THE FIELD DOCTOR · FDE

现场医生

中国 FDE 的崛起与实战密码

大模型像药厂，FDE 像医生。AI 落地最稀缺的，是愿意走进现场的人。

付伟　著

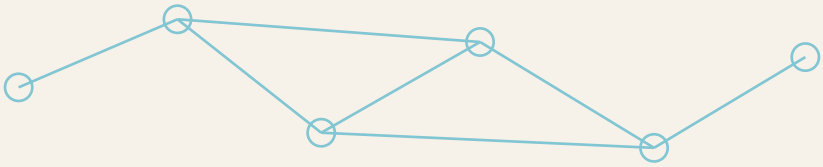
艺术排版预览版 · 2026 年 6 月

目录

第一篇

为什么

AI 落地为什么这么难



治病的第一步，是先承认病在哪里。

INTRODUCTION

引言 大模型越强，越需要现场医生

> AI 永远没有交付完毕的那一刻。
>
> ——高利明

模型会越来越强，但企业现场不会自动变简单。

这是我写这本书最想说的一句话。把它放在最前面，是因为接下来所有的篇章，都是在为这句话做注脚。

几乎前后脚，两家大模型公司做了同一件事

2026 年 5 月，两家做基础模型的公司，几乎前后脚宣布了重大战略布局。

OpenAI 推动成立了一家由它主导的独立部署公司，承诺资本超过 40 亿美元，由 TPG 领投，同时收购了一家位于英国爱丁堡的 AI 咨询与工程公司，把大约 150 名工程师整建制并入。几乎在同一个窗口，Anthropic 宣布成立一家承诺总资本 15 亿美元的合资公司，背后站着黑石、Hellman & Friedman、高盛、红杉一长串名字，并收购了旧金山一家叫 Fractional AI 的应用团队，作为这家新公司的运营内核。

请注意，它们做的不是发布新模型，不是给 API 降价。

是把人派到客户现场去。

两家公司的动作在时间上高度重叠，很难说是巧合。更合理的解释是：它们同时感觉到了同一个市场窗口——客户买完 API 之后，用不起来。模型能力在变强，但企业离真正用出效果，还隔着一段不近的距离。这段距离，一直没有人认真去走。

于是它们决定，自己来走。

市场对这件事的反应，比任何分析都直接。就在那前后，软件板块经历了一轮剧烈的重新定价。据公开报道，企业级 AI 工具的发布曾让一家老牌信息服务巨头单日大跌、标普软件指数连续多日下挫，到两家公司大举派驻现场团队的那几天，几家头部 IT 服务公司股价同步走弱，相关 IT 指数创下多年最差。有投行在报告里写得很直白：传统 IT 服务商的高利润模式，正在被“AI 原生团队”侵蚀。

逻辑其实不复杂：如果有外部团队能直接帮企业把 AI 嵌进工作流、在现场交付确定的业务结果，那么每年花几百万买一套需要自己研究怎么用的软件，还划算吗？

市场在重新给“卖软件”这门生意定价。而被它重新定价的另一面，是一种新的角色正在浮出水面。

我是怎么撞上这件事的

我不是一开始就盯着这个趋势的。我是先掉进坑里，才慢慢看清坑的形状。

2025 年 3 月，我带着团队启动了一个项目：在一家机构里，把大语言模型本地化部署起来，整合二十多年积累的专业资料建知识库，做研究报告生成、投资分析辅助、文档智能检索。本来以为三个月就能上线。结果用了大半年才正式跑起来，而且项目远没有结束——新的需求不断冒出来，新的子项目不断衍生出来。

这个过程里，我们受了各种各样的折磨，越过了很多沟和坎，也有不少欣喜和欣慰。踩过的坑一个接一个：选模型踩坑，国产 GPU 适配踩坑，最扎心的一个坑是——系统千辛万苦跑起来了，上线三个月却日活寥寥，同事们宁可去问人，也不愿打开它。到底有多冷清、为什么会这样，我留到第 1 章再细说；这里只先记住这个画面。

那一刻我盯着后台那条贴着地面的曲线，心里冒出一个问题：明明模型够强、工具够全、钱也花了，为什么就是用不起来？

差不多同一时间，我读到认识二十多年的老朋友——投资人高利明的一篇文章。他有一句话像钉子一样钉进了我脑子里：

> AI 永远没有交付完毕的那一刻。

后来我反复琢磨这句话，发现它不只是说“AI 项目需要持续优化”——那不新鲜，所有软件都需要优化。它指向的是一件更深的事：AI 的能力不是一次性封装好的，它是在使用中长出来的。模型每个月在变，业务每个月在变，组织每个月在变。它不是一个“做完”的项目，而是一个需要有人一直陪着、一直调的“活物”。高利明还打过另一个比方：传统软件像“宏”，AI 服务像“宏生成器”——前者交付的是确定的功能，后者交付的是持续演进的可能性。

我突然意识到，我们一直在用做软件的方式做 AI——以为做完、上线、验收，就算交付完了。可 AI 不是这样的东西。

我开始意识到，AI 落地最稀缺的，可能不是模型，也不是工具，而是连接模型与真实业务现场的那条神经通路。

后来我才知道，这条神经通路，国外早有人给它起了名字——Forward Deployed Engineer，前沿部署工程师，简称 FDE。一个二十年前在战场上被逼出来的角色，正在 AI 时代重新成为最关键的交付岗位。

这些年，我自己在一线踩坑、跑项目，也和一群同路人不断交流；尤其是我的老朋友陈澄波，他在 FDE 这条路上的实践给了我很多启发，本书里好几个案例也来自他。项目推进的过程中，我把很多体感写成了文章，发在了自己的公众号上；更多素材躺在了得到大脑的笔记里，等着被重新翻出来。加上得到新出了写作专家团的 Agent 可以辅助创作，几件事凑在一起，我越想越觉得，这件事值得认真写一本书。

