Sistemas Basados en el Conocimiento

Javier Béjar

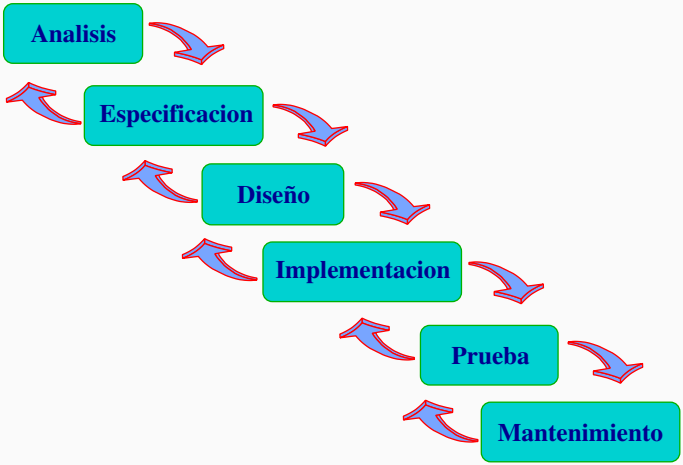
Curso 2021/2022

CS - FIB - UPC

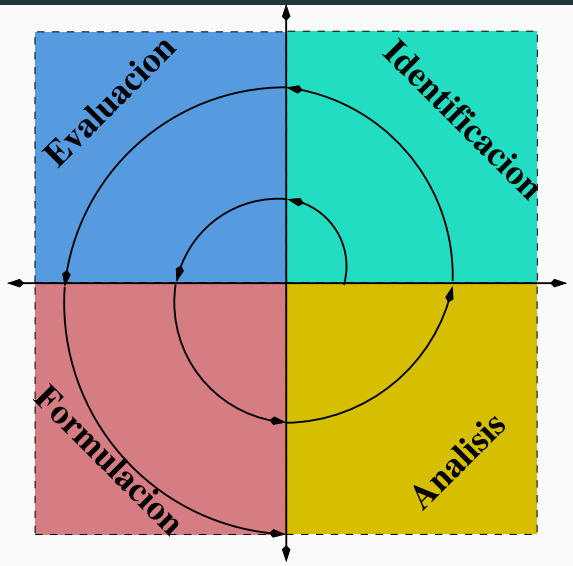
Desarrollo de SBC

- El punto más importante del desarrollo de SBC es la extracción del conocimiento
- Requiere la interacción entre el **Ingeniero del Conocimiento** y el experto
- Las metodologías de ingeniería de software han de encajar este proceso entre sus fases
- Las metodologías de ingeniería del software han de adaptarse a las características específicas de los SBC

IS: Modelo en cascada



IS: Modelo en espiral



Diferencias de los SBC

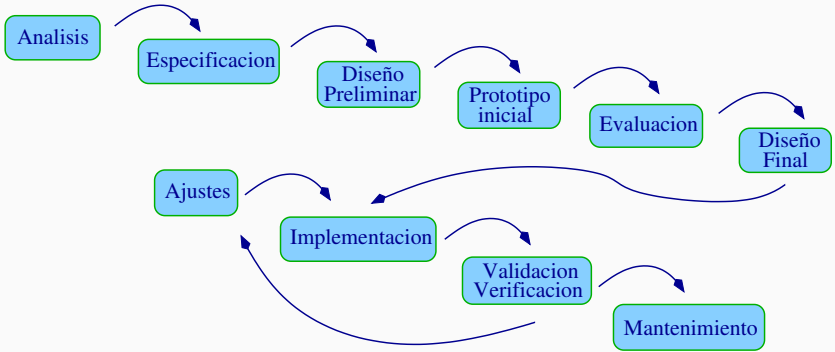
- Sistemas software convencionales $\Longrightarrow$ Algoritmos conocidos y de uso común
- SBC $\Longrightarrow$ Conocimiento incompleto, impreciso, heurístico
- Sistemas software convencionales $\Longrightarrow$ Posible estimar la naturaleza y cantidad del conocimiento
- SBC $\Longrightarrow$ Difícil estimar la naturaleza y cantidad del conocimiento

Diferencias de los SBC

- Es complicado obtener un diseño adecuado en las fases iniciales
- Decisiones iniciales erróneas pueden provocar el replanteamiento radical del diseño durante el desarrollo
- El **ingeniero del conocimiento** debe realizar un proceso de adquisición del conocimiento $\Longrightarrow$ Entrevistas con los expertos
  - El IC debe aprender los elementos básicos del dominio
  - Encontrar un formalismo representación que pueda entender el experto
  - Los expertos prefieren casos al razonamiento a partir de definiciones generales
  - A los expertos les es difícil explicitar su conocimiento en detalle (paradoja del experto)

