MAgent 多智能体概念优化引擎（创始人视角·重写稿）

0. 执行摘要（我们做什么，为什么现在，效果如何）

我们把"从灵感到落地"的过程做成一款本地桌面产品：把反问、协作、验证、收敛变成标准化的流水线，让一个模糊的想法持续产出可执行的高质量方案。选择现在，是因为多模型生态成熟、端侧能力可用、而企业与个人都更加重视数据主权。我们以"多智能体协作 + 质量门控 + 本地优先"形成差异化，目标是把方案生产从"靠灵感"升级为"靠过程与证据"。

1. 我们解决的问题（Pain → Value）

问题清单：

- 想不清：目标/边界不明确，讨论高频但低效；
- 想不全：缺反驳、缺证据，方案"好看但不稳"；
- 难落地：从点子到计划的转译成本高，路径与风险不清；
- 数据顾虑：云端工具难纳入敏感资料，协作与复盘难落本地。

价值闭环：

- 将"反问—发散—互证—收敛—验证—输出"变成可控流程；
- 将证据、不确定性与风险与结论并列展示，便于稳健决策；
- 将数据与历史记录留在本地，实现可追溯与复用。

Why Now：多模型并行与互评成为现实，Tauri/Rust 使本地应用具备跨平台与高性能，两者叠加让"本地多智能体协作"具有可行性与性价比。

问题成因剖析（根因而非表象）

- 过程不可视：头脑风暴/大模型对话看似热闹，缺少"阶段定义""退出条件""质量阈值"，导致早收敛或无限拖延；
- 缺少互证：单一角色/单一模型容易被首因偏见与幻觉带偏，缺反驳与证据强度评估；
- 缺少可追溯：会议纪要/聊天记录难以复盘"为什么选这个"，难以复用到下一次；
- 安全与主权：云端工具难落地到涉密/内网，导致真正需要的人用不上。

典型场景对照（现状 → 用 MAgent 之后）

- 现状：20 分钟会议里 15 分钟在对齐问题定义，剩余 5 分钟抛结论无证据； 用 MAgent：澄清代理先跑 1–2 轮把"目标/边界/约束/成功指标"问清，并生成澄清要素表；
- 现状：两个方案各自站队、难以选择； 用 MAgent：创新者×批判者互评 2 轮，输出"支持点/反驳点/不确定性"，并给出收敛建议；
- 现状：方案落地要再花半天整理文档； 用 MAgent：工作区直接生成"目标、里程碑、风险、指标、依赖"的结构化可执行稿，支持导出。

2. 我们的解决方案（产品概述）

产品形态：跨平台桌面应用（Windows/macOS/Linux），前端 React，后端 Rust（Tauri）。 工作方式：

1. 想法输入：文本/文档，自动解析领域、复杂度与完整度，生成引导策略；
2. 智能反问（澄清）：针对目标、边界、约束逐步问清；
3. 协作优化（创新×批判）：多代理发散、互评、迭代；
4. 验证评估：事实核查、逻辑一致性与可行性评估；
5. 综合输出：形成"可执行方案"（目标、里程碑、依赖、风险、指标）。

设计目标：

- 初步解析响应目标：≤3 秒（视模型与网络而定）；
- 过程可解释、路径可追溯、产出可直接用于汇报或执行。

3. 核心竞争力（为什么我们能赢）

- 结构化推进：将"提问—发散—互证—收敛—验证"做成默认工艺，稳定产出；
- 质量门控：完整度阈值、轮次上限与置信度三重门控，防止过早或过度迭代；
- 证据与不确定性并陈：不遮蔽灰度，辅助稳健决策与后续调研；
- 本地数据主权：默认本地存储与版本管理，更适配敏感场景与内网环境；
- 多模型互评：同一阶段可启用多模型轮次与互评，降低偏置与波动；
- 工程可扩展：清晰的 agents/core/models/storage 边界，方便新增搜索、知识库、导出模板与私有模型；
- 性能与体验：Tauri + Rust 冷启动快、占用低，适合长流程与多项目并行。

