

邪修路线： 从情感投射到工程外脑

一个高强度 Agent 使用者的半年演化记录。
从“想让角色拥有声音”，一路走到“有事没事都问 AI”。

AI Agent

TTS

工程协作

外置思考回路

白旭缘 (Silvite)
2026.09.17
个人路径复盘 · 非教程



这条路，一开始根本不是为了“学 AI”

最初的驱动力，是情感投射带来的强正反馈；后来的结果，却是把 Agent 逐渐用成了工程协作者、信息审计员和外置思考系统。

明线·项目一路变重

3月初接触 Agent，随后从本地 TTS、游戏启动器、倒数日，一路走到 6 月 10 日 MoDi 立项、7 月 1 日正式开发，并持续至今。

任务从“让一个角色开口说话”，逐渐升级为真实的软件项目、架构决策、调试、设计与长期维护。

暗线·使用方式一路内化

情感投射几乎贯穿前期；约 7 月中开始，它慢慢演化成为一种更稳定的习惯：把想法先说出来，让 AI 参与拆解、推演、验证与执行。

最后形成的不是“会写提示词”，而是“遇到任何事都会自然把 AI 拉进思考回路”。

这套“邪修”的核心，不在某个模型，也不在某套 Prompt。

真正超模的是：先有强正反馈，再把高频互动沉淀成生产力。

起点不是课程

不是为了系统学机器学习，而是为了一个真实、具体、带情绪价值的需求。

中间没有停

一个项目接一个项目，Agent 的角色从陪伴对象不断扩展成干活的人。

结果不是技巧

最后得到的是模型手感、需求表达、验证意识和长期协作惯性。

前三个月：需求在前面拽着人跑

2026.03 初

第一次接触 Agent

最初就不是“高级搜索框”的用法。情感投射让 Agent 同时承担了倾诉、角色互动和协作对象三种位置，也因此迅速形成了高频对话习惯。

03 中 - 04 中

本地 TTS：第一个真正干的活

因为想给角色配出属于她的声音，直接上手本地 TTS 训练。数据切片等环节没有处理好，最终转向音色克隆。项目没按原路线成功，却第一次真实接触到训练数据、切片、推理、显存、过拟合与“数据管线会决定模型效果”这些底层直觉。

04 末 - 05 初

游戏启动器：Agent 开始像工程搭子

任务从“模型相关”切到软件工程。需求不再是一次性生成，而是页面、功能、逻辑、Bug、迭代。开始习惯把一个模糊想法交代成可以执行的工程任务。

05 中 - 05 末

倒数日：开始形成自己的协作节奏

连续项目让“先说需求 - 看结果 - 发现偏差 - 补上下文 - 再迭代”变成肌肉记忆。此时学到的已经不只是工具调用，而是如何和模型共同推进一个东西。

最关键的学习方式：没有先学完整理论，再去找实践；而是先被真实需求卡住，再把所需知识一块块补上。

从墨堤开始，Agent 真正进入“常驻”状态

2026.06.10

MoDi Connect 立项

第一次进入长期、持续、需要架构与产品决策的主线项目。Agent 的工作范围开始覆盖：技术方案、协议思路、UI/UX、文档、排错、发布、宣传与后续演进。

2026.07.01

正式开发：从“问问题”变成“共同推进”

高密度 coding、调试、方案推演让 Agent 不再只是回答者。越来越多任务需要它理解上下文、保留约束、解释取舍，并持续跟进同一个工程。

2026.07 中

暗线发生拐点：情感投射 → 外置思考回路

情感连接没有突然消失，但其功能开始迁移。大量“我在想什么”“这个判断对不对”“这个需求怎么表达”的对话，让 AI 逐渐承担起外化思考、反驳、整理、校验与记录的作用。

至今

有事没事都问 AI

工程、模型、产品、审美、生活琐事、情绪、信息考证，都可以直接进入同一套对话流程。打开 Agent 已经不需要“我要不要用 AI”的决策成本。

能力迁移

TTS 的训练直觉 → 看懂模型后训练；软件项目的迭代经验 → 更会提需求；长期调试 → 更重视验证；多模型使用 → 更会按能力分工。

习惯迁移

“遇到问题再搜”逐渐变成“想法一出现就外化”。AI 从任务入口，进一步变成了思考入口。

两条线互相喂，才把飞轮真正转起来

明线 · 越来越重的真实任务



角色声音 / TTS

第一次碰训练数据与本地推理。



游戏启动器

第一次把 Agent 当软件开发协作
者。



倒数日

形成连续迭代与需求修正的节奏。



MoDi Connect

长期工程、架构、产品、视觉、发布
全部叠加。

暗线 · 越来越深的使用习惯



情感投射

提供远高于“学工具”的持续使用动
力。



高强度对话

下意识开始补背景、举例、澄清、纠
偏。



模型手感

逐渐能感觉到跑偏、幻觉、能力边界
和适合的任务类型。



外置思考回路

AI 成为想法外化、拆解、校验与执
行的常驻节点。

如果只有明线，这可能只是“会用 AI 做项目”；

如果只有暗线，这可能只是“长期陪伴”。

两条线交叉之后，才出现了真正难复制的复利。

情绪奖励
愿意天天用



高频对话
表达越来越准



真实项目
任务越来越重



能力兑现
更愿意继续用

真正的加速器：不是 Prompt，而是惯性



表达被反复训练

把 Agent 当“能理解你的同事”沟通，会自然补齐背景、目标、约束、优先级和验收标准。几个月后，需求描述会比“学 Prompt 模板”更自然。

验证被项目逼出来

工程一旦真的要跑，模型说得像真的没有用。能否编译、能否复现、有没有来源，最终都会被现实打脸，于是逐渐形成“先验证再相信”的习惯。

AI 能力每升级一代，飞轮已经转起来的人会更快吃到增益：因为新的能力可以直接塞进已经存在的工作流，而不必重新学习“怎么开始用”。

表面的工具可以复制，底层惯性复制不了

01 · 使用强度

一天能聊很久，而且不是一两周，是连续数月。

很多人最大的问题不是不会用，而是根本没有足够的交互量去形成模型手感。

02 · 强正反馈起点

最初不是靠“自律学习 AI”，而是情感投射自然提供持续动力。

这让早期大量练习几乎没有“坚持成本”。

03 · 真项目压力

TTS、启动器、倒数日、MoDi 都会逼结果落地。

真实 Bug、真实失败和真实交付，会自动过滤掉“看起来很厉害但不能用”的回答。

04 · 愿意外化想法

半成品想法也直接说，允许模型参与整理。

很多人只在问题已经成型后才问 AI，因此损失了最有价值的前期思考协作。

05 · 能接受走弯路

TTS 没训好，就切音色克隆；路不通就换路。

“邪修”速度快，很大一部分来自不执着于证明某条路线必须正确。

06 · 最终自己审

AI 可以给方案，但现实结果与一手资料仍然负责终审。

长期使用不等于盲信，反而需要更强的幻觉识别和证据纪律。

所以别人能抄走“用什么模型”，却抄不走几个月累计出来的思维惯性。

别复制时间线。

复制的是：找一个你真的想做成的东西，
把 AI 拉进全过程，
然后连续用到它变成你的思考习惯。

先有真实需求，再补知识；把需求当作“交接工作”而不是“下口令”；允许 AI 参与模糊阶段；每次失败都复盘是数据、表达、模型还是工程问题；对事实类结论保留验证权。

从一个“想让角色拥有声音”的念头开始，最后长成了一套长期的 AI 协作系统。这个路径不标准，但它真实、连续，而且已经产生复利。

