CommonKADS + SCRUM

Para desarrollar la base de conocimiento (BC) del Sistema Experto (SE), se utilizan metodologías específicas de Ingeniería del Conocimiento (IC), un subcampo de la Inteligencia Artificial enfocado en adquirir, modelar, validar y gestionar conocimiento experto.

Para nuestro caso usaremos la metodología CommonKADS (Knowledge Acquisition and Design Structuring). Esta es la metodología estándar para desarrollar sistemas basados en conocimiento. Se centra en **modelar el conocimiento** de forma sistemática, independiente de la implementación técnica.

* **Fases clave aplicables a la BC**:
  + **Modelo de Conocimiento**: Define conceptos, reglas, hechos y relaciones del dominio (ej.: "SI el paciente tiene fiebre alta Y tos persistente, ENTONCES posible neumonía").
  + **Modelo de Adquisición**: Guía la extracción de conocimiento de expertos humanos (mediante entrevistas, observación, análisis de casos, etc.).
  + **Modelo de Validación**: Verifica la consistencia, completitud y precisión de la BC.
* **Ventajas**:
  + Estructura clara, evita errores comunes como conocimiento incompleto o contradictorio.
  + Permite reutilizar patrones de conocimiento.
  + Documentación robusta.

Esta metodología estructurada para ingeniería del conocimiento (CommonKADS) la combinaremos con una metodología ágil para gestión de proyectos como lo es SCRUM, esto nos permitirá desarrollar la base de conocimiento (BC) de un Sistema Experto de forma **r**igurosa pero flexible. Esta integración aprovecha la solidez de CommonKADS en modelado de conocimiento y la adaptabilidad de SCRUM para gestionar incertidumbre y cambios.

**1. División Natural de Responsabilidades**

| **CommonKADS 🧠** | **Scrum 🚀** |
| --- | --- |
| Gestiona el **contenido** (conocimiento experto): reglas, ontologías, lógica de inferencia. | Gestiona el **proceso**: planificación, entregas iterativas, coordinación del equipo técnico. |
| Se enfoca en **qué** debe saber el sistema (modelado de dominio). | Se enfoca en **cómo** implementarlo (código, pruebas, despliegue). |
| Valida **calidad del conocimiento** (precisión, cobertura). | Valida **funcionalidad técnica** (rendimiento, usabilidad). |

**2. Roles Híbridos en el Equipo**

| **Rol Scrum** | **Adaptación para CommonKADS + Sistema Experto** |
| --- | --- |
| **Product Owner** | **Knowledge Owner**: Un experto del dominio (ej: médico en un SE médico) que prioriza reglas y valida el modelo de conocimiento. |
| **Scrum Master** | Mantiene su rol, pero facilita la comunicación entre *ingenieros de conocimiento* y *desarrolladores*. |
| **Equipo de Desarrollo** | Incluye: - **Ingenieros de conocimiento** (modelado CommonKADS). - **Desarrolladores** (implementación técnica). - **Data Scientists** (si se usa ML para generar reglas). |

**3. Flujo de Trabajo Integrado (Sprints Adaptados)**

**Fase 0: Pre-Sprint (CommonKADS Inicial)**

* **Actividades**:
  1. Modelado de alto nivel del conocimiento (identificar conceptos clave y alcance).
  2. Extracción inicial de reglas con expertos (entrevistas).
* **Salida**:
  1. Un **"Knowledge Backlog"** (priorizado por el Knowledge Owner).

**Sprint Planning (Híbrido)**

* **Knowledge Backlog ➝ Sprint Backlog**:
  + Se seleccionan *historias de usuario técnicas* (ej: "Implementar motor de reglas") y *tareas de conocimiento* (ej: "Modelar 20 reglas para diagnóstico de diabetes").
* **Criterio de Aceptación Dual**:
  + Técnico: "La regla se ejecuta sin errores".
  + De Conocimiento: "La regla fue validada por el experto con 95% de precisión".

**Ejecución del Sprint**

| **Tarea CommonKADS** | **Actividades en Scrum** | **Herramientas** |
| --- | --- | --- |
| Adquisición de conocimiento | Reuniones diarias con expertos (vía *Daily* extendida). | Zoom, Miro para lluvia de ideas. |
| Modelado de reglas | Desarrollo de "artefactos de conocimiento": tablas de decisiones, diagramas. | Protégé, Excel. |
| Validación | Pruebas con casos de uso reales (el experto prueba el incremento). | Sets de datos históricos. |

**Sprint Review (Validación Conjunta)**

* **Demostración técnica** (desarrolladores): Muestra la funcionalidad implementada.
* **Validación de conocimiento** (Knowledge Owner): Verifica que las reglas reflejen su expertise.
* **Ejemplo**:

\*"Regla #45: Si glucosa > 140 mg/dL y sed excesiva ➔ Prediabetes.  
✅ Técnico: Se ejecuta en 0.2 segundos.  
✅ Conocimiento: Aprobada por endocrinólogo (precisión 98%)."\*

**Retrospectiva (Mejora Continua)**

* **Preguntas clave**:
  + ¿El flujo de adquisición de conocimiento fue ágil?
  + ¿Las reglas validadas se implementaron sin distorsión?

**4. Ejemplo Práctico: Sistema Experto Médico**

**Sprint 1: Diagnóstico de Enfermedades Respiratorias**

* **Objetivo**: Implementar 30 reglas validadas para 5 enfermedades.
* **Tareas**:

| **CommonKADS** | **Scrum** |
| --- | --- |
| Adquirir reglas de neumonía/gripe (entrevista con 2 médicos). | Crear módulo de motor de reglas en Python. |
| Validar con 50 casos históricos. | Desarrollar interfaz para ingresar síntomas. |

* **Resultado**:
  + Incremento funcional: Diagnóstico de neumonía con 90% precisión.

**5. Ventajas Clave de la Combinación**

* **Riesgos mitigados**:
  + **Knowledge drift**: El conocimiento se valida *cada sprint* (no al final del proyecto).
  + **Desalineación**: Los desarrolladores ven el contexto del dominio en tiempo real.
* **Eficiencia**:
  + El modelo de conocimiento (CommonKADS) guía la implementación (Scrum), evitando reprocesos.
  + Los expertos participan activamente (no son "recursos externos").

**Conclusión**

**CommonKADS + Scrum = Calidad de conocimiento + Entrega ágil**:

* CommonKADS estructura el **"qué"** (conocimiento experto riguroso).
* Scrum gestiona el **"cómo"** (implementación iterativa y adaptativa).  
  **Recomendación final**: