

# Sesión 1

Unidad 1

**Introducción al Análisis de Sistemas**

Mg. Gustavo G. Delgado Ugarte

# **INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE**

# Ingeniería del Software

- La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza.

# Ingeniería del Software

- Disciplina de la Ingeniería
  - Los ingenieros
    - Aplican teorías, métodos y herramientas
    - Trabajan con restricciones financieras y organizacionales
- Aspectos de la producción de Software
  - No sólo comprende los procesos técnicos, sin también
    - Gestión de proyectos de software
    - Desarrollo de herramientas
    - Métodos y teorías de apoyo a la producción de software

# Proceso del Software

- Conjunto de actividades y resultados asociados que producen un producto de software
- Todos los procesos de software tienen 4 actividades comunes
  - Especificación del software
  - Desarrollo del software
  - Validación del software
  - Evolución del software

# Proceso del Software

- **Especificación.-** Se define el software a producir y las restricciones sobre su operación
- **Desarrollo.-** Diseño y programación del software
- **Validación.-** se valida el software para asegurar que es lo que el cliente requiere
- **Evolución.-** el software se modifica para adaptarlo a los cambios requeridos por el cliente y el mercado

# Modelo de Proceso

- Descripción simplificada de un proceso del software que presenta una visión de ese proceso
- La mayor parte de los modelos de procesos del software se basan en uno los siguientes modelos generales o paradigmas de desarrollo
  - Enfoque de la cascada
  - Desarrollo iterativo
  - Ingeniería del software basada en componentes (CBSE)

# Modelo de Proceso

- Enfoque de la cascada
  - Las actividades se representan como fases
    - Especificación de requerimientos
    - Diseño del software
    - Implementación
    - Pruebas, etc.
  - Después que cada etapa queda definida “se firma” y el desarrollo continúa con la siguiente etapa



# Modelo de Proceso

- Desarrollo iterativo
  - Entrelaza las actividades de especificación, desarrollo y validación
  - El sistema inicial se desarrolla rápidamente a partir de especificaciones muy abstractas
  - El sistema se refina en base a peticiones del cliente para producir un sistema que satisfaga sus necesidades
  - Luego, el sistema es entregado

# Modelo de Proceso

- Ingeniería del software basada en componentes
  - La técnica supone que las partes del sistema existen
  - El proceso de desarrollo del sistema se enfoca en integración de estas partes

# Métodos de Ingeniería del Software

- Enfoque estructurado para el desarrollo de software cuyo propósito es facilitar la producción de software de alta calidad de una forma costeable.
- Años 70
  - Métodos orientados a funciones
    - Análisis Estructurado
    - JSD

# Métodos de Ingeniería del Software

- Años 80 y 90
  - Métodos orientados a objetos
    - Propuestas de Booch y Rumbaugh
    - UML (Lenguaje de Modelado Unificado).- enfoque que integra los otros enfoques.

# **INTRODUCCIÓN A LA INGENIERÍA DE REQUERIMIENTOS**

# Introducción

- El desarrollo de sistemas basados en computador ha estado plagado de problemas desde la década de 1960 (Sommerville, 1997)
  - Sistemas entregados fuera de tiempo
  - Sistemas fuera del presupuesto
  - Sistemas que no hacen lo que los usuarios quieren
  - Sistemas que nunca son usados completamente

# Introducción

- Problemas comunes relacionados con los requerimientos de sistemas son (Sommerville,1997)
  - Los requerimientos no reflejan las necesidades reales de los clientes
  - Los requisitos son inconsistentes y/o incompletos.
  - Es muy caro hacer cambios a los requerimientos una vez que han sido acordados.
  - Existen malos entendidos entre los clientes, quienes desarrollan los requerimientos del sistema y los ingenieros de software que desarrollan o mantienen el sistema.