WOW 清单（让评委眼前一亮）

- 状态机门控的"可控创意流水线"：每一步有进入/退出条件与质量阈值，创意不再失控；
- 多智能体互证：创新者×批判者×澄清者×验证者形成闭环，显著降低幻觉与拍脑袋；
- 证据与不确定性并列展示：把"证据来源、置信度、未决问题"与结论并排，方便评委追问；
- 本地优先的数据主权：默认 SQLite + 文件系统，敏感资料可内网/离线使用；
- 三分钟从"一句话想法"到"可执行方案"：目标、里程碑、风险、指标一次成稿；
- 可插拔模型与私有化：OpenAI/Anthropic/DeepSeek/私有模型一键切换，成本与可用性可控；
- 轻量高性能：Rust+Tauri 冷启动与资源占用显著优于 Electron 方案，适合长流程。

4. 产品能力与体验（输入→验证→可交付）

- 可视化流程：7 个核心页面串联过程，展示阶段、轮次、评分、阈值与进度；
- 过程资产化：澄清记录、互评结果、验证报告、最终方案可沉淀为项目资产；
- 可执行输出：面向执行的结构化产物（目标、里程碑、风险、指标、依赖）；
- 可配置：多模型供应商（OpenAI/Anthropic/DeepSeek）、网络代理与参数阈值；
- 可恢复：版本管理与历史回放，异常可恢复，支持断点续作。

示例结果（非涉密示例）：

- 输入"校园二手交易平台优化思路"，系统输出：
  - 澄清要素表（目标用户、约束与成功指标）；
  - 多方案清单（差异化策略 + 互评理由）；
  - 验证报告（证据来源、置信度与不确定点）；
  - 最终方案（3 个月里程碑、关键风险与度量指标）。

5. 技术方案与工程落地（架构/机制/取舍）

前端：React 18 + TypeScript + Vite + Tailwind CSS + React Router（过程可视与交互）。 容器：Tauri 2.x（跨平台桌面，前后端通信）。 后端：Rust + Tokio；依赖 reqwest(rustls)、sqlx(SQLite)、serde、anyhow/tracing、config/notify/handlebars。 数据：SQLite 本地持久化（项目/对话/证据/版本），文件系统用于导出与大对象。 AI：OpenAI / Anthropic / DeepSeek（config.toml 配置）；支持多模型轮次与互评；预留本地/私有模型与知识库接入。

关键机制：

- 状态机门控（src-tauri/core）：基于完整度阈值与最大轮次控制收敛；
- 代理契约（src-tauri/agents, models）：统一入参/出参，便于扩展新角色与模型；
- 可观测性：anyhow/tracing 记录上下文，便于定位与回放；
- 数据安全：默认本地优先，按需联网，支持企业代理配置。

设计取舍：

- 选择 Tauri + Rust：资源占用低、启动快，贴合长流程应用场景；
- 选择 SQLite：零运维、稳定可控，适合个人与小团队；
- 选择可插拔模型管理：适配多供应商与私有部署，保障连续可用与成本优化。

架构与数据流（从输入到可交付）

1. UI（`src/`）：`pages/QuestioningPage.tsx`、`DiscussionPage.tsx`、`WorkspacePage.tsx` 呈现阶段、轮次、评分与导出；
2. 桌面容器（Tauri）：前端通过 Tauri IPC 调用后端 Rust 命令；
3. 核心运行时（`src-tauri/core`）：
    - `state_machine.rs` 定义阶段（澄清→发散/互评→综合→验证→收敛→输出）与转换条件（完整度阈值、最大轮次、失败重试）；
    - `agent_runtime.rs` 负责在各阶段调度对应 Agent 并合并产物；
4. 多智能体（`src-tauri/agents`）：
    - `clarifier.rs` 澄清者、`innovator.rs/innovator_new.rs` 创新者、`critic.rs/critic_new.rs` 批判者、`synthesizer.rs` 综合者、`verifier.rs` 验证者；
5. 模型管理（`src-tauri/models`）：`manager.rs` 统一接入 OpenAI/Anthropic/DeepSeek 或私有模型；
6. 存储（`src-tauri/storage`）：`database.rs`（SQLite 项目/阶段/证据/版本）、`vector_store.rs`（向量检索预留）、`cache.rs`；
7. 配置（`src-tauri/config`）：读取 `config.toml`，含 API Key、代理与模型参数。

