INGENIERÍA DE REQUERIMIENTOS

Ciclo de vida y Requerimientos de software

Laboratorio de Programación

Parte 1

CICLO DE VIDA DEL SOFTWARE
(INTRODUCCIÓN)

Lorena Castañeda Bueno - Universidad Icesi – Laboratorio de Programación - 2010

MODELOS DE PROCESO DE SOFTWARE

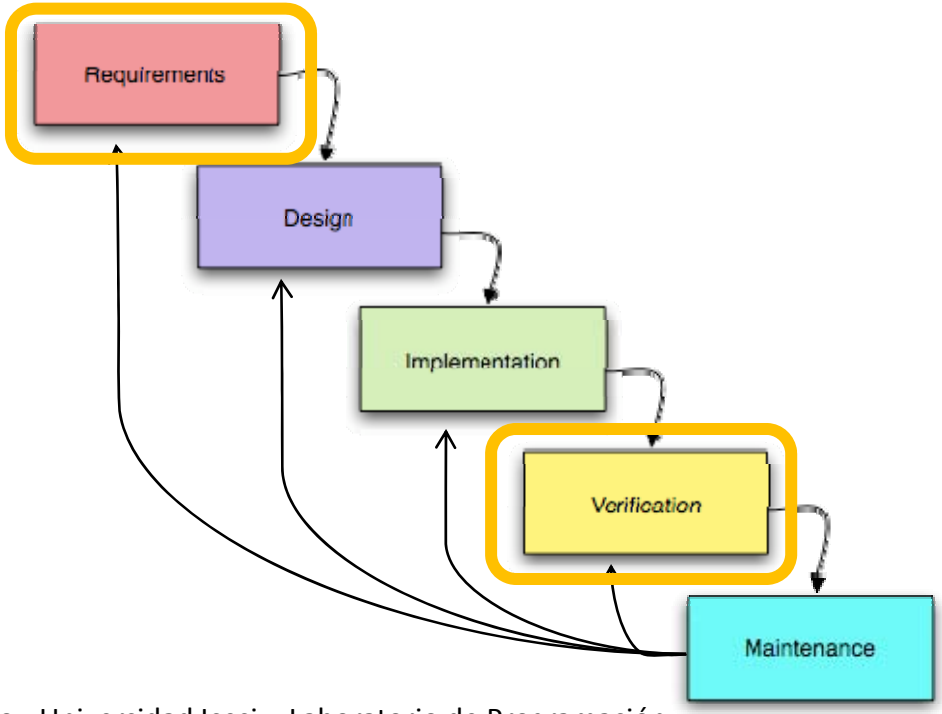
- Un modelo es una estructura guía, abstracciones, marcos del proceso que pueden se extendidos y adaptados para crear procesos más específicos
- Los modelos no son excluyentes
- Las actividades fundamentales de un proceso de software son
  - Especificación del software
  - Diseño e implementación del software
  - Validación del software
  - Evolución del software

MODELOS DE PROCESO DE SOFTWARE

- Para el caso del software existen 3 procesos comunes
  - Modelo en cascada: presenta las actividades como fases separadas
  - Desarrollo evolutivo: entrelaza las actividades, se desarrolla en forma de espiral
  - Basado en componentes: se basa en la idea de la existencia de componentes reutilizables, los incorpora mas no los desarrolla

MODELO EN CASCADA

- También conocido como “ciclo de vida del software”
- Planteado por primera vez por **Winston W. Royce** (1929–1995) en 1970 como parte del proceso de ingeniería de sistemas



CICLO DE VIDA DEL SOFTWARE

(ANÁLISIS Y DEFINICIÓN DE REQUERIMIENTOS)

- Los servicios, restricciones y metas del sistema
- Se definen a partir de las consultas de los usuarios
- Especificación del sistema
- Especificación del software

CICLO DE VIDA DEL SOFTWARE
(DISEÑO DEL SISTEMA Y DEL SOFTWARE)

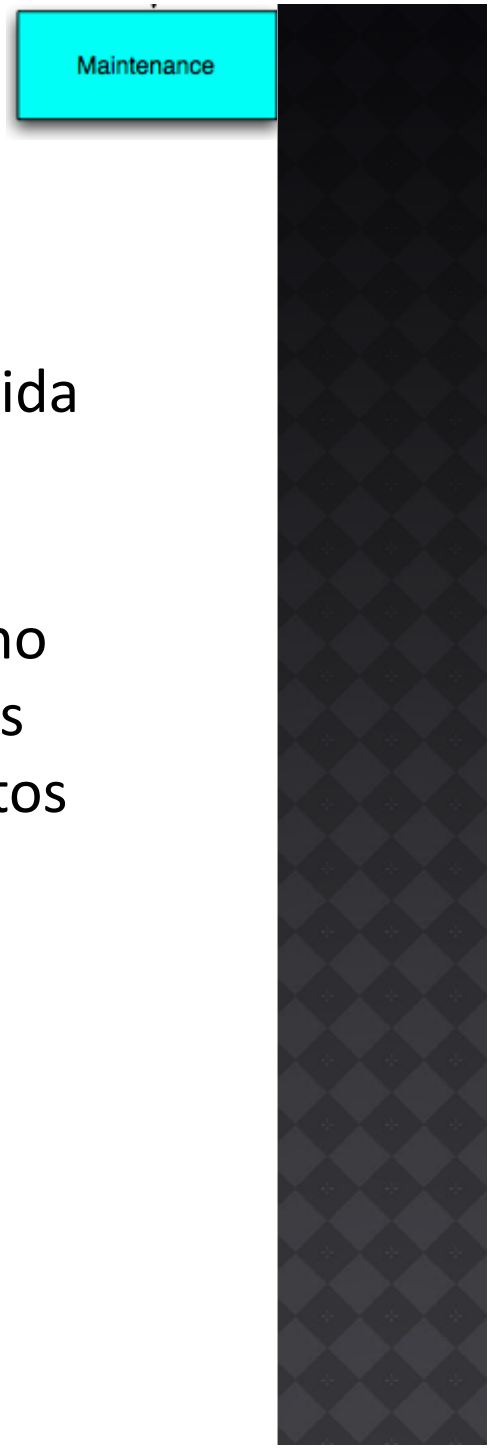
- Divide los requerimientos en hardware y software y sus relaciones
- Establece la arquitectura del sistema

CICLO DE VIDA DEL SOFTWARE
(IMPLEMENTACIÓN Y PRUEBAS)

- El diseño de software se hace como un conjunto de unidades de programas (“módulos”)
- Las pruebas de esta etapa se llaman **pruebas de unidad** y tienen como objetivo velar que cada parte cumpla su especificación

CICLO DE VIDA DEL SOFTWARE
(INTEGRACIÓN Y PRUEBAS)

- Las unidades de programa (módulos) se integran
- Las pruebas de esta etapa se llaman **pruebas de integración** y asegura que se cumplan los requerimientos de software
- Después de estas pruebas exitosas se entrega el producto al cliente



CICLO DE VIDA DEL SOFTWARE
(FUNCIONAMIENTO Y MANTENIMIENTO)

- Por lo general es la etapa más larga del ciclo de vida del software
- Luego de instalado el software, la etapa de mantenimiento incluye la corrección de errores no descubiertos en las etapas anteriores, mejorar las implementaciones y ajustar nuevos requerimientos
- A partir de esta fase la cascada se devuelve a cualquiera de las etapas anteriores

MODELO DE CASCADA

- El resultado de cada fase son documentos “firmados”
- Cada final de fase es requisito para el inicio de la siguiente
- No es un modelo lineal sino que implica ciertas iteraciones entre las etapas
- Su desventaja es que es inflexible ante nuevos requerimientos del cliente
- Es recomendado cuando los requerimientos se comprenden bien y sea improbable que cambien radicalmente durante el desarrollo del sistema

Parte 1

ANÁLISIS Y DEFINICIÓN DE REQUERIMIENTOS

ESPECIFICACIÓN DEL SOFTWARE

- Es la primera actividad dentro de los procesos de software
- Es el proceso de comprensión y definición de que servicios requiere el sistema
- Identificación de restricciones de funcionamiento y desarrollo
- Es la etapa mas importante dentro del proceso de software pues se delimita el alcance del desarrollo
- Un error en esta etapa representa inevitablemente problemas en el resto del desarrollo

INGENIERÍA DE REQUERIMIENTOS

- Este proceso de ingeniería conduce al documento de requerimientos (especificación del sistema)
- Se presenta en dos niveles de detalle:
  - Los usuarios y clientes: necesitan las especificaciones de alto nivel (funcionalidad)
  - Los desarrolladores: Necesitan las especificaciones de bajo nivel (programación)

