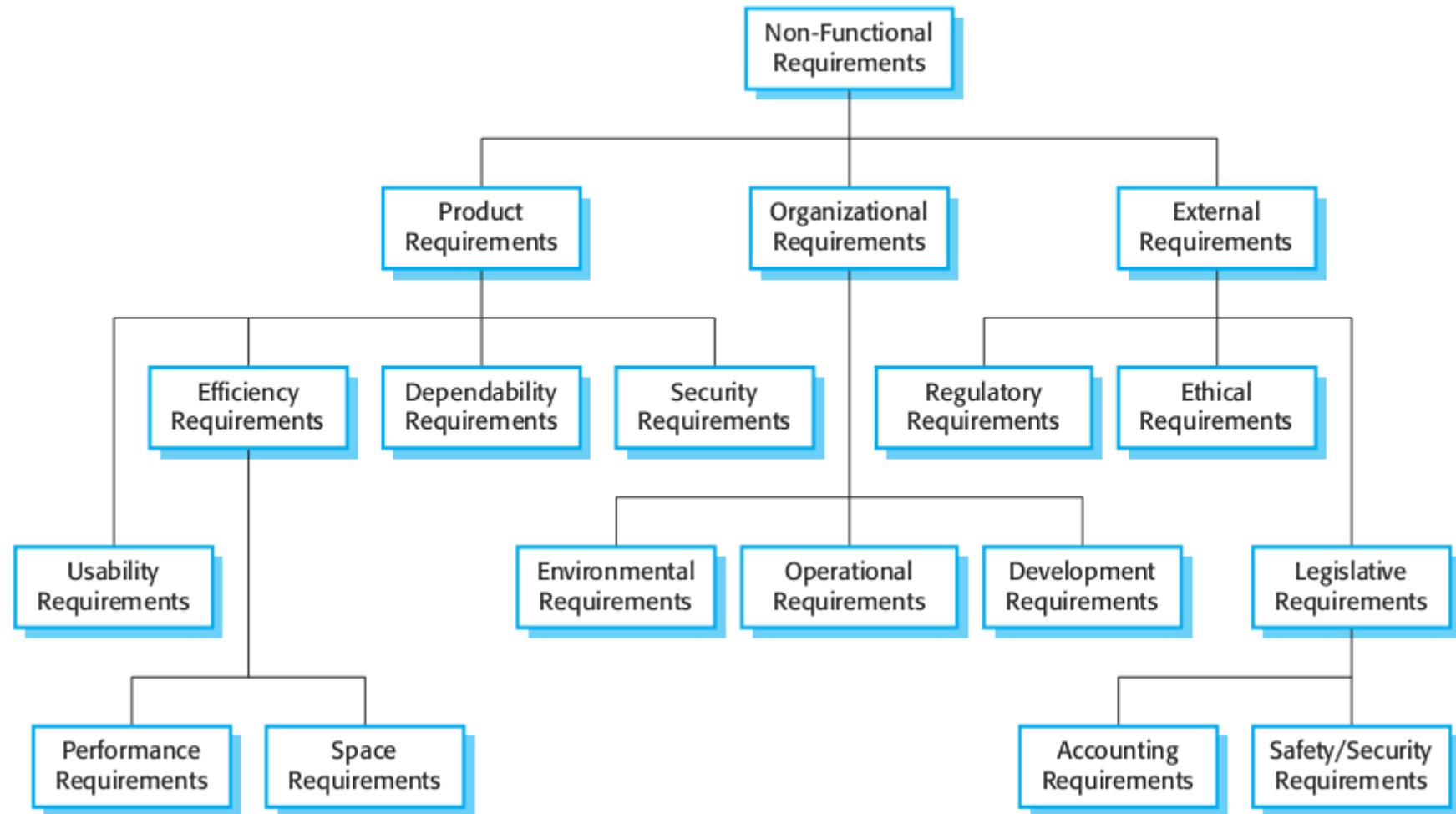
- En general es posible identificar los componentes que implementan un requisito funcional específico. Es mucho más difícil identificar los componentes que implementan un requisito no funcional.
- Esto se debe a que:
  - Los requisitos no funcionales pueden afectar a la arquitectura general del sistema en lugar de a componentes individuales.
  - Un único requisito no funcional, por ejemplo un requisito de seguridad, puede generar varios requisitos funcionales que definen servicios requeridos para el sistema.