# Introducción

- Lo más difícil en la construcción de un sistema de software es decidir precisamente qué construir... No existe tarea con mayor capacidad de lesionar al sistema, cuando se hace mal... Ninguna otra tarea es tan difícil de rectificar a posteriori (F.P. Brooks, 1987)



# Introducción

- Boehm, 1975: 45% de los errores tienen su origen en los requisitos y en el diseño preliminar.
- DeMarco, 1984: 56% de los errores que tienen lugar en un proyecto de software, se deben a una mala especificación de requisitos.
- Chaos Report, 1995: Los factores principales que conducen al fracaso en los proyectos de software son:
  - Falta de comunicación con los usuarios
  - Requisitos incompletos
  - Cambios a los requisitos



LO QUE PIDE EL  
CLIENTE



LO QUE QUIERE OFRECER  
EL DPTO. COMERCIAL



LO QUE EXIGE  
GARANTÍA DE CALIDAD



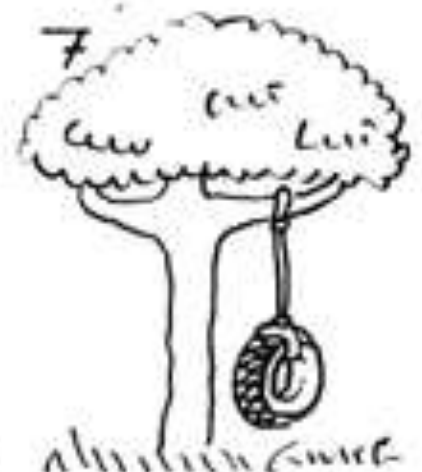
LO QUE DISEÑA  
PROYECTOS



LO QUE MATA  
PRODUCCIÓN



LA DISPOSICIÓN DEL  
DPTO. DE POSVENTA



LA ENTREGA FINAL  
AL CLIENTE



EL INFORME DEL  
JEFE DE PROGRAMA

# Ingeniería de Requerimientos

- El proceso de establecer los servicios que el cliente requiere de un sistema y las restricciones bajo las cuales opera y será desarrollado.
- Los requerimientos son descripciones de los servicios y restricciones generadas durante el proceso de ingeniería de requerimientos.

# ¿Qué es un requerimiento?

- Puede ser un conjunto de descripciones abstractas de alto nivel de un servicio o una restricción del sistema.
- Es una especificación formal matemáticamente detallada de una función del sistema.
- Por lo tanto, un requerimiento puede tener una función dual
  - Puede ser la base de un contrato, por lo tanto, debe tener una interpretación abierta a distintas propuestas de solución.
  - Puede ser el contrato por si solos, por lo tanto deben ser definidos en detalle.
  - Ambos se denominan documentos de requerimientos.

# Tipos de Requerimientos

- **Requerimientos de Usuario**
  - Son declarados en lenguaje natural apoyado por diagramas de los servicios que el sistema proveerá y sus restricciones operacionales. Escrito para clientes.
- **Requerimientos del Sistema**
  - Se establece en detalle y con precisión los servicios y restricciones del sistemas. Se denomina también especificación funcional. Sirve como contrato entre cliente y desarrollador.
- **Especificación de Software**
  - Descripción abstracta del diseño del software base que implementa un requerimiento. Escrito para desarrolladores.

# Requerimientos Funcionales y no Funcionales

- **Requerimientos Funcionales**
  - Declaraciones de los servicios del sistema, la manera como reaccionará a entradas límites o particulares. En algunos casos, declaran lo que el sistema no hará.
- **Requerimientos No Funcionales**
  - Son restricciones de los servicios o funciones ofrecidas por el sistema (restricciones de tiempos, restricciones sobre procesos de desarrollo, uso de estándares, etc.)
- **Requerimientos del Dominio**
  - Son requerimientos que provienen del dominio de aplicación del sistema y reflejan las características de este dominio.

# Requerimientos Funcionales

- Describen la funcionalidad o servicios del sistema.
- Dependen del tipo de software, expectativas del usuario y el tipo de sistema donde el software será usado.
- Los requerimientos funcionales del usuario pueden ser establecidos en alto nivel, en términos de lo que el sistema debe hacer, pero los requerimientos de funcionalidad del sistema deben describir los servicios del sistema en detalle.

# Ejemplos de requerimientos funcionales

- El usuario podrá buscar en todo la base de datos o en un subconjunto de ella.
- El sistema proveerá un visualizador apropiado para que el usuario lea los documentos almacenados en la base de datos.
- Toda orden se la asignará un identificador único (OrdenID) con el cuál el usuario será capaz de copiarla.



