

# *Ingeniería de Requerimientos*

## Prácticas – Curso 2007/08

### ■ Objetivos

- Aprender el manejo de una herramienta avanzada para el desarrollo rápido de prototipos: **Visual Prolog**

### ■ Plan

- Semana 1: Recomendaciones IEEE para la ERS (ETSIA?)
- Semanas 2-4: Tutorial (Visual) Prolog (LPP?)
- Semanas 5-12: Especificación lógica en Visual Prolog

### ■ Cambios de grupos

### ■ Prof. sustituto: Alicia Villanueva

# *Ingeniería de Requerimientos*

## Prácticas – Curso 2007/08

- **Evaluación (60% teoría – 40% prácticas)**
  - Para el enunciado obligatorio:
    - Especificación de requerimientos según IEEE
    - Implementación estilo operacional
    - Implementación estilo deductivo
  - Para cada uno de los enunciados voluntarios:
    - Especificación de requerimientos según IEEE
    - Implementación estilo operacional o estilo deductivo
- Grupos máximo 2 personas – **Evaluación individual!**
- Entrega parte obligatoria antes de Navidad: +1 punto

# Práctica 1

Recomendaciones prácticas de  
IEEE para la Especificación de  
Requerimientos Software (ERS)

Estándar ANSI/IEEE 830-1998

# Objetivos de la ERS

- Ayudar a los clientes a describir claramente lo que se desea obtener mediante un cierto software
- Ayudar a los desarrolladores a entender qué quiere exactamente el cliente
- Servir de base para desarrollos de estándares de ERS particulares para cada organización

# Ventajas de una buena ERS

- Contrato cliente – desarrolladores
- Reducción del esfuerzo de desarrollo
- Base para la estimación de costes y planificación
- Punto de referencia para procesos de verificación y validación
- Base para posibles mejoras

# Entorno de la ERS

- Una ERS forma parte de la documentación asociada al software:
  - Debe definir correctamente todos los requerimientos del software, pero no más
  - No debería describir ningún detalle de diseño, verificación, gestión del proyecto, etc.
- De esta forma, se deja el mayor grado de flexibilidad a los desarrolladores...

# Características de una buena ERS

- Correcta
- No ambigua
- Completa
- Consistente
- Etiquetada su **Novedad!** importancia/estabilidad
- Verificable
- Modificable
- Explorable

# 1.- Corrección

- Todos los requerimientos que aparecen en la ERS deben ser cumplidos por el software a desarrollar
- Debe ser coherente con cualquier documentación de mayor nivel



## 2.- Ambigüedad

- Lenguaje natural:

- Ej.: “Todos los clientes tienen el mismo campo de control”

- (1) ¿Todos tienen el mismo valor en el campo de control?
- (2) ¿Todos los campos de control tienen el mismo formato?
- (3) ¿Un campo de control se usa para todos los clientes?

- Lenguajes formales:

- No ambiguos
- Pero difíciles de aprender...

## 3.- Completitud

- Inclusión de todos los requerimientos significativos
- Definición de respuestas a todas las posibles clases de entradas, tanto válidas como inválidas, en todas las posibles situaciones
- Etiquetado de figuras, tablas, diagramas, etc., así como definición de términos y unidades de medida empleados
- **Evitar en lo posible el uso de la frase**  
*A determinar...*

## 4.- Consistencia

- Una ERS es consistente si no contiene requerimientos en conflicto mutuo:
  - Descripción del mismo objeto real con diferentes términos
  - “A siempre tras B” versus “A y B simultáneos”
  - Uso de diferentes términos para referirse al mismo objeto

## 5.- Etiquetada su importancia/estabilidad

- **Importancia**: No todos los requerimientos tienen la misma importancia
  - esenciales
  - condicionales
  - opcionales
- **Estabilidad**: se puede expresar en términos del número de cambios que se espera que puedan producirse sobre el requerimiento (experiencia)

## 6.- Verificabilidad

- Un requerimiento es verificable si existe algún proceso—no excesivamente costoso—por el cual una persona/máquina pueda chequear que el software satisface el requerimiento
- **No verificables:**
  - El producto debería funcionar bien
  - El producto debería tener una buena interfaz de usuario
- **Verificable:**
  - La salida se suministra dentro de los 20 segundos siguientes al evento E el 60% de las veces, y en los 30 segundos siguientes en el 100%

## 7.- Modificabilidad

- Una ERS es modificable si cualquier cambio puede realizarse de manera fácil, completa y consistente. Para ello es deseable:
  - Emplear una organización coherente y fácil de usar (índice, referencias cruzadas, etc.)
  - Evitar la redundancia
  - Expresar cada requerimiento de forma independiente (no mezclar la definición de varios requerimientos)

## 8.- Explorabilidad

- Una ERS es explorable si el origen de cada requerimiento es claro tanto hacia atrás como hacia delante
- Hace referencia al hecho de que la ERS no es documento estático, sino que evolucionará durante toda la vida del software...