# Tipos de requisitos no funcionales

---

- **Requisitos del producto**
  - Especifican comportamiento del producto, por ejemplo la velocidad de ejecución, confiabilidad, etc.
- **Requisitos organizativos**
  - Son consecuencia de las políticas y procedimientos de la organización, por ejemplo estándares de procesos utilizados, requisitos de implementación, etc.
- **Requisitos externos**
  - Surgen de factores externos al sistema y su proceso de desarrollo, por ejemplo: requisitos de interoperabilidad, requisitos legislativos, etc.

# Tipos de requisitos no funcionales



# Requisitos no funcionales verificables

---

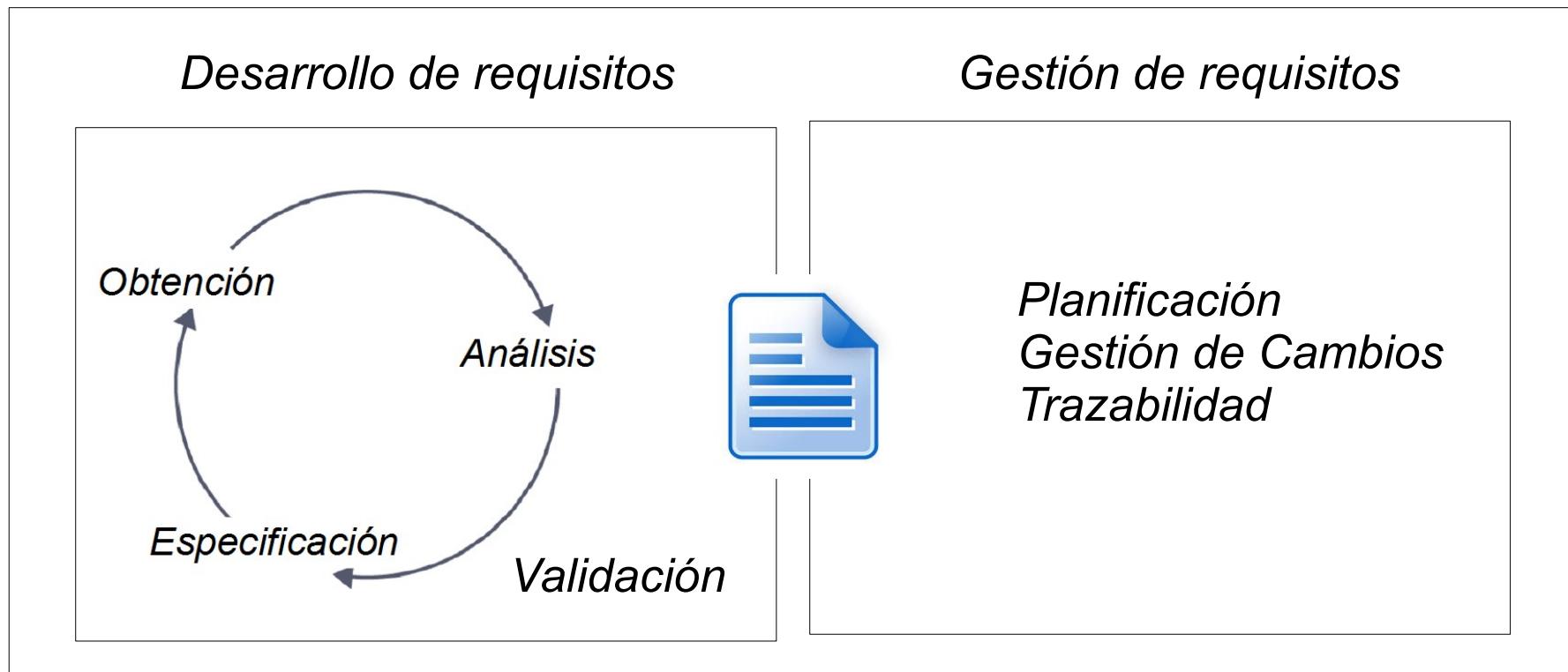
- Los requisitos no funcionales pueden ser muy difíciles de declarar de forma precisa y los requisitos imprecisos pueden ser difíciles de verificar.
- **Objetivo**
  - Una intención general del usuario como la facilidad de uso
- **Requisito no funcional verificable**
  - Una declaración con alguna medida que pueda ser probada objetivamente.
- Los objetivos son útiles para los desarrolladores, ya que transmiten las intenciones de los usuarios del sistema.