状态机门控（为什么能稳定产出）

- 进入条件：每一阶段检查输入上下文是否满足（目标已澄清、约束已收集等）；
- 退出条件：达到完整度阈值或迭代至最大轮次，否则继续；
- 失败恢复：阶段失败会重试/降级（切换模型、降低温度或缩短上下文），并记录日志；
- 可回放：每次收敛生成"版本点"，支持差异查看与回放。

Agent 契约（输入/输出约定，便于扩展）

输入：

- goal：目标/任务；
- context：澄清要素（边界、约束、成功指标、受众等）；
- history：历史对话与中间产物；
- params：模型/温度/轮次等运行参数。

输出：

- artifacts：结构化产出（澄清要素表/方案清单/收敛建议/验证报告等）；
- evidence：证据与来源；
- uncertainties：未决问题与风险；
- scores：完整度/一致性/可行性等评分；
- next_hint：建议的下一步动作。

简化伪代码（对应 `core/agent_runtime.rs` + `agents/*.rs`）：

```
for stage in pipeline:
  ensure(state_ready(stage, ctx))
  for round in 1..=max_rounds(stage):
    outputs = parallel_run_agents(stage.agents, ctx)
    merged = synthesize(outputs)
    if quality_gate_pass(merged, threshold(stage)): break
  persist_version(stage, merged)
```

可观测与可恢复（工程级保障）

- tracing：贯穿阶段/轮次/请求/错误；
- sqlite：本地事务化持久化，崩溃可恢复；
- 重跑策略：允许"仅重跑本阶段"，避免前功尽弃；
- 安全：默认本地，联网仅在调用外部模型时发生，支持企业代理。

6. 使用与部署（Windows / PowerShell）

1. 环境准备：Node.js 18+、Rust 稳定版、@tauri-apps/cli。
2. 安装依赖：

```
npm install
```

3. 配置模型：

```
cp config.example.toml config.toml
# 在 config.toml 中填入可用 API Key（可配置 HTTP/HTTPS 代理）
```

4. 启动开发：

```
npm run tauri:dev
```

5. 构建安装包：

```
npm run tauri:build
```

常见问题：

- 未配置 API Key 导致部分阶段无响应；
- 企业网络需配置代理；
- 首次构建 Rust 依赖较慢，属正常现象。

7. 应用场景与价值（谁在用，用来解决什么）

- 产品与增长：在有限时间内产出"可信 + 可执行"的备选方案与评审材料；
- 活动策划与运营：将创意快速落成可执行脚本与风险清单；
- 科研与学习：用于课题方向澄清、路线规划与资料核对；
- 内部流程优化：以证据与不确定性并陈的方式推动稳健改造。

8. 竞争格局与差异化（我们与替代方案）

- 单一大模型对话：难以保证结构化推进与可验证；我们强调角色互证与质量门控。
- 思维导图/头脑风暴：偏发散缺收敛与验证；我们默认"发散→验证→收敛→落地"。
- 云端 SaaS 方案工具：数据在云端、敏感资料受限；我们默认本地数据主权与可回溯。

9. 发展路线图（把想象力落到里程碑）

- 证据溯源图谱：将验证证据自动汇成可交互图谱，支持一键追根溯源；
- 私有知识库连接器：对接本地/内网文档与数据库，统一检索与引用；
- 离线小模型支持：在弱网或离线环境下维持基础反问/验证能力；
- 多方案 Pareto 收敛：以多目标优化思路收敛更"均衡"的解；
- 模板与导出集市：一键导出 Markdown/PDF/PPT，沉淀行业模板；
- 团队协作：项目协作权限、审计追踪与可重放的评审脚本。

10. 商业模式与增长（简述）

- 路线：个人/小团队（社区版）→ 专业版（模板/导出/本地知识库）→ 企业版（私有模型/内网连接器/审计追踪）；
- 价值：把"方案生产"变成可复用的过程资产，持续复利。

11. 结语与承诺

我们相信，好的方案不是"灵感的偶然"，而是"过程的必然"。MAgent 让这个过程标准化、可追溯、可验证，并且掌握在使用者手里。这就是我们的产品边界与承诺。

12. 快速演示指南（3 分钟）

- 场景：将"校园二手交易平台优化思路"转为可执行方案。
- 入口：主页 → "新建项目"。

1. 输入想法：在"想法输入页"粘贴一句话目标；点击"开始"。
2. 智能反问：进入"澄清页（Questioning）"，连续确认目标、边界、约束与成功指标；预计 1–2 轮。

