

智能知识工程体系

从“原子化笔记”到 代理式工作流 的系统化重构

“知识不再是静止的文本，而是可计算、可版本化、可执行的代码。我们将从知识的‘记录者’进化为智能系统的‘架构师’。”

认知范式：从“存取”到“工程”

传统知识管理

📖 隐喻：图书馆 (Library)

静态存储，关注分类与归档。

💡 痛点：版本混乱

"最终版_v2_修改.docx"，难以回溯。

智能知识工程

🔗 隐喻：软件仓库 (IDE)

动态流转，关注版本演进与自动化。

⚡ 优势：可计算性

知识即代码，支持 CI/CD 与 Agent 调用。

核心基座：知识即代码 (Git Flow)

引入分布式版本控制，让思维的演化可追溯、可分叉、可合并。



数据层：机器可读的原子结构

MARKDOWN SOURCE

```
---
uuid: "20250501-A1"
type: "Concept"
status: "Evergreen"
tags: [#AI/RAG, #Engineering]
---
```

RAG的原子化原理

传统的固定分块容易截断上下文。原子化笔记要求每个存储单元对应一个 **完整概念** ...

GraphRAG 增强

- 元数据：RAG 预过滤 (Pre-filtering)
- 合成问题：增加语义检索入口
- 图谱关系：支持多跳推理与溯源

交互与代理：构建自主闭环

Prompt 定义协议，Workflow 赋予行动力



1. 规划 (Planning)

任务拆解与路径生成，将宏大目标转化为可执行步骤。

Input: Goal
Action: Decompose



2. 工具 (Tool Use)

自主调用 RAG 检索、搜索 API 或代码解释器。

Func: search_docs()
Func: run_python()



3. 反思 (Reflection)

自我修正与结果评估，形成 "Generate-Critique-Refine" 闭环。

Check: Consistency
Loop: Refine

🔄 自主迭代闭环 (Self-Correction Loop)

Knowledge Ops：持续集成流水线



IssueOps

结构化需求工单
自动化触发



CI 质量门禁

自动 Linting
一致性扫描 (LLM)



CD 自动分发

重建向量索引
发布静态花园

FINAL TAKEAWAY

从「我用 AI 做任务」
到「AI 与我一起构建任务系统」。