# Problemas en los requisitos

---

- La mayor consecuencia es el **retrabajo** (en etapas más avanzadas del desarrollo o después de liberar).
- Ejemplos:
  - Poco involucramiento de los usuarios.
  - Planes inadecuados – utilizar requisitos muy vagos para crear planes.
  - Recortes en los requisitos del usuario.
  - Requisitos ambiguos.
  - Gold plating – chapado en oro: requisitos que creemos que el usuario va a amar.
  - No identificar correctamente a los usuarios correctos.

# Actividades de la ingeniería de requisitos



# Especificación de requisitos

---

- Es el proceso de escribir los requisitos de usuario y del sistema en un documento de requisitos.
- Los requisitos de usuario deben ser entendidos por los usuarios finales y los clientes que no tienen formación técnica.
- Los requisitos del sistema son más detallados y pueden incluir mas información técnica.
- Los requisitos pueden ser parte de un contrato para el desarrollo del sistema. Por lo tanto, se debe intentar que sean lo más completos posible.

# Formas de escribir una especificación de requisitos

Notación	Descripción
Lenguaje natural	Los requisitos son escritos usando frases numeradas escritas en lenguaje natural. Cada frase debe expresar un requisito.
Lenguaje natural estructurado	Los requisitos son escritos en lenguaje natural utilizando un formulario estándar o plantilla. Cada campo proporciona información acerca de un aspecto del requisito.
Lenguajes de descripción de diseño	Este enfoque usa un lenguaje como un lenguaje de programación pero con características más abstractas para especificar los requisitos mediante la definición de un modelo operativo del sistema. Ahora este enfoque es raramente usado aunque puede ser útil para las especificaciones de interfaz.
Notaciones gráficas	Modelos gráficos complementados por anotaciones de texto son usados para definir los requisitos funcionales para el sistema. Comúnmente se utilizan diagramas UML de casos de uso y diagramas de secuencia.
Especificaciones matemáticas	Estas notaciones son basadas en conceptos matemáticos tales como máquinas de estados finitos o conjuntos. Aunque estas especificaciones pueden reducir la ambigüedad la mayoría de los clientes no entienden la especificación formal. Ellos no pueden chequear que se representa lo que ellos quieren y se resisten a aceptarlo como un contrato del sistema.

# Requisitos y diseño

---

- En principio, los requisitos deben indicar lo **qué** el sistema debe hacer y el diseño debe describir **cómo** lo debe hacer.
- En la práctica, es prácticamente imposible excluir toda la información de diseño al especificar en un nivel adecuado los requisitos de software.
  - La arquitectura del sistema puede ser diseñada para estructurar los requisitos.
  - El sistema puede interactuar con otros sistemas que generan requisitos de diseño.
  - El uso de una arquitectura específica para satisfacer los requisitos no funcionales puede ser un requisito.

# Especificación en lenguaje natural

---

- Los requisitos son escritos en frases en lenguaje natural.
- Se utiliza para escribir los requisitos porque es expresivo, intuitivo y universal. Esto significa que los requisitos pueden ser entendidos por los usuarios y los clientes.