# Imprecisiones en los Requerimientos

- Los problemas provienen de requerimientos no precisamente establecidos.
- Requerimientos ambiguos pueden ser interpretados de diferentes maneras por clientes, usuarios y desarrolladores.
- Considere el término “rendimiento óptimo”

# Completitud y Consistencias de Requerimientos

- Los requerimiento tienen que ser tanto completos como consistentes.
- Completos
  - Se debe incluir todos los servicios requeridos por el usuario.
- Consistentes
  - No hay conflictos o contradicciones en los requerimientos definidos.

# Requerimientos No Funcionales

- Define propiedades y restricciones del sistema
- Ejemplo: Confiabilidad, tiempo de respuesta, seguridad, uso de herramientas y lenguajes, etc.
- Los requerimientos no funcionales pueden ser más críticos que los funcionales. Si estos no se cumplen, el sistema no se puede usar.

# Clasificación de Requerimientos No Funcionales

- **Requerimientos del Producto**
  - Especifican que el producto entregado debe tener una determinada velocidad de ejecución, confiabilidad, etc.
- **Requerimientos Organizacionales**
  - Son consecuencia de políticas y procedimientos de la organización, ejemplo: uso de estándares, lenguajes y herramientas, etc.
- **Requerimientos Externos**
  - Proviene de factores externos al sistema y su proceso de desarrollo, ejemplo: interoperabilidad con otros sistemas, legislación.