- Solución: **Diseño incremental y prototipado rápido**
- Objetivo: Desarrollar un prototipo funcional que recoja las funcionalidades básicas del sistema
- El análisis y la especificación deben tener en cuenta el sistema completo
- El diseño e implementación se limita al prototipo inicial
- Este prototipo se completa incrementalmente
- Ventaja: Disponemos de un sistema funcional durante todo el proceso

Ciclo de vida de un SBC



Ciclo de vida de un SBC (I)

1. **Análisis del problema:** Recopilar información sobre el proyecto y determinar su viabilidad.
2. **Especificación de requerimientos:** Fijar los objetivos y métodos para conseguirlos.
3. **Diseño preliminar:** Decisiones a alto nivel sobre el diseño (formalismo de representación del conocimiento, herramientas, fuentes de conocimiento)
4. **Prototipo Inicial y evaluación:** Construir un prototipo con cobertura limitada, evaluar las decisiones de diseño a partir del prototipo
5. **Diseño final:** Validar las decisiones y proponer el diseño del sistema de manera que permita un desarrollo incremental.

Ciclo de vida de un SBC (II)

6. **Implementación:** Completar la adquisición del conocimiento, ampliar incrementalmente el prototipo inicial.
7. **Validación y verificación:** Comprobar que el sistema cumple las especificaciones.
8. **Ajustes de diseño:** Realimentar el proceso (los cambios en el diseño deberían se mínimos)
9. **Mantenimiento:** Mantener el sistema.

- **CommonKADS**
  - Ciclo de vida en espiral y modelado mediante herramientas parecidas a UML
  - Se construyen seis modelos: Organización, tareas, agentes, comunicación, conocimiento y diseño.
- **MIKE**
  - Ciclo de vida en espiral: Adquisición del conocimiento (modelo de adquisición y modelo de estructura), diseño, implementación, evaluación.

Una metodología simplificada

- Para aplicaciones pequeñas se puede aplicar una metodología en cascada que integra todo el proceso de desarrollo
  1. **Identificación del problema**
  2. **Conceptualización**
  3. **Formalización**
  4. **Implementación**
  5. **Validación y Prueba**

Identificación

- Debemos determinar si el problema es adecuado
  - ¿Hay una solución algorítmica?
  - ¿Disponemos de fuentes de conocimiento?
  - ¿El tamaño/objetivo/complejidad del problema es adecuado?
- Buscar y evaluar las fuentes de conocimiento
- Determinar el conocimiento necesario para el sistema
- Establecer los objetivos del sistema (¿Que respuesta esperamos?)

Conceptualización

Esta fase nos debería dar la perspectiva del problema desde el punto de vista del experto

- Deberemos:
  - Detallar los elementos del dominio $\Longrightarrow$ Descripción informal de la ontología
  - Descomponer el problema en subproblemas mediante refinamientos sucesivos, descubriendo los bloques de razonamiento
  - Detallar el flujo de razonamiento y las entradas y salidas de cada subproblema
  - Detallar y distinguir entre evidencias, hipótesis y acciones y descubrir sus relaciones
- Toda esta información la obtendremos a partir de la interacción con el experto (entrevistas) y las fuentes de conocimiento
- El resultado será un modelo semiformal del dominio y de los problemas y métodos de resolución

Formalización

Esta fase transformará la perspectiva del experto en la perspectiva del ingeniero del conocimiento

- Decidir el formalismo de representación del conocimiento adecuado
- Identificar el espacio de búsqueda
- Analizar la tipología de los problemas y bloques de razonamiento y decidir los métodos de resolución de problemas adecuados
- Analizar la necesidad de tratamiento de incertidumbre y/o información incompleta

Implementación

- Construir una ontología del dominio
- Encajar los problemas identificados en las metodologías de resolución de problemas escogidas
- Construir los diferentes módulos que correspondan a cada problema siguiendo el conocimiento obtenido
- Si utilizamos una aproximación basada en prototipado rápido construiremos el prototipo inicial y lo iremos aumentando incrementalmente

Validación y Prueba

- Escoger casos representativos y resolverlos mediante el sistema
- Los casos deberían incluir tanto casos usados para la construcción del sistema como casos nuevos
- Si seguimos una estrategia de construcción incremental esta fase se irá repitiendo a medida que se desarrolle el prototipo
- La validación de SBC es más compleja que la de los sistemas de software habituales