# Especificación en lenguaje natural - Ejemplo

---

- 3.2 El sistema debe medir el azúcar en la sangre y liberar insulina, si es necesario cada 10 minutos. (*Los cambios en el azúcar en la sangre son relativamente lentos entonces mediciones más frecuentes son innecesarias; mediciones menos frecuentes podría llevar a niveles innecesariamente altos de azúcar.*)
- 3.6 El sistema debe ejecutar un auto-test de rutina cada minuto con las condiciones que deben verificarse y las acciones asociadas definidas en la Tabla 1. (*Una rutina de auto-test puede descubrir problemas en el hardware y software y alertar al usuario sobre el hecho de que la operación normal puede ser imposible.*)

# Guía para escribir requisitos

---

- Crear un formato estándar y usarlo para todos los requisitos.
- Usar un lenguaje de una manera consistente. El lenguaje usado para requisitos obligatorios debe ser el mismo que para requisitos deseables.
- Utilizar texto subrayado para identificar las partes clave de los requisitos.
- Evitar el uso de jerga informática.
- Incluir una explicación (lógica) de porqué es necesario un requisito.

# Problemas con el lenguaje natural

---

- **Falta de claridad**
  - Es difícil la precisión sin hacer que el documento sea difícil de leer.
- **Confusión de requisitos**
  - Los requisitos funcionales y no funcionales tienden a mezclarse.
- **Requisitos mezclados**
  - Varios requisitos diferentes pueden ser expresados juntos.

# Especificaciones estructuradas

---

- Es una aproximación a los requisitos donde la libertad del escritor de los requisitos es limitada y los requisitos son escritos de una manera estándar.
- Esto funciona bien para algunos tipos de requisitos ej: sistemas de control embebidos pero a veces es demasiado rígido para escribir los requisitos de sistemas de negocio.

# Especificaciones estructuradas – posibles secciones

---

- Definición de la función o entidad.
- Descripción de las entradas y de dónde vienen.
- Descripción de las salidas y a dónde ir.
- Información acerca de los datos necesario para el cálculo y otras entidades usadas.
- Descripción de las acciones a tomar.
- Pre y post condiciones (si es apropiado).
- Los efectos secundarios (si hay alguno) de la función.

# Especificación estructurada - ejemplo

## *Insulin Pump/Control Software/SRS/3.3.2*

<b>Function</b>	Compute insulin dose: Safe sugar level.
<b>Description</b>	Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.
<b>Inputs</b>	Current sugar reading (r2), the previous two readings (r0 and r1).
<b>Source</b>	Current sugar reading from sensor. Other readings from memory.
<b>Outputs</b>	CompDose—the dose in insulin to be delivered.
<b>Destination</b>	Main control loop.
<b>Action</b>	CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.
<b>Requirements</b>	Two previous readings so that the rate of change of sugar level can be computed.
<b>Pre-condition</b>	The insulin reservoir contains at least the maximum allowed single dose of insulin.
<b>Post-condition</b>	r0 is replaced by r1 then r1 is replaced by r2.
<b>Side effects</b>	None.

# Especificación tabular

---

- Usada para complementar el lenguaje natural.
- Particularmente útil cuando se tienen que definir una serie de posibles flujos alternativos.
- Por ejemplo, el sistema de la bomba de insulina basa sus cálculos en la tasa de cambio de nivel de azúcar en la sangre y la especificación tabular explica como calcular los requisitos de insulina para diferentes escenarios.