这本书，就是这么来的。

一个贯穿全书的比方：药厂与医生

为了让后面的内容好读，我先把一个比方立在这里，它会一直陪着我们走到最后一页。

大模型公司，像药厂。

药厂负责研发药物、生产标准化的通用能力。这种能力一旦成熟，就会被快速复制、压低价格、变成像水电一样的基础设施。药很重要，但药厂不会上门给你看病。

FDE，像医生。

医生不生产药。但医生知道病人哪里痛、病史是什么、有什么禁忌、药该怎么配、疗效怎么观察、方案什么时候要调整。

企业，就是那个病人。很多企业不是没有 AI 需求，而是不知道该怎么描述自己的 AI 需求。老板说想降本，现场可能卡在排产；业务说想提效，真正的问题可能在数据口径；部门说想要智能客服，根子可能是知识库没人维护。

如果只把一个大模型账号交给企业，就像把一箱药扔给病人：药可能很好，但病不一定能治好。硅谷投资机构 a16z 有一个更损但很贴切的说法——企业买 AI，有点像奶奶拿到一部 iPhone，她当然想用，但需要有人帮她设置好、手把手教她怎么真正用起来。

模型越强，这个比方就越成立。因为药越强，越需要会用药的人；现场越复杂，越需要愿意走进现场的人。

这本书要回答的，就是三个问题：

第一，为什么 AI 越强，企业反而越用不起来？这条鸿沟到底卡在哪里？

第二，谁来填这条鸿沟，他到底在做什么，凭什么值钱？

第三，一个普通人——哪怕没有技术背景——怎么抓住这个机会？

这本书怎么读

这本书分成七个部分。

前两部分讲“为什么”：AI 落地为什么这么难，这门生意的逻辑为什么正在被重写。这部分有故事、有数据，是给所有关心 AI 的人看的。

中间几部分讲“怎么做”和“现场长什么样”：我会把自己亲手跑过的项目摊开来讲——本地化部署的坑、模型怎么选、GPU 怎么适配、系统上线后怎么从没人用变成有人用；也会把几个真实的脱敏案例，一个一个拆开给你看。这部分是这本书的硬核，也是它和市面上那些隔岸观火的 AI 趋势文最不一样的地方。我写进书里的做法，都是在真实项目里跑通过的；我写进书里的判断，都会尽量说清它的适用范围和边界。

后几部分讲“谁能做”“能做多大”和“往哪里走”：什么样的人能成为现场医生，这条路有几种活法，投资人怎么看，以及它对一个普通人、一个组织、乃至这个时代，意味着什么。

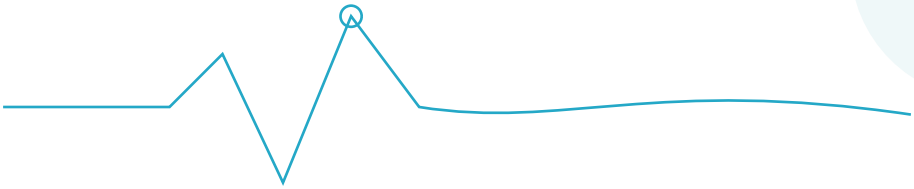
你不需要懂代码。事实上，这本书里有一个反复出现的主角——我的老朋友陈澄波，一个做了十几年人才发展、一行代码不会写的非技术老兵——他用两个月做出了一套支持一万人使用的企业级系统。如果你担心自己“太外行”，请记住：这本书想证明的恰恰是，那道曾经只属于技术人员的门，已经开了。

模型层很耀眼。但在整条 AI 产业链里，应用层最稀缺的，可能正是这群愿意走进现场、把药真正喂进病人嘴里、并且守着他吃下去、看着他好起来的人。

AI 时代真正值钱的，不是喊“我要做 AI”。

而是你能不能把 AI 送进一个真实的企业，让它留下来，跑起来，并且持续变好。

这本书，写的就是这群人，和这件事。



CHAPTER 01

第 1 章 88% 的企业花了钱，却没听到响

> 买得到的是模型，买不到的是把模型用起来的人。
>
> ——付伟

主诉 本章要回答：为什么 88% 的企业用了 AI，只有 6% 真正用出了结果？

先看三组数据。它们各自来看都不稀奇，但放在一起，会拼出一幅让人心里一紧的图景。

第一组来自麦肯锡 2025 年的全球 AI 调研。它问了全球大量企业一个问题：你们用 AI 了吗？88% 的企业回答，已经在至少一个业务职能里用上了 AI。这个数字很漂亮，几乎可以宣布“全员上 AI”了。

但同一份调研里还有第二个数字：在这些用了 AI 的企业里，能说出“AI 对公司利润（EBIT）产生了正向影响”的，只有 39%——而且大多数影响幅度不到 5%。再往里看，真正能被定义为“高绩效者”、靠 AI 实打实把利润做出 5% 以上增量的组织，大约只有 6%。

从 88%，到 39%，再到 6%。这是一条断崖式下跌的曲线——与其说是漏斗，不如说是瀑布：绝大多数企业的 AI 预算，都顺着这道瀑布流进了下水道。用上 AI 的有一大片，真正用出钱的，只剩一小撮。

第二组来自 Gartner。它的判断更冷一点：到 2025 年底，**至少 50%** 的生成式 AI 项目，会在概念验证（PoC）之后被放弃。也就是说，钱花了，试点做了，Demo 跑通了，然后呢？然后就没有然后了，系统没能真正上线。试点汇报会上热热闹闹，PPT 做得漂漂亮亮，散会之后，日常工作还是老样子。

第三组来自斯坦福 HAI 的《2026 AI Index》。2025 年，全球生成式 AI 的私人投资规模高达约 1708 亿美元，同比增长超过 200%，几乎拿走了全部私人 AI 投资的一半。资本像潮水一样往上游涌，热度丝毫没降。

