



AI Product Case Study

LifeOS AI

个人AI记录与行动助手

从个人记录出发，让AI将日记转化为理解、洞察与可确认的行动。

Windows-first MVP · v0.1.2

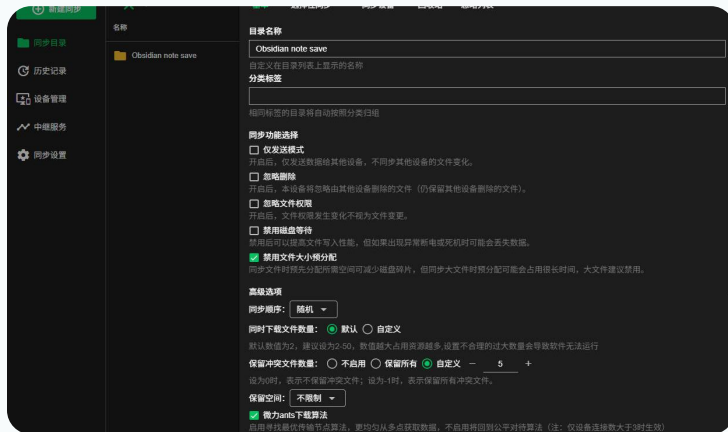
Product Owner · 谭正杰



为什么会有 LifeOS?