INGENIERÍA DE REQUERIMIENTOS

- Se compone de cuatro fases principales
  1. Estudio de viabilidad
  2. Obtención y análisis de requerimientos
  3. Especificación de requerimientos
  4. Validación de requerimientos

La ingeniería de requerimientos es el arte de "saber preguntar"

VALIDACIÓN DEL SOFTWARE

- (V&V) La Verificación y Validación
- Se utiliza para demostrar que el sistema se ajusta a las especificaciones y cumple las expectativas del usuario final.
- Los sistemas no se deben probar como una simple unidad monolítica. Por eso se divide en 3 fases
  1. Pruebas de componentes (o unidades)
  2. Pruebas del sistema
  3. Pruebas de aceptación

VALIDACIÓN DEL SOFTWARE

- Las pruebas del software debe hacerse por un equipo aparte a los programadores
- Las pruebas son diseñadas previamente al desarrollo del software
- Las pruebas pueden dividirse comercialmente así:
  - Alfa: Pruebas de aceptación para un único cliente
  - Beta: Pruebas de aceptación de un producto comercial que se le entrega a un núcleo de clientes potenciales.

Parte 2

REQUERIMIENTOS DE SOFTWARE

Lorena Castañeda Bueno - Universidad Icesi – Laboratorio de Programación - 2010

DOCUMENTO: REQUERIMIENTOS DE SOFTWARE

- ¿Por qué no pueden ser el mismo, los requerimientos de usuario y los requerimientos del sistema?
- Explique con sus palabras qué es un requerimiento de dominio. Cite un ejemplo diferente al del libro
- Explique la importancia de la especificación de requerimientos.

REQUERIMIENTOS DE SOFTWARE

- Requerimientos de usuario
- Requerimientos del sistema (∃(x) de dominio*)
  - Funcionales
  - No Funcionales
    - De producto
    - Organizacional
    - Externo

- Otros: Requerimientos de interfaz de integración con otros sistemas: procedimientos, lenguajes, estructura de datos, etc.