把这三组数字摆在一起，一个尴尬的画面就出来了：

钱在疯狂涌进来，工具在飞速变强，用 AI 的企业越来越多——但真正用出结果的比例，几乎没有同步提升。

投入和产出之间，裂开了一道大口子。

问题不在供给端

很多人下意识地会把这道口子归因到技术：是不是模型还不够强？是不是工具还不够好？再等等，等下一代模型出来就好了。

但请注意第一组数据的细节：**88% 的企业都已经用上 AI 了，问题不是它们买不到、用不上模型。** 模型就在那里，账号也开了，钱也付了。供给端其实已经相当充分。

如果问题真的出在模型不够强，那为什么同样的模型，6% 的企业能做出 5% 以上的利润增量，而剩下的大多数只能做出不到 5%、甚至什么都没有？

同一把药，有人吃好了，有人吃了没反应。差别显然不在药本身。

我们不妨看一个正面的极端。OpenAI 派 FDE 团队深扎进摩根士丹利之后，据公开报道，相关工作流的采用率做到了约 98%，效率提升 20% 到 50%。同样的底层模型，为什么在摩根士丹利能跑出 98%，在别处却日活寥寥？差的不是模型，是那支愿意“深扎进去”的团队。

这正是这本书要反复回到的那个原点：AI 落地的瓶颈，早就从“造不造得出来”，转移到了“用不用得起来”。供给端的仗已经打得差不多了，胜负手挪到了应用端、挪到了现场。

那么，现场到底卡在哪儿？

RAND 的判断：八成是组织问题

兰德公司（RAND）做过一项研究，访谈了大量资深从业者，想搞清楚企业 AI 项目到底为什么失败。它给出的结论，可能会让很多技术团队意外：

纯技术问题，大约只占两成。剩下的八成，是组织层面的问题。

八成。这个比例值得停下来想一想。

它意味着，当一个 AI 项目失败时，十有八九不是因为模型不行、代码写错、算法不够先进，而是因为——人、流程、利益、习惯这些“软”的东西没有理顺。

我自己撞过这堵墙，撞得很实。

一个上线了却没人用的系统

2025 年 3 月，我们启动了一个智能投资平台的建设项目，核心是把大语言模型本地化部署起来：整合团队二十多年的专业资料，建知识库，做研究报告生成、投资分析辅助、文档智能检索。

一年多下来，账面上的成绩单很好看：完成了数万份资料的治理，搭起了上百个专业知识库，开发了 50 多个场景化的智能体。从技术交付的角度，这是一个相当扎实的成果。

然后，项目启动大约三个月，我们拿到了第一份使用数据。

日活寥寥。 同事们更愿意去问人，而不是打开我们的 AI 系统。几十个智能体里，真正有人每天还在持续用的，只有两个。

那一刻，我对着后台那条贴着地面的曲线，第一次真切地理解了那 88% 和那 6% 之间的距离，到底是一段多长的路。我们造出了药，甚至造了一柜子药，但病人没把它吃下去。

为什么没人用？背后的原因，我放在第十二章专门讲——那里才是“日活寥寥”这个故事真正值得展开的地方。这里先说一个结论：几乎每一条原因，都印证了 RAND 那句“八成是组织问题”。

钱花了，账却算不明白

把现场再放大一点看，你会看到一幅更普遍的图景。这幅图景，凡是推过 AI 项目的人，多半都眼熟：

数据散落在三四个互不相通的系统里，光是把它们拼到一起就足以耗尽团队的精力；核心业务流程从来没有被正式描述过，全在几个老员工的脑子里，他们一忙、一休假、一离职，流程就跟着断片；一线员工怕被 AI 替代，表面配合，暗地里能拖就拖；IT 部门的排期已经排到三个月之后，你的需求得乖乖排队；分管领导上周还说要“全面 AI 化”，这周出了个小错，又觉得不如多招两个人靠谱。

在这种环境里，模型升级一次，能解决上面哪一条？

一条都解决不了。

因为这些问题不在模型里，在模型和现实之间那段没人走的路上。

技术问题，是有解法的——模型不够强可以换、上下文不够长可以扩、检索不准可以调。但人的习惯、人的利益、人的不安全感，没有快捷键。你没法给一个怕丢工作的员工“一键升级”成拥抱 AI，也没法给一个扯了三个月皮的跨部门流程“重启”一下就理顺。