# Especificación tabular - ejemplo

Condition	Action
Sugar level falling ( $r_2 < r_1$ )	CompDose = 0
Sugar level stable ( $r_2 = r_1$ )	CompDose = 0
Sugar level increasing and rate of increase decreasing ( $(r_2 - r_1) < (r_1 - r_0)$ )	CompDose = 0
Sugar level increasing and rate of increase stable or increasing ( $(r_2 - r_1) \geq (r_1 - r_0)$ )	CompDose = round $((r_2 - r_1)/4)$ If rounded result = 0 then CompDose = MinimumDose

# Características deseables de un requisito según Wiegers

---

- **Completo** — contiene toda la información para entenderlo.
- **Correcto** — describe de forma precisa una necesidad y de forma clara la funcionalidad a desarrollar para cumplirla.
- **Factible** — es posible implementarlo dada las capacidades y limitaciones del sistema y su ambiente operacional.
- **Necesario** — tiene cierto valor para el negocio.
- **Priorizado** — fue priorizado según su importancia utilizando la perspectiva de todos los interesados.
- **No ambiguo** — no admite múltiples interpretaciones.
- **Verificable** — se pueden realizar pruebas para determinar si se encuentra presente en un producto de software o no.

# Características deseables de un conjunto de requisitos según Wiegers

---

- **Completo** — no hay información necesaria o de requisitos ausente.
- **Consistente** — no hay requisitos en conflictos con otros.
- **Modificable** — es posible reescribir un requisito.
- **Trazable** — un requisito puede ser rastreado hacia atrás a su origen y hacia adelante a elementos de diseño, código implementado y pruebas que verifican su implementación.

# Guías para la escritura de requisitos

---

- **Perspectiva del sistema o del usuario**
  - *[optional precondition] [optional trigger event] the system shall [expected system response].*
  - Si el químico solicitado se encuentra en el almacén químico, el sistema mostrará una lista de todos los contenedores del producto químico que actualmente hay en el almacén.
  - *The [user class or actor name] shall be able to [do something] [to some object] [qualifying conditions, response time, or quality statement].*
  - El químico deberá ser capaz de volver a ordenar cualquier producto químico que ha ordenado en el pasado, para lo cual podrá recuperar y editar una orden antigua.

# Guías para la escritura de requisitos

---

- **Estilo de escritura**

- Tratar de incluir la frase clave al principio de cada requisito (la declaración de la necesidad o de la funcionalidad) y luego los detalles accesorios (justificación, origen, prioridad, etc.)
- Evitar utilizar voces activas y pasivas de forma intercalada. Es más, es mejor usar voz activa.
- No utilizar múltiples términos para el mismo concepto.
- Escribir frases completas y utilizar correctas gramática, ortografía y puntuación.
- Mantener las oraciones y los párrafos cortos y directos.
- Evitar escribir largos párrafos con más de un requisito.

# Guías para la escritura de requisitos

---

- **Nivel de detalle**
  - Se debería incluir más detalle sí:
    - El trabajo será hecho para un cliente externo.
    - El desarrollo o las pruebas serán tercerizados.
    - Los miembros del equipo del proyecto están dispersos.
    - Las pruebas del sistema se basan en los requisitos.
    - Son necesarias estimaciones precisas.
    - La trazabilidad de los requisitos es importante.
  - Se puede incluir menos detalle cuando:
    - Los clientes están muy involucrados.
    - Los desarrolladores tienen mucha experiencia en el dominio.
    - Se dispone de precedentes (software a reemplazar)
    - Se usará una solución empaquetada.