3. 多代理发散：切换到"讨论页（Discussion）"，启用"创新者 × 批判者"并行 2 轮；观察互评分与理由。
4. 验证与收敛：点击"验证"，生成证据与不确定点；随后在"收敛"中选择最终方案。
5. 输出与沉淀：在"工作区（Workspace）"查看结构化方案（目标、里程碑、风险、指标、依赖），导出 Markdown/JSON。

演示提示：

- 任何阶段都可"暂停/继续/重跑本阶段"；异常可在"历史版本"回滚；
- 证据来源、置信度与未决问题与结论并列展示，便于评委追问与复盘。

13. 性能与成本基线（方法与目标）

说明：性能/成本随模型与网络波动，本节给出"测量方法 + 目标区间"；建议在评审机位现场执行 3–5 次取中位数。

测量方法（示例）：

- 硬件/网络：记录 CPU/内存/网络（例如 i7/16GB/100Mbps）；
- 模型配置：在 `config.toml` 指定同一供应商与型号；
- 任务脚本：固定同一输入，运行澄清 2 轮 + 发散互评 2 轮 + 验证 1 次；
- 采样口径：记录阶段用时、总用时、失败重试次数、请求 Token 量（若可得）。

目标区间（建议上限，非承诺值）：

- 初步解析：≤ 3s；
- 单轮澄清：≤ 20s/轮；
- 发散互评（2 代理×2 轮）：≤ 90s；
- 验证报告：≤ 30s；
- 全流程（示例脚本）：≤ 3 分钟；
- 成本目标：≤ $0.20/完整流程（随模型计费而变动）。

注：若在弱网/内网环境，可启用代理与更轻量型号；长文本或复杂任务会增加时延与成本。

14. 导出、集成与流程控制

- 导出格式（当前）：Markdown、JSON 结构（包含目标、里程碑、风险、指标、证据、不确定点）；
- 导出格式（路线图）：PDF/PPT 模板化导出（参见第 9 节"模板与导出集市"）；
- 流程控制：任意阶段支持"暂停/继续/重跑本阶段/设定最大轮次/设定完整度阈值"；
- 版本与回放：每次收敛都会生成版本点，可在"项目历史"查看差异与回放；
- 集成点：预留本地/私有模型与知识库连接器，支持企业代理与离线模式（基础能力）。

15. 与替代方案对比（摘要）

维度	单一大模型对话	思维导图/脑暴	云端 SaaS 方案工具	MAgent
结构化推进	弱	中	中	强（状态机门控）

维度	单一大模型对话	思维导图/脑暴	云端 SaaS 方案工具	MAgent
证据与不确定性	弱	弱	中	强（并列呈现）
数据主权	中	强	弱	强（本地优先）
多模型互评	弱	无	因厂而异	强（可并行轮次）
可扩展/可回放	弱	中	因厂而异	强（清晰分层）
完整对比见：docs/对比表-一页版.md。				

16. 评审 FAQ（摘录）

Q1：企业/高校内网如何使用？

- A：默认本地优先，支持 HTTP/HTTPS 代理；可接入私有模型与知识库。弱网下提供轻量模型与降级路径。

Q2：隐私与合规如何保障？

- A：数据保存在本地 SQLite 与文件系统；联网仅在调用外部模型时发生，支持审计日志与按需脱敏。

Q3：为什么选择多智能体而非单 Agent 提示词工程？

- A：多角色互评可降低偏置与幻觉，配合"完整度阈值 + 最大轮次"门控，稳定产出可执行方案。

Q4：失败或偏题如何恢复？

- A：阶段可重跑，项目支持版本回滚；可调整阈值与轮次，或切换模型以获得稳健结果。

Q5：是否支持离线？

- A：基础功能可离线；涉及外部大模型推理的阶段需可用的本地/私有推理服务或联网。

更多问答见：docs/评审FAQ.md。

17. 评审素材与截图位（可替换示例）

- 架构与状态机示意（占位）：docs/assets/architecture.png、docs/assets/state-machine.png；
- Demo 路径截图（占位）：输入 → 澄清 → 讨论/互评 → 验证 → 工作区导出；
- 对比矩阵与一页版链接：docs/对比表-一页版.md、docs/评审速览-一页版.md。