这就是为什么单纯卖 API 走不通。技术到了，组织没准备好。模型公司把最好的药递过来，可病人这边——身体里的管路是堵的，吃药的人是抵触的，连哪儿疼都说不清楚。

而这“八成的组织问题”，恰恰不是大模型公司擅长、也不是它们愿意天天去趟的地方。大模型公司习惯了在云端做干净、宏大、性感的底层革命，而现场恰恰

相反——它太脏、太碎、太不性感。它需要有人真的走进那间安静的会议室，坐到一线员工旁边，看他到底卡在哪儿。

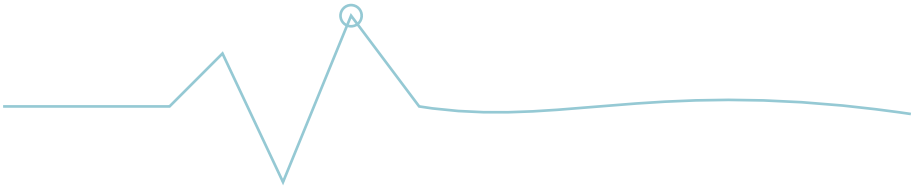
这个愿意走进去的人，就是这本书的主角。

小结

这一章，我只想先把那道裂口摆到你面前：

钱在涌进来，工具在变强，88% 的企业都用上了 AI，但只有 6% 真正用出了利润；一半的项目活不过概念验证；而失败里有八成，根子在组织，不在技术。

我自己那套技术上挑不出毛病、却三个月无人问津的系统，就是这道裂口的一个缩影。



FIELD NOTES

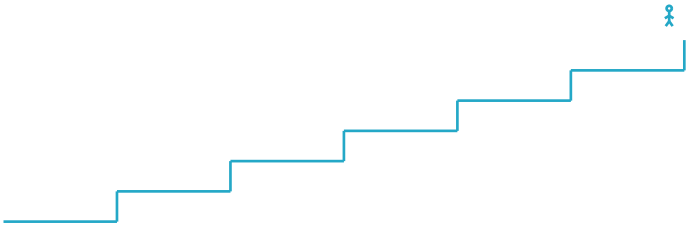
医嘱

01 别问“模型够不够强”，先问“有没有人把业务说清楚”。

02 落地失败，八成是组织问题，技术只占两成。

03 衡量 AI 项目，不看上线率，看采用率。

CHAPTER 01 · END



CHAPTER 02

第 2 章 AI 转型真正的摩擦力

> AI 出错不可怕，可怕的是改它的错，比自己重做还累。
>
> ——付伟

主诉 本章要回答：AI 落地真正的阻力，到底卡在哪一个具体的点上？

上一章说，AI 落地的失败，八成是组织问题。这一章我想把这“八成”再往下钻一层，钻到一个具体得不能再具体的点上。

因为“组织问题”四个字太大了，大到没法下手。你跟一个老板说“你们公司的组织没准备好”，他只会反问你一句：那到底是哪儿没准备好？

我自己趟过来，找到的那个最锋利的答案是四个字：

纠错成本。

不是不会做，是做错之后没人愿意收拾

前段时间，我请了高风咨询的两位专家给团队做分享。董事总经理 Vicki Wu 讲金融行业的 AI 转型路径，核心意思可以用四句话概括：从业务域而非点状需求开始推动，域要够大才能体现价值；用白纸视角重塑流程，不要只在现有流程上做优化；核心是业务知识萃取而不是技术，AI 降低了技术门槛，业务深度才决定价值；以 AI 为核心、人做补充，不需要追求全流程自动化。谢祖墀博士也强调，AI 引发的不只是技术更新，而是对经营范式、组织架构和管理哲学的系统性挑战。

每一条都对，方向都挑不出毛病。

但我注意到，台下坐着的业务骨干，脸上是另一种表情。散会后我私下问了一句，有人说了大实话：

“道理都懂。可 AI 出的错，我查起来比自己从头做还累。”

这句话，比任何转型框架都更接近现场的真相。

它不是抵触，不是焦虑，更不是“不懂 AI”。它是一笔朴素到不能再朴素的成本账：我用 AI 写一份报告，它幻觉了三个数据，我得花半小时去查、去改；而我自己写，可能二十分钟就完了。那我凭什么换？

这就是 AI 转型真正的摩擦力。不是技术能力不够，不是员工心理抵触，而是纠错成本。**谁为 AI 的错误买单，谁就没有动力用 AI。**

管理层看到的，是试点、流程、治理、架构、效率指标这些“大词”。一线员工看到的，是另外三件特别小、特别实的事：这次错了吗？错在哪里？我要花多久才能查清楚？

类似的场景在企业里太常见了：AI 几秒钟就能起草一份合同、一份纪要、一张分析表，可它后面还得排进法务、风控、业务专家的人工审核队列。前面省下的是几分钟，后面还回去的，可能是几小时甚至几天。

所以问题从来不是“AI 有没有价值”。问题是：**AI 的错误，是不是可定位、可复核、可收敛。** 过不了这条线，AI 就是负效率——它只是把人的工作从“生产内容”换成了“检查内容”，员工当然会默默退回老方法。

更麻烦的是，纠错成本不是一次性的。大模型的幻觉是概率性的，今天对了，明天可能错。你不是查一次就一劳永逸，而是每一次都得重新确认。这种“永远不知道这回要不要改、坑藏在哪儿”的不确定性，带来的心理负担，比多花半小时更重。

别等模型变好，先把查错方式变轻

那怎么办？等大模型自己变好？

这是大多数人的选择，也不是全无道理——模型确实在快速迭代，幻觉率也在下降。但问题是，等你等到那一天，先跑的人早就把流程蹚通了、把认知差拉开了。

这里有一句话，我想请你记住，后面还会反复用到：**AI 是放大器，不是平衡器。** 目标清晰的人会被放大，流程清楚的团队会被放大；目标混乱的人，连混乱也会被一起放大。等的人，不是在等技术成熟，是在等别人把路蹚出来、把差距拉开。

我自己在团队里推“数字员工”，一开始也踩在这个坑里。

最早让 AI 出一份行业简报，它幻觉了两个数据，同事花了四十分钟查证，结论很干脆：还不如我自己写。

后来我们逼出来一个办法，叫**多 AI 交叉验证**。同一个任务，让两个以上的大模型分别产出，再把它们的输出按同一结构对齐，比较差异点。差异的地方，就是高风险区域，优先核查。

这个转变很关键——你不再是“逐行查错”，而是在“定位分歧”。查错像大海捞针，定位分歧像看一张热力图：哪儿红，先看哪儿。效率差了一个量级。

我们团队每天用这套方法做一件事，内部叫“知识咖啡”：从上百篇文章里筛出跟业务相关的内容，自动生成摘要和行动建议。靠人做至少两小时，AI 五分钟出结果，但早期幻觉率不低。用了交叉验证之后，错误被压到了可接受的范围内。

重点从来不是追求“零错误”，而是把查错成本，压到比自己做更低。

过了这条线，人才愿意用。这句话，是整件事的转折点。今天我们团队有一个叫“AI 先锋队”的飞书群，9 个真人配 8 个机器人，日常运转已经离不开 AI 了。但它不是规划出来的，是一次次被幻觉坑了之后，硬磨出来的。

为什么必须先个人，再组织

你发现没有——上面这套“多 AI 交叉验证”，没有一条是规划会议上拍出来的，全是在一线被坑出来的。这就引出一个很多转型方案都搞反了的关键问题：这种一线试错的经验，能靠自上而下的全员培训去推广吗？