# Guías para la escritura de requisitos

---

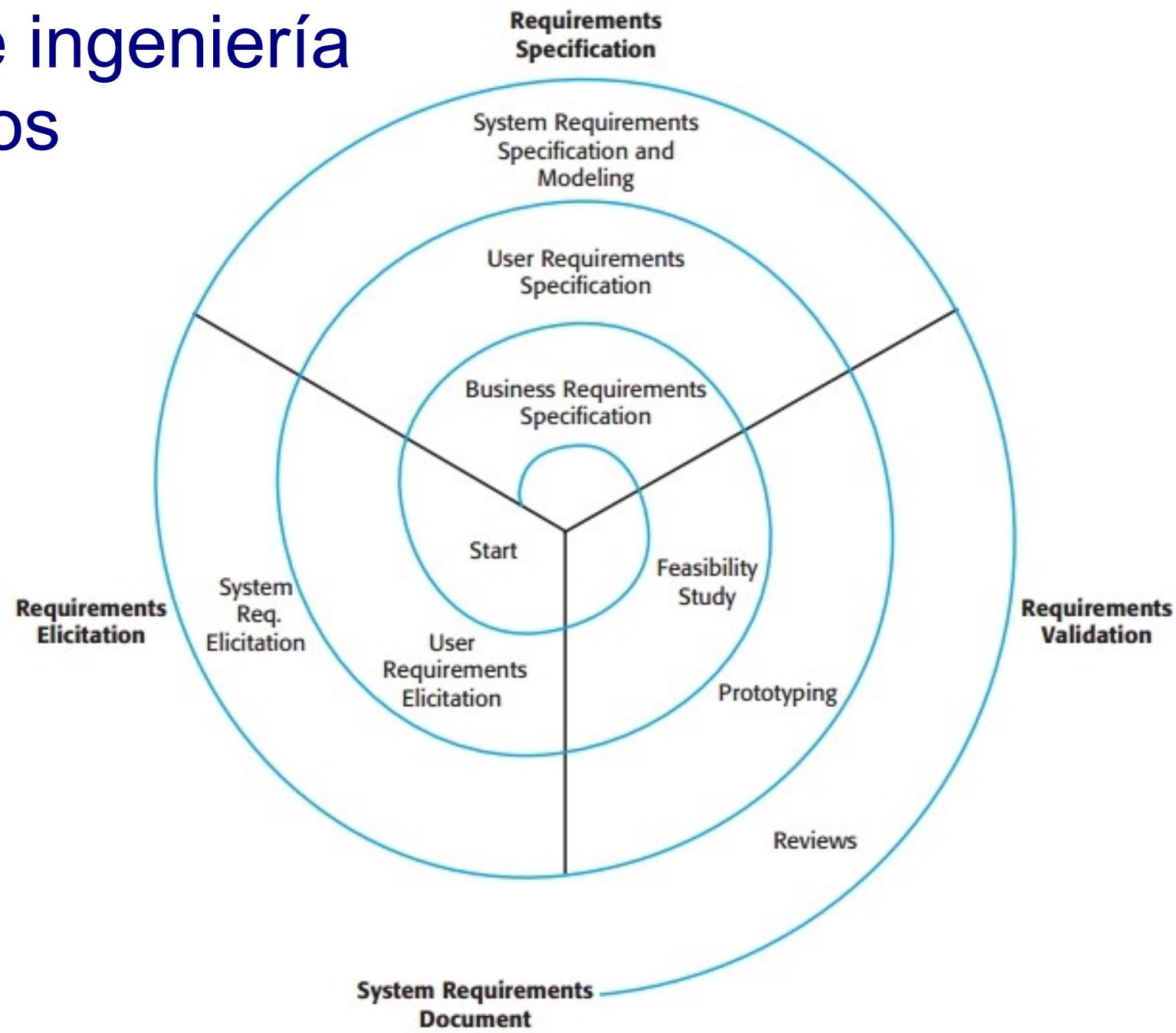
- **Nivel de detalle**
  - No todos los requisitos tienen que tener el mismo nivel de detalle. Aunque se recomienda mantener el mismo nivel en requisitos relacionados.
  - Una guía útil es escribir requisitos testeables de forma individual.
- **Evitar la ambigüedad**
  - Utilizar términos de forma consistente y como están definidos en el glosario.
  - Evitar adverbios (generalmente, rápidamente, etc)
  - Tratar de escribir todos los requisitos de forma positiva.
- **Evitar incompletitud**
  - Incluir operaciones simétricas (guardar borrador/recuperar)
  - Revisar en búsqueda de excepciones que falten.