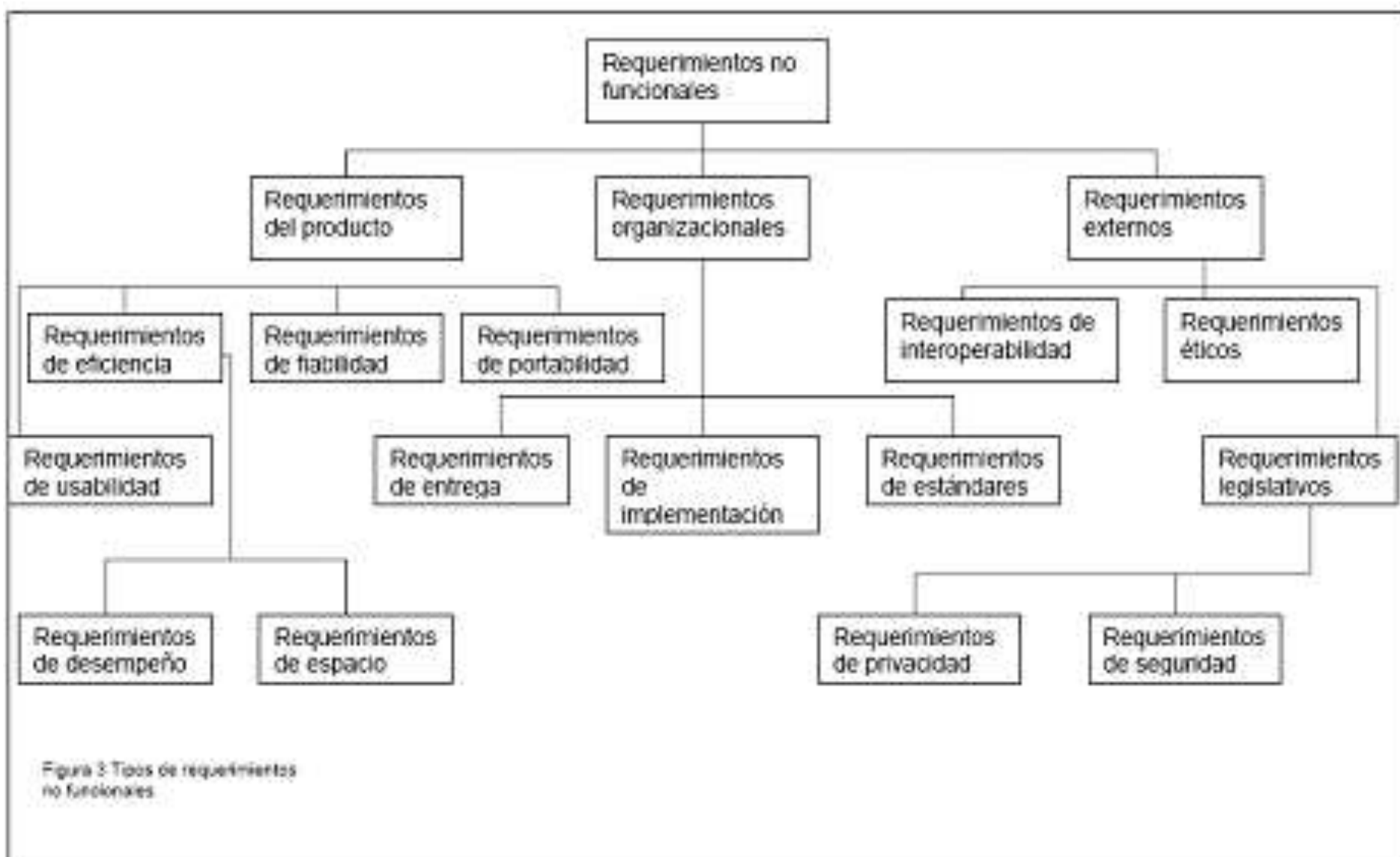


Figura 3 Tipos de requerimientos no funcionales

# Objetivos y Requerimientos

- Cumplimiento de Metas vs Verificabilidad
- Objetivo
  - Intención general de logro
- Verificabilidad
  - Declaración usando unidades de medida que pueden ser objetivamente verificadas
- Los objetivos son útiles para los desarrolladores ya que ellos expresan las intenciones de los usuarios del sistema.

# Ejemplo

- Objetivo del Sistema
  - El sistema debe ser fácil de usar de tal manera que se minimice la ocurrencia de errores
- Requerimiento no funcional verificable
  - Todos los usuarios deben ser capaces de usar el sistema después de 2 horas de entrenamiento. El número promedio de errores de un usuario entrenado no debe ser superior a 2 por día.

# Medidas de Requerimientos

Propiedad	Medida
Velocidad	Transacciones procesadas/segundo Tiempo de respuesta Usuario/Evento Tiempo de refresco de pantalla
Tamaño	K Bytes Número de chips RAM
Facilidad de Uso	Tiempo de entrenamiento Número de ayudas
Confiabilidad	Tiempo antes de falla Probabilidad de no disponibilidad Tasa de ocurrencia de fallas Disponibilidad
Robustez	Tiempo de recuperación ante fallas Porcentaje de eventos causantes de fallas Probabilidad de corrupción de datos cuando falla
Portabilidad	Porcentaje de líneas de código dependiente meta Número de sistemas meta



# Interacción entre Requerimientos

- Existe la posibilidad de conflictos entre diferentes requerimientos en un sistema complejo
- Sistema de una Nave Espacial
  - Para minimizar el peso, el número de chips separados que se debe usar debe ser el mínimo.
  - Para minimizar el poder de consumo, se debe usar chips de bajo consumo.
  - **Conflicto!!** Usar chips de bajo consumo puede significar tener que usar más chips.
  - **Decisión!!** ¿Cuál es el requerimiento más crítico?

# Requerimientos del Dominio

- Son derivados desde el dominio de la aplicación y describen características y atributos del sistema que reflejan los fundamentos del dominio de aplicación
- Pueden ser funcionales o no funcionales, nuevos o restringiendo los existentes o estableciendo cómo hacer cálculos específicos
- Si los requerimientos de dominio no se satisfacen, puede que sea imposible trabajar con el sistema.

# Problemas en los Requerimientos del Dominio

- Entendibilidad
  - Los requerimientos son expresados en el lenguaje del dominio de la aplicación
  - Este lenguaje, a menudo no es entendido por los ingenieros de software y desarrolladores del sistema.
- Implicancias (Obviedad)
  - Los especialistas del dominio entienden el área, pero pueden dejar fuera alguna información sobre algún requerimiento porque para ellos resulta obvio.

# Requerimientos de Usuario

- Describen los requerimientos funcionales y no funcionales, de tal forma que sean entendibles por los usuarios del sistema que no poseen conocimientos técnicos detallados.
- Los requerimientos de usuario son definidos usando lenguaje natural, representaciones y diagramas intuitivos sencillos.