不能。

因为决策者如果没亲自被幻觉坑过，就不知道错在哪、痛在哪、查错的时间到底耗在了哪个环节——这种东西，你坐在会议室里是画不出来的。

高风的蒋逸明提出过一个“三条曲线”的观察，我觉得特别准：技术曲线持续陡峭上行，人才曲线在 AI 赋能下明显跃升，而组织曲线却在短暂兴奋之后陷入停滞。很多企业不是没有聪明人，也不是没有工具，而是个人已经开始变快，组织流程却没跟上。个人效率的改善，不会自动变成组织能力。介一凡在论坛上把组织层面的解法落到四个点：知识萃取、生产关系重塑、复合思维人才、领导力聚焦——这些框架回答了“该往哪走”，但每个实践者还要回答一个更具体的问题：怎么让组织曲线真的跟上来？

我的答案是：**先让个人跑通，再谈组织**。让一线的人在自己的真实业务里找到纠错方法，再把方法沉淀成流程，最后才推到组织层面。

这跟进化的逻辑是一样的：变异、选择、隔离。个人先跑通，是变异——总得有人先冒出来，用自己的方式试出一条路；验证有效，是选择——只有经过真实业务检验的方法，才值得保留；固化成流程，是隔离——把验证过的方法从个人经验里提取出来，变成组织可复制的流程、工具和检查清单。

你不能跳过变异直接做选择，也不能跳过一线试错，直接给组织下标准。很多转型方案之所以落地不了，就是因为它们试图从“隔离”开始——先定流程、再推标准——而不是从“变异”开始——先让人在真实业务里自己跑通。

这一章留下的那个点

所以，如果你正在推 AI 落地，我有一个特别具体的建议：

别急着做全员培训，也别急着写一份很漂亮的转型规划。先找一两个愿意试的人，挑一个高频、真实、边界清楚的业务场景，让他们用自己的方式跑一遍。跑到一个具体的点上——查错比自己做更省。

这个点一旦找到，路就通了。然后再把它写成流程、工具、模板、检查清单，组织才有东西可以复制。

而能把这个点找出来、能在现场陪着一线员工一次次试、一次次磨，直到把纠错成本压下去的那个人——他干的，已经不是“培训”，也不是“咨询”，而是这本书要讲的那个角色的活儿了。

FIELD NOTES

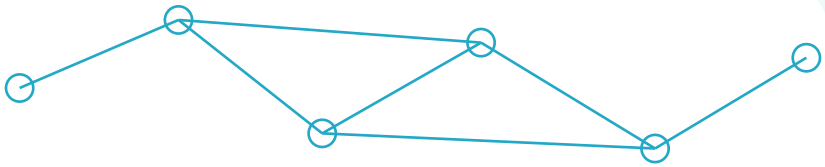
医嘱

01 真正的摩擦力不是抵触，是纠错成本——谁为 AI 的错买单，谁就没动力用。

02 别等模型变好，先把查错方式变轻：把纠错成本压到比自己做更低，人才会用。

03 先让个人跑通，再谈组织：变异 → 选择 → 隔离，顺序不能反。

CHAPTER 02 · END



CHAPTER 03

第 3 章 问题不在执行，而在斜率

> 一个坏系统，每一次都能打败一个好人。
>
> ——爱德华兹 · 戴明

主诉

本章要回答：为什么有些 AI 项目，方案扎实、团队也认真，却怎么也推不动？

有一次，我和一个做企业服务的朋友喝茶聊天。

他说起帮一家制造业企业用 AI 改造内部流程的事。立项、谈合同、组团队，前后投入了半年，到现在还卡在“试点阶段”，没有任何要推广的迹象。

他说了一句话，我觉得挺扎心：

“老板是真的想推，但下面的人有一百种方式让你推不动。”

我问他，觉得是哪儿出了问题。他想了半天，说：“也说不清楚，感觉哪里都没问题，但就是推不动。”

这种“哪儿都没问题，但就是不行”的感觉，是不是很熟悉？上一章我们讲了纠错成本，那是现场层面的摩擦力——AI 出了错没人愿意查，所以没人愿意用。这一章我想往更深一层走，走到结构层面——因为有些项目卡住，真的不是执行的问题。

不是执行的问题

这类项目卡住，通常会被归到几个原因上：推进不力、沟通不到位、执行团队能力不够。

但我观察到的情况恰恰相反：很多项目，执行层面其实没什么大毛病。方案扎实，团队认真，流程也都走了。但就是推不动。

为什么？

因为问题不在执行，在结构。

有些事情，放在特定的系统里，本来就不被允许发生。不是因为有人在明着拦你，而是这个系统的运行逻辑，会自动把它消化掉、排异掉。它表面上在“拥抱变化”，底层却在本能地抵抗变化。

要说清楚这件事，我想借一个词——**斜率**。

斜率：一件事的“结构性可展开能力”

这个词，是我最近听张帆老师在混沌学园讲“AI 时代第一性原理”时被重新激活的。

我说的斜率，不是增长快慢的意思。它指的是一件事在 AI 时代的**结构性可展开能力**：在给定的系统里，这件事到底有多大空间，能被 AI 真正放大？

你也可以把它理解成一个**放大系数**：在低斜率的系统里，AI 是一面放大镜，把你放大两倍；在高斜率的结构里，AI 是一根杠杆，把你放大一百倍。决定这个倍数的，从来不是 AI 本身，而是系统的结构。

有些事情天生低斜率。不是做的人不努力，是这件事所处的结构，决定了它的展开上限就那么高。有些事情天生高斜率，AI 一介入，每一分投入都被放大，边界持续往外扩。

斜率不是努力程度的函数，是系统结构的函数。

举个最直白的例子。同样是用 AI 处理工作：一个人在三人合伙的小团队里，效率提升了，省下的钱、多赚的钱，直接进自己口袋；另一个人在层级分明的大组织里，效率提升了，省下来的钱跟他一点关系没有。同样的 AI，同样的投入，斜率完全不同。不是人的问题，是他们所在的结构不同。

理解了斜率，再回头看我那位朋友卡住的项目，就清楚多了——他不是做错了什么，他是进了一个低斜率的结构，然后在里面拼命努力。努力是真的，但结构的天花板，明晃晃地摆在那儿。