# Preparación conjunta de las ERS

- Es fundamental que la ERS se escriba de forma conjunta entre el cliente y el equipo de desarrollo de software

**Novedad!**

- Los clientes no saben lo suficiente de informática para escribirla ellos mismos
- Los desarrolladores no conocen suficientemente los problemas y el campo de trabajo del cliente



# Evolución de las ERS

- La ERS debe evolucionar conforme lo hace el proceso de desarrollo de software:
  - Puede ser imposible especificar ciertos detalles al comienzo del proyecto (y cuando esto es así, debe indicarse claramente)
  - Conforme el producto evoluciona pueden aparecer deficiencias, incorrecciones, etc., en los requerimientos no detectadas previamente

# Métodos usados para expresar requerimientos

- Especificaciones de entradas / salidas
- Uso de ejemplos representativos
- Especificando modelos
  - matemáticos
  - funcionales (máq. estados finitos, redes de Petri, etc.)
  - temporales (en STR)

# Prototipado

**Novedad!**

- El uso de prototipos tiene muchas ventajas:
  - el cliente comprende mejor los requerimientos que sobre un documento en papel
  - permite descubrir aspectos imprevistos del comportamiento del sistema (dando lugar a nuevos requerimientos)
  - en general, un ERS desarrollado a partir de un prototipo suele ser más estable

# Recomendaciones generales (1/2)

## ■ Cuestiones a tratar en la ERS

- Funcionalidad: ¿Qué debe hacer el software?
- **Prestaciones**: Rendimiento, tiempo de respuesta, ...
- **Restricciones de diseño**: Lenguaje de implementación, recursos disponibles, entorno(s) de operación, etc.
- **Atributos**: Seguridad, portabilidad, mantenibilidad, etc.
- **Interfaces externos**: Gente, hardware, otro software

# Recomendaciones generales (2/2)

## ■ Se debe evitar

### ■ Introducir ideas de diseño

- estructura modular, flujos de información entre módulos
- lenguaje de programación
- estructuras de datos

### ■ Introducir ideas de gestión del proyecto

- costes
- plazos de entrega
- métodos de desarrollo
- plan de validación

# Esquema de una ERS

## Índice

### 1. Introducción

- 1.1. Propósito
- 1.2. Ámbito
- 1.3. Definiciones, acrónimos  
y abreviaturas
- 1.4. Referencias
- 1.5. Visión global

### 2. Descripción general

- 2.1. Perspectiva del producto
- 2.2. Funciones del producto
- 2.3. Características del usuario
- 2.4. Restricciones generales
- 2.5. Supuestos y dependencias
- 2.6. Requerimientos pendientes

### 3. Requerimientos específicos *(diferentes posibilidades de organización)*

## Apéndices

## Glosario

# Descripción secciones (1/4)

**1.1. Propósito:** esbozar el propósito de la ERS y especificar la supuesta audiencia

**1.2. Ámbito:**

- identificar el tipo de producto software por su nombre (por ejemplo, Editor, Base de datos, etc)
- explicar lo que hará el producto (y, si es necesario, lo que **no** hará)
- describir la aplicación del producto (beneficios y objetivos)

**1.3 Definiciones, acrónimos y abreviaturas**

**1.4 Referencias**

**1.5 Visión global:** describe el resto de la ERS y cómo está organizada

# Descripción secciones(2/4)

**2.1 Perspectiva del producto:** debe establecer las relaciones del producto con otros productos relacionados (y, si no hay relación, decirlo). Aquí se puede describir también los siguientes puntos:

- interfaces de sistema (por ej., con respecto a un sistema operativo)
- interfaces de usuario (formato de pantallas, disponibilidad de botones programados, etc.)
- interfaces hardware (configuración, periféricos, etc)
- interfaces software (bases de datos, etc)
- interfaces de comunicaciones (protocolos de red local, etc)
- restricciones de memoria (máximo de memoria disponible)
- operaciones (modos de operación, backups, etc)
- requerimientos de instalación



# Descripción secciones(3/4)

- **2.2 Funciones del producto:** suele ser un resumen de los requerimientos funcionales
- **2.3 Características del usuario:** nivel de estudios, experiencia, etc.  
(no establece requerimientos, sino que a menudo justifica los requerimientos que aparecen luego)
- **2.4 Restricciones generales:** debe describir de forma general los objetos que rodean al producto:
  - Regulaciones, limitaciones hardware, interfaces con otras aplicaciones, funciones de auditoría, requerimientos de seguridad, etc.