Obsidian 个人工作区

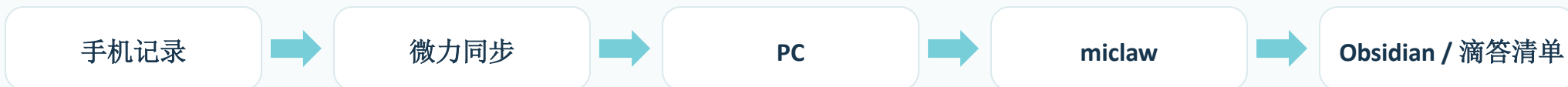


微力同步



miclaw 自动任务

个人真实工作流：每天记录 → 同步到电脑 → AI读取与整理 → 写入Obsidian / 滴答清单



多工具工作流



统一产品 LifeOS AI

核心闭环：让记录真正进入行动

LifeOS不只总结日记，而是尝试把个人记录转化为可由用户确认的行动。



AI负责理解和建议；最终是否行动，由用户确认。为保护真实个人隐私，本页使用Synthetic Demo Diary / 演示测试日记。

作为 Product Owner，我做出的3个关键决策

01

数据策略

Local First + Markdown First

用户正文长期由Markdown持有，SQLite承担可重建索引。

数据自主可控
+可检索性

02

AI控制边界

建议 $\neq$ 自动执行

AI输出：理解 / 洞察 / 行动建议
只有用户确认后，才正式创建Task。

AI负责建议
+用户保留决定权

03

MVP范围

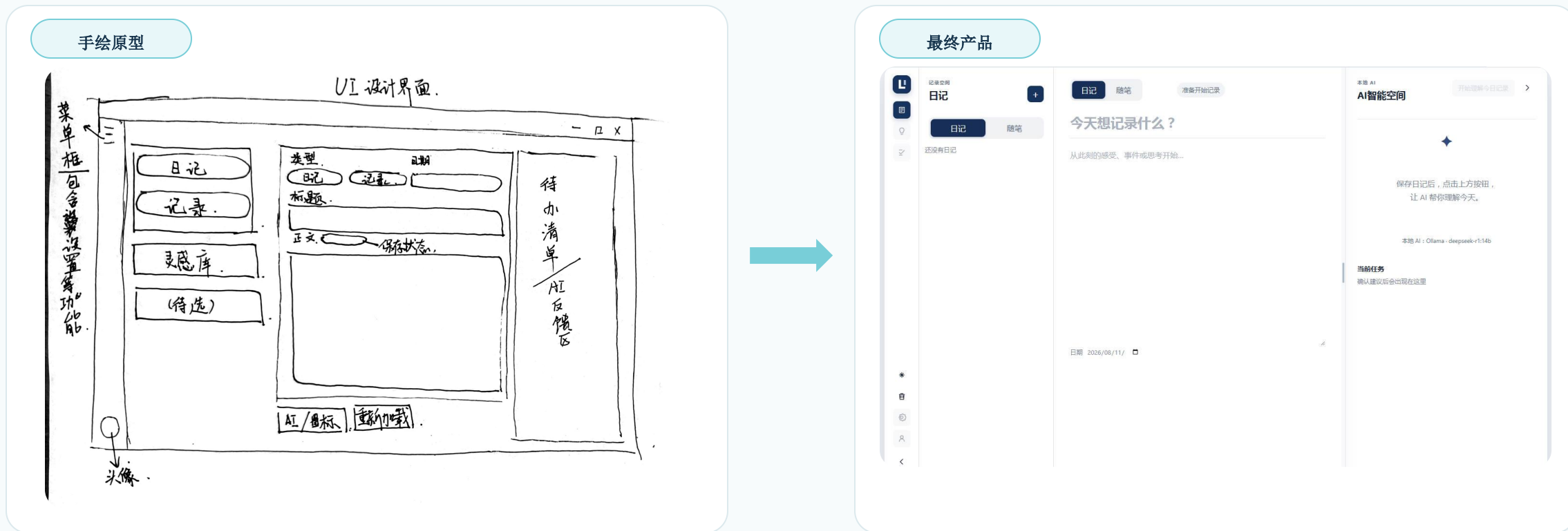
早期多端规划 →
Windows-first

优先完成数据可信、AI核心闭环、
Production Runtime与Windows真实交付。

不提前堆叠
Web / Mobile等功能

从手绘原型，到实际可运行产品

AI参与方案生成，但页面结构、需求方向与最终确认由Product Owner负责。



需求描述 → AI生成方案 → 人工修改与交互确认 → 进入实现 → 人工验收

Product Owner × AI Agent: AI协作开发方式

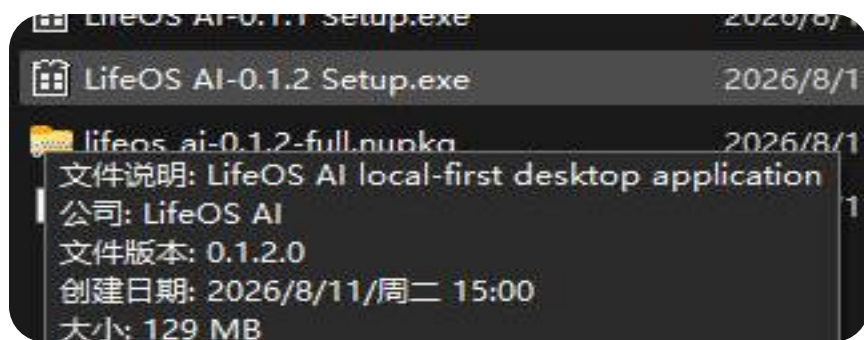
Decision ≠ Implementation ≠ Acceptance



Product Owner不是独立工程开发者：核心职责是定义要做什么、为什么做、优先做什么，以及最终是否通过产品验收。

v0.1.2: 从产品方案，到可安装 Windows MVP

WINDOWS MVP RELEASE COMPLETE



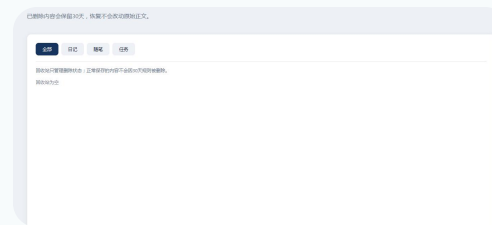
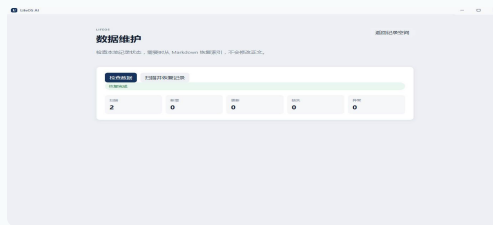
Installer



已安装应用



应用启动



数据维护 / 回收站

完成：Desktop记录与自动保存 · Markdown First · SQLite可恢复索引
本地Ollama理解 · Summary / Insights / Action Candidate · 用户确认创建Task
回收站 · Production Runtime · Installer · 升级与数据保留验收

已完成可安装的 **Windows MVP**，而非停留在
产品方案与原型阶段。

真实不足，与下一步判断

当前真实不足

- 01 本地Ollama分析等待时间较长，是目前最直接的体验问题。
- 02 当前尚未进行外部真实用户测试。
- 03 以下能力仍属于Deferred:
Web · Mobile · Cloud/API AI · Sync · Scheduler
Memory/RAG · 完整Task Space · 用户系统

vNext Product Prioritization

- Cloud/API AI是否优先解决响应速度
- Local / Cloud Data Trust Model
- Web Demo与低门槛外部体验入口
- 外部小范围体验测试
- 哪些能力真正值得进入下一版本

最大的产品收获：在有限时间内定义可信的核心闭环，并明确知道哪些功能现在不应该做。