旧逻辑的三种排异机制

那个推不动的项目背后，我觉得有三种旧逻辑在同时运作。它们不写在任何一份会议纪要里，却比任何明面上的反对都更有杀伤力。

第一种，权力护城河——权力结构不允许效率被替代。

老板支持 AI 增效，这往往是真的。但项目落地绕不过中层，而中层的权力，恰恰建立在“管人”上。AI 一旦提效，要管的人少了，他们的权力基础就动摇了。他们不需要明着反对，只需要“配合”得慢一点、“测试”得仔细一点、在每个审批节点上多问两句——项目就在流程里被慢慢耗掉了。这不是某几个人坏，是权力结构在自我保护。

MIT 有一项持续了十多年的跟踪研究，结论很清楚：影响企业数字化或 AI 转型成功的关键因素，不是高层的决心，而是基层员工的反馈和感受。如果新技术在基层的接触度非常高，大概率能成功——前提是基层的应用跟企业业绩结果呈正相关。换句话说，基层愿不愿意用、用得顺不顺，才是决定性的。而中层的权力结构，恰恰卡在了这个决定性的位置上。

第二种，线性算账法——价值体系只认线性收益。

就算绕过了阻力，做成了，最好的结果是什么？降本 30%，提速 40%。这是线性收益。而旧逻辑里衡量“价值”的尺子，本来就是线性的——做了多少、省了多少、交付了什么。AI 真正能带来的是非线性的变化（一个人带十个 Agent、一个流程被彻底重构），但非线性的收益，在旧的价值体系里往往看不见、也算不清，

所以它不会被优先支持。不是没人想要，是这套体系的眼睛，天生只能看见线性的东西。

第三种，基因排异反应——组织基因无法承载非连续变化。

有些组织，是被某种特定逻辑“长出来”的——它的能力结构、决策方式、利益分配，全都围着那套逻辑建立。你要它去承载一种完全不同的运行方式，那不是改几个流程的事，是要动基因。而基因层面的改变，代价极高、阻力极大、时间极长，大多数情况下，改造的成本远大于改造的收益。组织会本能地算这笔账，然后选择“维持原样”。

三种错配：为什么斜率低

三种排异机制，是从“旧逻辑怎么排异”的角度看的。换个方向，从“新逻辑为什么展开不了”的角度看，会看到三种更具体的错配。

第一种，杀鸡用牛刀——问题类型错配。

传统数字化擅长的是结构性、规则明确的问题——流程审批、报表生成、库存管理。这类问题，用传统 IT 工具就能解决，AI 的增量价值不大。AI 真正的核心价值，在那些非标的、半结构化的场景——需要经验判断的环节、需要综合多方信息的决策、需要理解上下文的交互。但很多企业把 AI 用在了前一类问题上，后一类反而没人管。结果就是：AI 在不痛不痒的地方做了优化，真正依赖人类经验的高价值判断环节，还在原地踏步。

第二种，步调不一致——速度错配。

AI 工具的迭代周期已经缩短到三个月——从 RPA 到 Workflow 再到 Agent，一年之内连换三代。而传统企业开发一个系统，动辄以年计。追求“一步到位”的大系统，等它上线的时候，当初的技术选型可能已经过时了。斜率低的项目，往往卡在这个时间差里：AI 跑得太快，组织的节奏跟不上，最后只能用旧方法做新事情。

第三种，永远在施工——验证方式错配。

传统项目制验收，要求系统完全成型才上线。但 AI 需要的是最小闭环验证——一个单次数据分析、一次文案修改、一个问答场景，每个微场景都要有“输入-处理-可评估输出”的完整链路。验证方式不对，项目要么永远停在“还没做完”的状态，要么做完了一看，跟真实业务对不上。

三种错配，指向同一个结论：**不是 AI 不行，是你把它放进了它展开不开的结构里。** 就像一颗种子，落在水泥地上，再怎么浇水也长不出来——不是种子的问题，是地面的问题。

什么样的结构，天生适合被 AI 放大

既然问题在结构，那就反过来想：什么样的结构，天然适合被 AI 放大？我观察到几个共同特征。

利益结构简单的地方，变化更容易发生。 当所有人都能从 AI 增效里直接受益，推进就不需要克服阻力——不是大家觉悟更高，是结构让每个人的利益方向一致了。

没有历史包袱的系统，更容易进化。 这跟规模大小无关，关键是运行逻辑越单纯，AI 嵌入的阻力越小。没有要保护的权力，没有要维持的格局，AI 就是在一张白纸上写字。

把 AI 当“生存条件”而非“工具”的地方，斜率最高。 这是最关键的一条。如果一个团队把 AI 当工具用，AI 带来的是效率提升，是加法；如果一个团队离开 AI 就活不下去，AI 就成了它的基础设施——AI 每进化一步，它的能力边界就往外扩一步，斜率是随着 AI 本身一起变陡的。这两种状态，不是程度的差异，是结构的差异。

这一章为什么重要

我那位朋友喝完咖啡，临走撂下一句话：

“也许我一开始就该想清楚，这个系统到底允不允许这件事发生。”

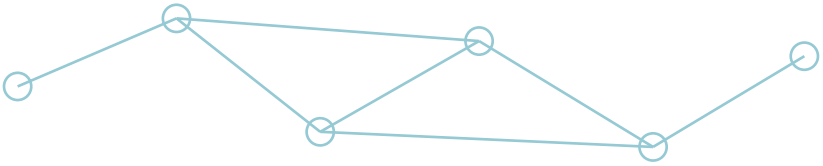
我觉得他碰到了一个真问题。

对正在推 AI 的人，斜率这把尺子有两个用法。一是**选战场**：你现在做的事，是在一个低斜率的系统里做优化，还是在一个高斜率的结构里放大变化？前者越努力越累，后者顺势就起。二是**改结构**：如果非要在低斜率的地方做，那真正该动的，可能不是方案，而是利益怎么分、流程归谁、谁从这件事里受益——先把结构的斜率抬上去，再谈落地。