# Los procesos de la ingeniería de requisitos

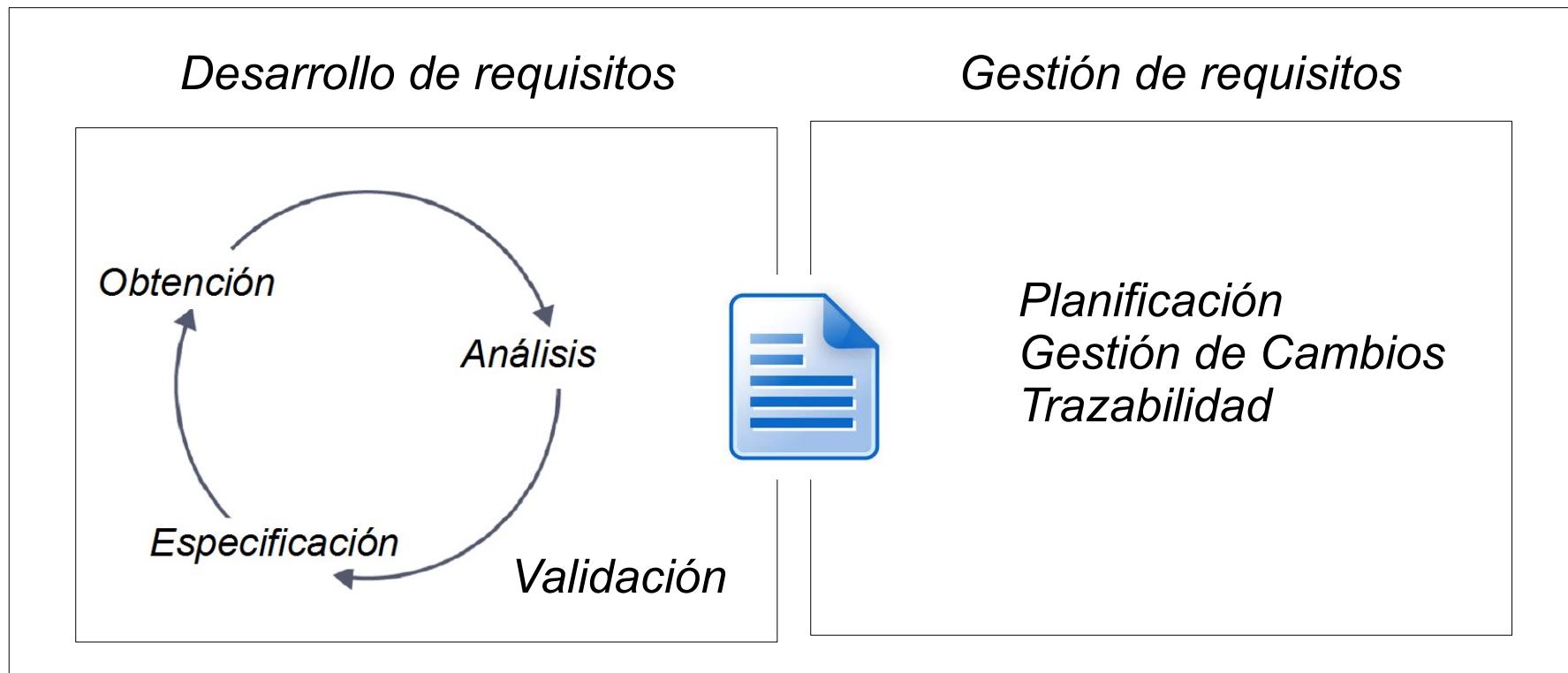
---

- Varían ampliamente dependiendo del dominio de la aplicación, las personas involucradas y la organización que desarrolla los requisitos.
- Sin embargo, hay una serie de actividades genéricas que son comunes a todos los procesos:
  - Estudio de factibilidad.
  - Relevamiento (u obtención) y análisis de requisitos.
  - Validación de requisitos.
  - Gestión de requisitos.
- En la práctica, la ingeniería de requisitos es una actividad iterativa en la cual se intercalan los procesos.