*de dominio: Significa en el ambiente donde existe el sistema

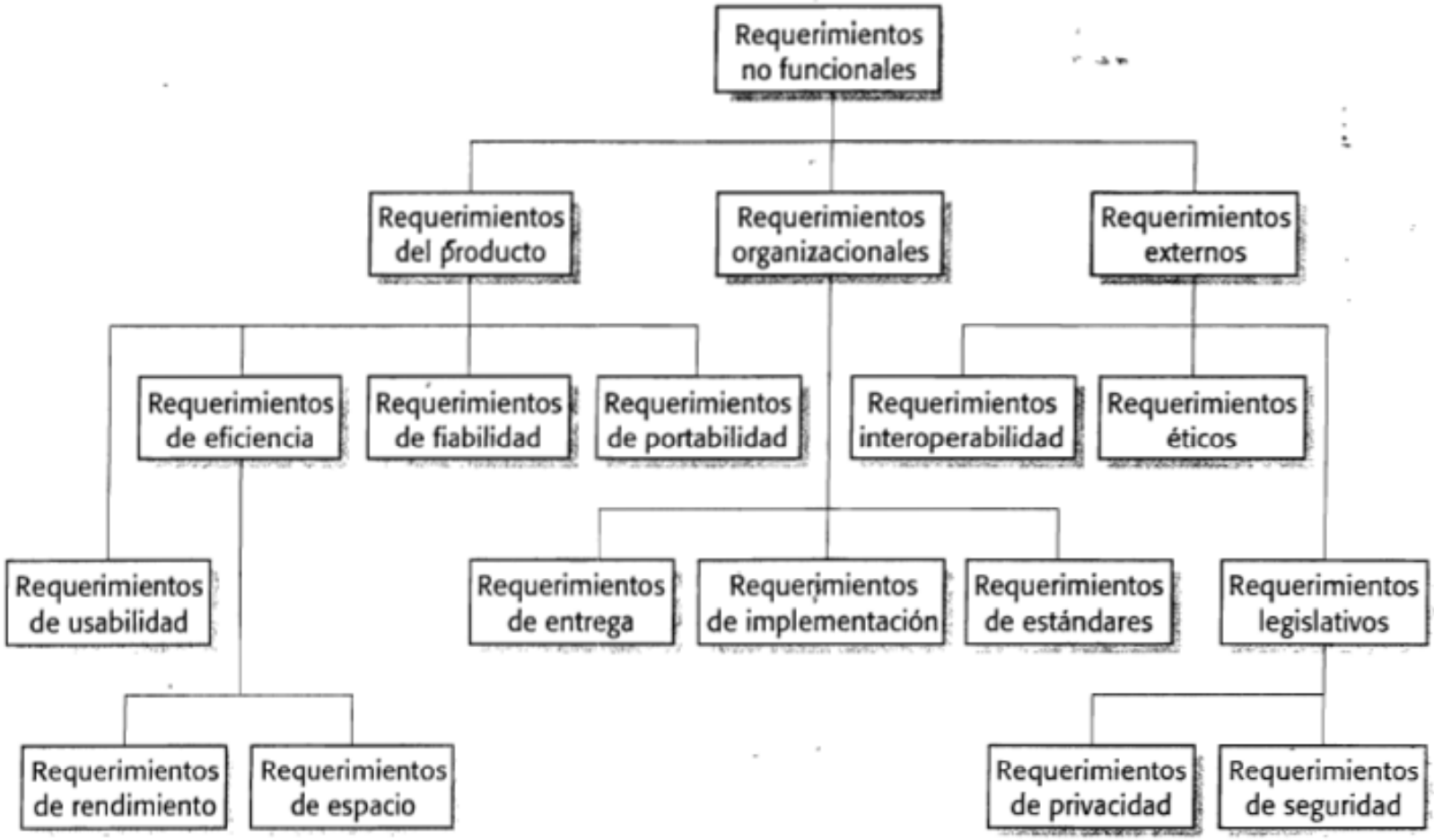


Figura 6.3 Tipos de requerimientos no funcionales.

Propiedad	Medida
Rapidez	Transacciones procesadas por segundo. Tiempo de respuesta al usuario y a eventos Tiempo de actualización de la pantalla
Tamaño	K Bytes Número de chips de RAM
Facilidad de uso	Tiempo de formación Número de cuadros de ayuda
Fiabilidad	Tiempo medio entre fallos Probabilidad de no disponibilidad Tasa de ocurrencia de fallos Disponibilidad
Robustez	Tiempo de reinicio después de fallos Porcentaje de eventos que provocan fallos Probabilidad de corrupción de los datos después de fallos
Portabilidad	Porcentaje de declaraciones dependientes del objetivo Número de sistemas objetivo

Figura 6.6 Métricas para especificar requerimientos no funcionales.

ESPECIFICACIÓN DE LOS REQUERIMIENTOS

- ¿A quién se le hacen especificaciones de los requerimientos?

- R:// A TODOS!!!
  - A los usuarios
  - A los clientes
  - A los administradores
  - A los ingenieros de sistemas
  - A los ingenieros de pruebas
  - A los ingenieros de mantenimiento

ESPECIFICACIÓN DE LOS REQUERIMIENTOS

- Estilos de especificación de los requerimientos:
- Estructurado (Formularios o plantillas)
- Descripción de diseño (secuencias algorítmicas)
- Gráfico (Casos de uso)
- Matemático (Notaciones máquinas de estado)

Bomba de insulina/Software de Control/SRS/3.3.2	
Función	Calcular la dosis de insulina: nivel de azúcar en sangre.
Descripción	Calcula la dosis de insulina a suministrar cuando el nivel medido actual del azúcar está en la zona segura entre 3 y 7 unidades.
Entradas	Lectura del azúcar actual (r2), las dos lecturas previas (r0 y r1).
Fuente	Lectura actual del azúcar del sensor. Otras lecturas de la memoria.
Salidas	CompDose: la dosis en insulina a suministrar.
Destino	Bucle de control principal.
Acción: CompDose es cero si el nivel de azúcar está estable o disminuyendo o si el nivel está aumentando pero la tasa de incremento disminuyendo. Si el nivel está aumentando y la tasa de incremento disminuyendo, CompDose se calcula dividiendo la diferencia entre el nivel de azúcar actual y el nivel anterior por 4 y redondeando el resultado. Si el resultado se redondea a cero, se fija CompDose a la dosis mínima que puede ser suministrada.	
Requerimientos	Las dos lecturas previas para poder calcular la tasa de cambio del azúcar.
Precondición	La reserva de insulina contiene al menos el máximo permitido para una única dosis de insulina.
Postcondición	r0 es reemplazada por r1 y r1 es reemplazada por r2.
Efectos colaterales	Ninguno.