# Descripción secciones(4/4)

- **2.5 Supuestos y dependencias:** debe listar aquellos factores que pueden hacer que los requerimientos de la ERS cambien
  - Por ejemplo, la existencia o no de un sistema operativo determinado
- **2.6 Requerimientos pendientes:** requerimientos cuya definición se deja para futuras versiones de la ERS

**Novedad!**

# Requerimientos específicos

- Se puede organizar por
  - **módo** (entrenamiento, demo, normal, emergencia)
  - **clase de usuario** (administrador, usuario, cliente)
  - **objetos** (paciente, enfermera, sensor, médico, medicinas)
  - **función** (llamada local, conferencia, etc)
  - **estímulo** (pérdida de altura, frenos bloqueados, etc)
  - **respuesta** (generación de cheques de pago, generación de listados, etc)
  - **jerarquía funcional** (funciones con las mismas entradas, mismas salidas, o acceso a los mismos datos)

# Modelo Sección 3 (por modo)

## 3.1. Requerimientos de interfaces externos

- 3.1.1. Interfaces de usuario
- 3.1.2. Interfaces hardware
- 3.1.3. Interfaces software
- 3.1.4. Interfaces de comunicaciones

## 3.2. Requerimientos funcionales

- 3.2.1. Modo 1
  - 3.2.1.1. Requerimiento funcional 1.1
  - ...
  - 3.2.1.n. Requerimiento funcional 1.n
- 3.2.2. Modo 2
- ...

## 3.3. Requerimientos de eficiencia

## 3.4. Restricciones de diseño

- 3.4.1. Estándares cumplidos
- 3.4.2. Limitaciones hardware
- ...

## 3.5. Atributos

- 3.5.1. Seguridad
- 3.5.2. Mantenimiento
- ...

## 3.6. Otros requerimientos

- 3.6.1. Bases de Datos
- 3.6.2. Operaciones
- 3.6.3. Requerimientos de adaptación a situaciones
- ...

# Descripción secciones (1 / 3)

- **Requerimientos de interfaces externos:** descripción de las interfaces de usuario, hardware, software, de comunicaciones (como en el punto 2.1 de la ERS pero con más detalle y sin repetir información)
- **Requerimientos funcionales:** listado completo de todas las funciones del sistema (suelen comenzar con “El sistema debe...”) y puede incluir:
  - condiciones de validez de los datos de entrada
  - secuencia exacta de operaciones
  - respuesta a situaciones anómalas
  - relación entre entradas y salidas

# Descripción secciones (2/3)

- **Requerimientos de eficiencia:** requerimientos numéricos del software, tales como
  - el número de terminales que deben ser soportados por el software
  - el número de usuarios simultáneos
  - cantidad de información a manejar, etc
- **Restricciones de diseño:** restricciones impuestas por otros estándares, por limitaciones hardware, etc

# Descripción secciones (3/3)

- **Atributos:** incluyen restricciones de
  - **fiabilidad** (condiciones que debe cumplir en el momento de la entrega)
  - **seguridad** (técnicas de criptografía, mantener históricos, passwords, etc)
  - **mantenimiento** (modularidad, interfaces, etc)
  - **portabilidad** (porcentaje de código dependiente del sistema, uso de lenguajes portables, etc)
- **Otros requerimientos:** pueden incluir requerimientos sobre modos de operación, bases de datos (volumen, tipos de accesos, etc), ...

# Para la memoria de prácticas...

## Índice

### 1. Introducción

1.1. Propósito

1.2. Ámbito

1.3. Definiciones, acrónimos  
y abreviaturas  
generales

1.4. Referencias

1.5. Visión global

### 2. Descripción general

2.1. Perspectiva del producto

2.2. Funciones del producto

2.3. Características del usuario

2.4. Restricciones

2.5. Supuestos y dependencias

2.6. Requerimientos pendientes

3. Requerimientos específicos (*diferentes posibilidades de organización*)

Apéndices

Glosario



# Para la memoria de prácticas...

## 3.1. Requerimientos de interfaces externos

- 3.1.1. Interfaces de usuario
- 3.1.2. Interfaces hardware
- 3.1.3. Interfaces software
- 3.1.4. Interfaces de comunicaciones

## **3.2. Requerimientos funcionales**

### **3.2.1. Modo 1**

#### **3.2.1.1. Requerimiento funcional 1.1**

...

#### **3.2.1.n. Requerimiento funcional 1.n**

### **3.2.2. Modo 2**

...

## 3.3. Requerimientos de eficiencia

## 3.4. Restricciones de diseño

- 3.4.1. Estándares cumplidos
- 3.4.2. Limitaciones hardware
- ...

## 3.5. Atributos

- 3.5.1. Seguridad
- 3.5.2. Mantenimiento
- ...

## 3.6. Otros requerimientos

- 3.6.1. Bases de Datos
- 3.6.2. Operaciones
- 3.6.3. Requerimientos de adaptación a situaciones
- ...