# Una vista en espiral del proceso de ingeniería en requisitos



# Actividades de la ingeniería de requisitos



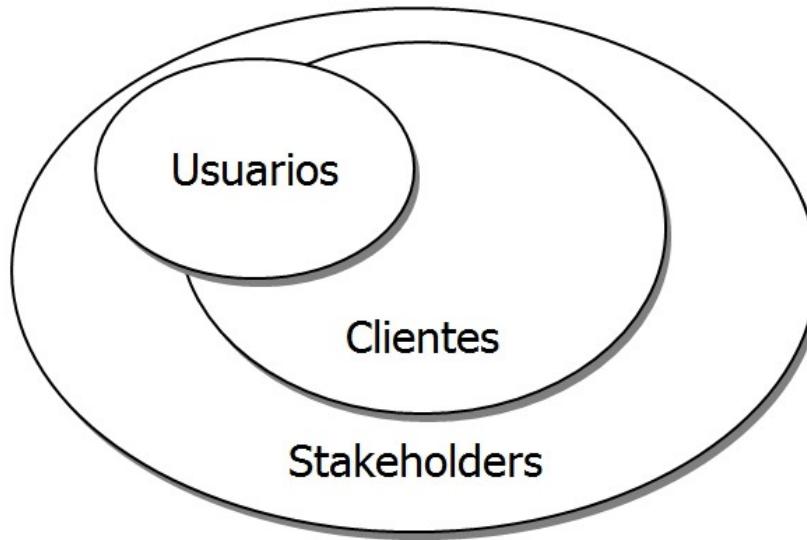
# Estudio de factibilidad

---

- Tiene como objetivo averiguar sí vale la pena implementar el sistema y sí es posible implementarlo dadas las restricciones existentes (calendario, presupuesto, tecnología, etc).
- Recibe como entradas un conjunto de requisitos del negocio preliminares, una breve descripción del sistema y cómo será destinado a apoyar los procesos del negocio.
- Brinda como salida un informe que recomienda sí vale la pena o no realizar el proceso de desarrollo del sistema.
- Si un sistema no da soporte a los objetivos del negocio, entonces no tiene ningún valor real para el negocio.

# Stakeholders - interesados

---



- **Cliente y Usuarios** — Requisitos adecuados a sus necesidades.
- **Diseñadores** — Comprenderlos para lograr diseño que los satisfaga.
- **Supervisores del Contrato** — Sugieren hitos de control, cronogramas.
- **Gerentes del Negocio** — Entienden impacto en la organización.
- **Verificadores** — Comprenderlos para poder verificar si el sistema los satisface.

# Ejemplo de stakeholders

---

- Pacientes cuya información es registrada en el sistema.
- Los doctores quienes son responsables por evaluar y dar tratamiento a los pacientes.
- Las enfermeras quienes coordinan las consultas con los doctores y administran algunos tratamientos.
- Recepcionistas médicos que administran las citas de los pacientes
- El personal de TI que son responsables de la instalación y mantenimiento del sistema.
- Un administrador de ética médica que debe asegurarse de que el sistema cumple con las normas éticas vigentes para la atención del paciente.
- Gerentes de la salud quienes obtienen información para la gestión del sistema.
- Equipo de registros médicos, son responsables de asegurarse que la información del sistema se mantiene y se conserva y que los procedimientos de mantenimiento de registro se ejecutan correctamente.

# Dificultades comunes en el proceso

---

- Los stakeholders no saben lo que realmente quieren.
- Los stakeholders expresan los requisitos en sus propios términos.
- Diferentes stakeholders pueden tener conflictos con sus requisitos.
- Factores organizacionales y políticos pueden influir en los requisitos del sistema.
- Los requisitos cambian (o su prioridad) durante el proceso de análisis. Pueden surgir nuevos stakeholders y el entorno empresarial puede cambiar.
- Influyen factores políticos.