Figura 6.12
La especificación de requerimientos del sistema utilizando un formulario estándar.

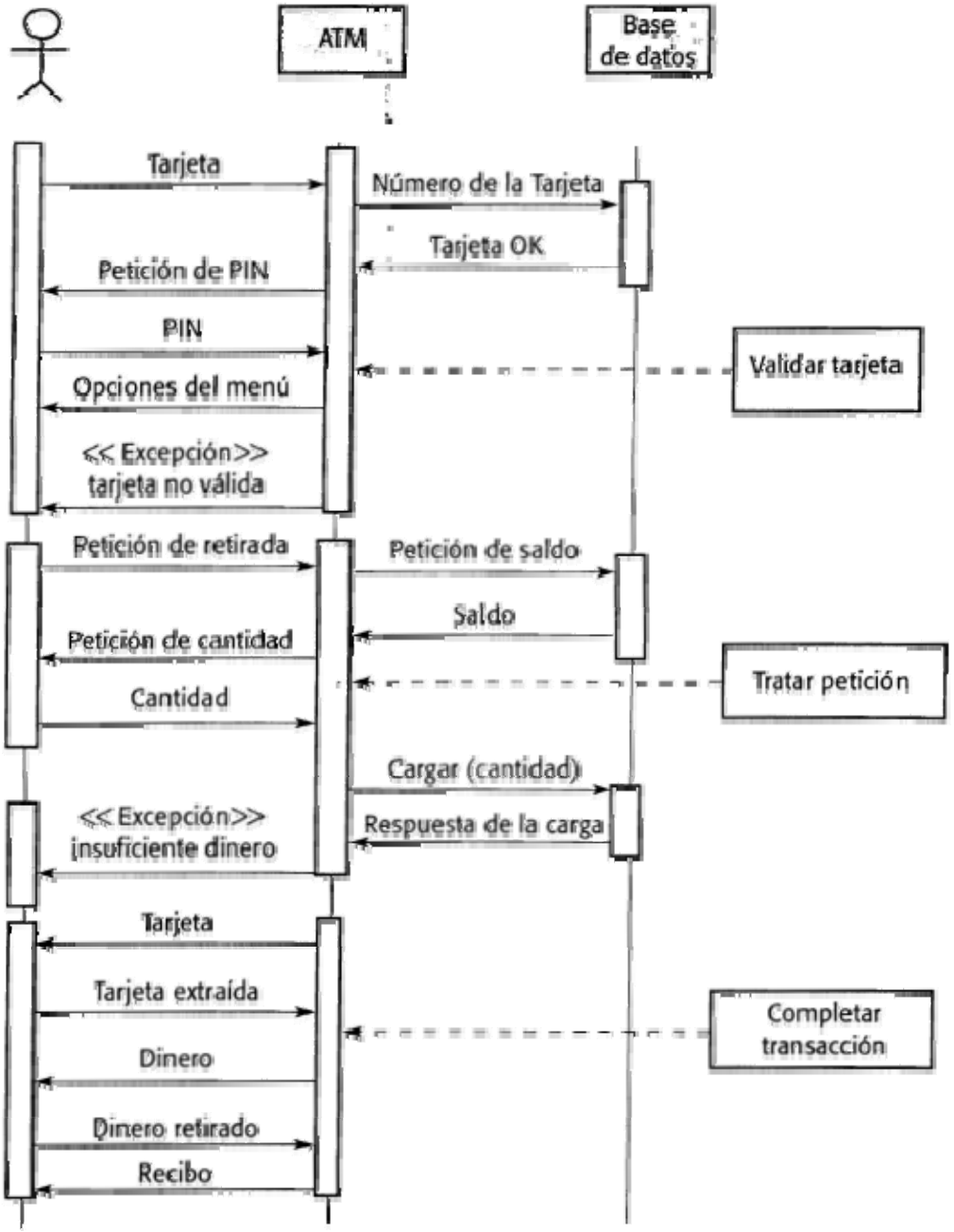


Figura 6.14
Diagrama de secuencia de la retirada de dinero en un ATM.