三种错配，也是两把尺子。选战场时，先问：你要解决的问题，是不是 AI 真正擅长的类型？你的迭代节奏，跟得上 AI 的速度吗？你打算怎么验证——是等项目做完，还是先跑一个最小闭环？如果三个答案都不乐观，也许该换的不是方案，是战场。

而这第二件事——进到现场，先判断这个结构允不允许 AI 发生，再动手去调利益、调流程、调嵌入点，让一件低斜率的事变得高斜率——恰恰是普通的咨询和外包碰都不会碰的活儿。

它是现场医生的活儿。



FIELD NOTES

医嘱

01 项目推不动，先看结构允不允许，别只怪执行不力。

02 斜率是系统结构的函数，不是努力程度的函数。

03 选战场比埋头努力更重要：低斜率的地方，越使劲越累。

CHAPTER 03 · END

CHAPTER 04

第 4 章 FDE 的起源：从战场上的 Palantir 到 AI 时代的核心岗位

> 没有什么比第一手证据更可靠。
>
> ——阿瑟 · 柯南 · 道尔《血字的研究》

主诉

本章要回答：FDE 这个角色从哪来，为什么二十年后突然变成 AI 行业最贵的岗位？

“派人到客户现场，把东西用起来”——这件事，听起来像是 AI 时代的新发明。

其实不是。它有名有姓，有出身，已经二十岁了。

一个被战场逼出来的角色

故事要从 2000 年代初说起。

那时候，一家叫 Palantir 的公司，拿到了第一批客户——美国中央情报局（CIA）和国防部下属的情报机构。

团队很快发现，他们熟悉的那套软件开发流程，在这个场景里彻底失灵了。

正常情况下，做企业软件是有标准动作的：先谈需求，再做设计，然后开发、测试、上线。这套流程的前提是，客户能说清楚自己要什么。

可情报机构呢？他们既说不清自己的需求——因为很多任务本身就是模糊的、临时的；也不能把原始数据交给外部工程师——那是国家机密；甚至连真实的使用场景，都没法完整描述给你听。

需求说不清，数据不能给，场景看不到。传统软件开发的三个前提，一个都不成立。

怎么办？

Palantir 的联合创始人 Shyam Sankar 后来回忆这段时，给了一个很朴素的答案：你只能把工程师直接放到客户那里去，在现场观察，在现场构建，在现场迭代。

这个被逼出来的岗位，他们给它起了个名字——Forward Deployed Engineer，前沿部署工程师，简称 FDE。

Sankar 对它的描述，我特别喜欢，因为它一点都不高大上：

> FDE 在现场为客户铺一条能立刻通车的砂石路，后方的产品团队再把它翻修成一条通往所有客户的高速公路。

你品品这句话。它把 FDE 这个角色的两个本质，一句话说清了：第一，先解决“这一个”客户的真问题，哪怕用的是砂石路这种不完美但能立刻通车的办法；第二，再把现场打磨出来的东西，沉淀成能复用到所有客户的产品。

前者是脏活、急活、定制活；后者是把脏活变成资产。一个 FDE，得同时干这两件事。这条“从砂石路到高速公路”的暗线，会在本书后面反复出现——它几乎是现场医生这门生意的底层代码。

二十年后，这个角色突然变得很贵

把镜头快进二十年。

Palantir 长成了一家市值惊人的公司。一个被很多人忽略的数字是：到 2025 财年，它的人均年收入突破了 100 万美元——这在软件行业是个极其夸张的效率

指标。支撑这个数字的核心交付方式，正是当年那套“把工程师派到现场去”的 FDE 模式。

更值得注意的是，FDE 不再是 Palantir 一家的独门秘籍了。它正在变成整个 AI 行业最抢手、也最贵的岗位之一。

我们在引言里讲过，2026 年 5 月，OpenAI 和 Anthropic 几乎同时成立独立的部署公司、整建制收购咨询团队——OpenAI 收的那家英国公司，一口气并入了约 150 名 FDE 和部署专家；Anthropic 收的 Fractional AI 也有几十人的团队。据公开信息，两家公司的 FDE 团队规模分别在一百多人和几十人。按新公司的估值和团队规模折算，资本市场对单名 FDE 工程师的“定价”，已经逼近 1 亿美金——这是传统软件工程师难以想象的天价。

这里有一个有意思的传承。OpenAI 的首席研究官 Bob McGrew——主导了 ChatGPT 和 GPT-4 的开发——正是 Palantir 的早期高管（系作者转述，以本人公开表述为准）。二十年前他在 Palantir 亲身经历了 FDE 模式的诞生，二十年后他把同样的逻辑带进了 OpenAI 的部署战略。从战场到金融街，从情报机构到企业 AI，FDE 的底层逻辑没有变：客户说不清需求、数据不能动、场景看不见——把人派到现场去。

热度还体现在更具体的地方。你去 Y Combinator 的招聘板上看，会发现超过 100 家创业公司正在招 FDE，而三年前，这个数字是零。薪酬也水涨船高——据公开的薪资数据，Palantir 的 FDE 平均年薪在二十万美元以上，高的能到五六十万美元；与此同时，Palantir 的 FDE 还在被 Anthropic、OpenAI、Anduril 这些公司高价挖角。多家科技媒体在 2026 年的判断很一致：Forward Deployed Engineer，是这一年最热的岗位。

一个二十年前在战场上被逼出来的角色，正在重新成为 AI 行业的核心交付岗位。

为什么是现在？

因为历史在押韵。当年情报机构面对 Palantir 的困境——需求说不清、数据不能动、场景看不见——今天几乎每一家想用 AI 的传统企业，都在重演一遍。老

板说不清自己到底要 AI 干什么，核心数据不能随便喂给外部模型，真实业务流程从没被正式描述过。

二十年前那套“先谈需求再开发”的流程治不了情报机构的病，今天那套“卖你一个 API 自己回去用”的打法，也治不了企业的病。

同样的困境，召唤同样的解法：把人派到现场去。

FDE 不是什么

理解一个新角色，有时候说清楚它“不是什么”，比定义它“是什么”更有效。

FDE 不是外包程序员。 外包是客户提需求、团队按单开发、交付验收走人。FDE 不坐在家里等需求单——他常驻现场，自己观察业务流程，自己把那个说不清的需求挖出来、定义清楚。

FDE 不是咨询顾问。 咨询交付的是 PPT、方法论和一份转型建议，至于落不落得了地，那是客户和 IT 部门的事。FDE 交付的是一个能跑起来的东西，并且陪着客户用、陪着一起迭代，直到它真的在业务里产生效果。

FDE 也不是卖一套软件就走的 SaaS 销售。 标准化软件追求的是最大公约数，适合共性需求；FDE 开的是处方，解决的是这家企业自己特有的、高频的那个痛点。

FDE 更不是提示词专家。 提示词（Prompt）是你给 AI 的指令，提示词工程就是优化这些指令让 AI 输出更好——但它只管“怎么问”；FDE 打通的是 Prompt、Agent（智能体，简单说就是“能自动完成一串步骤的 AI 程序”，比如自动筛选信息、自动生成报告、自动比对数据）、数据、系统和员工操作的一整条链路。

把这四个“不是”叠在一起，FDE 的轮廓就清晰了：他是一个常驻现场、自己定义问题、用 AI 工具组合把方案做出来跑通、并且陪着客户算出结果的人。

如果还嫌抽象，就用全书那个药厂与医生的比方收个尾：外包程序员，是流水线上按图纸造药的工人；咨询顾问，是写论文告诉你这药该怎么吃的医学教授；SaaS 销售，是药店里卖非处方药的店员；提示词专家，是教你把“哪儿不舒服”说

清楚的导诊台。而 FDE，是那个坐在你病床前、看着你把药吃下去、还盯着你有没有过敏反应的现场医生。

他卖的不是工具，是结果。

不过，这里需要做一个说明。本书所说的“现场医生”，不是对 FDE 岗位说明的逐字翻译。在 Palantir 的原始定义里，FDE 仍然要求较强的工程能力——毕竟他们面对的是情报机构的数据管道和战场系统。但 AI 时代发生了一件事：“做出一个能跑的系统”的门槛，被狠狠往下压了一截。低代码平台、AI 辅助编程、智能体编排工具，让一个没有技术背景但懂业务的人，也能组合出可用的方案。所以本书用“现场医生”这个词，指的是一个更宽泛的角色：**凡是能进入现场、定义问题、组织技术能力并对业务结果负责的人，都属于这个角色**。他可以是写代码的工程师，也可以是不写代码但懂业务的专家——第 8 章你会看到一个一行代码不会写的人，怎么做出一套企业级系统。

而前面那个对比——同样的模型，FDE 团队能在摩根士丹利做到约 98% 的采用率，普通企业却日活寥寥——恰恰说明：决定成败的不是那盒药，是这个开方、陪诊、盯着疗效的人。

一条会贯穿全书的暗线

这一章我们追溯了 FDE 的来历，但有一个问题我先按下不表，留到后面慢慢展开：

为什么二十年前，FDE 这个角色那么稀有、那么贵，只有 Palantir 这种公司、面对 CIA 这种客户才用得起？而今天，连一个没有技术背景的业务专家，都有可能干上 FDE 的活儿？

答案藏在 AI 工具这几年发生的一件事里——它把“做出一个能跑的系统”的门槛，狠狠往下压了一截，压到了业务专家伸手就能够着的地方。

这件事，彻底改变了谁能当现场医生。但在讲“谁能做”之前，我们得先把这门生意的账算清楚：现场医生到底靠什么挣钱，凭什么它不是“更贵的咨询”或“重人力的外包”。

那是下一篇的事。

FIELD NOTES

医嘱

01 FDE 不是外包、不是咨询、不是卖软件——它对“业务结果”负责。

02 它的逻辑二十年没变：需求说不清、数据不能动、场景看不见，就把人派到现场。

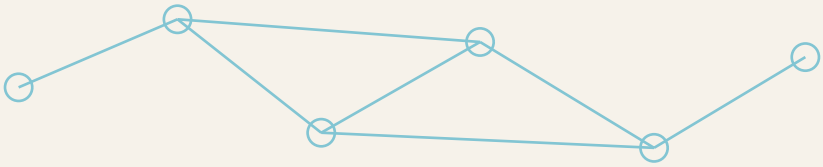
03 决定成败的不是那盒药，是那个开方、陪诊、盯着疗效的人。

CHAPTER 04 · END

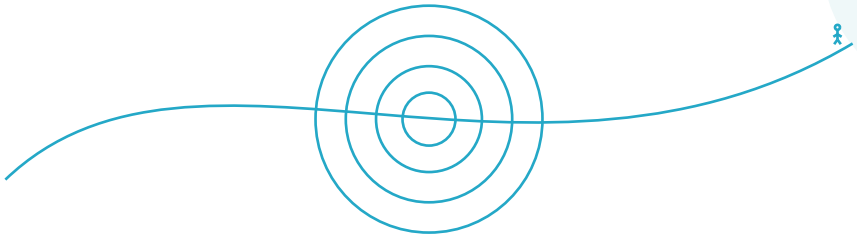
第二篇

凭什么

这门生意的逻辑



卖药的赚的是药钱，治病的赚的是把病治好。



CHAPTER 05

第 5 章 药厂与医生：造工具与用工具的分化

> 工欲善其事，必先利其器。
>
> ——《论语 · 卫灵公》

主 诉 本章要回答：AI 行业正在怎么分钱，FDE 凭什么能分到一杯羹？

引言里我立了一个比方：大模型像药厂，FDE 像医生。这一章，我想把这个比方从修辞推进到商业逻辑——因为它不只是一个好记的说法，它背后是一个正在发生的、关于 AI 行业怎么分钱的结构性判断。

一个三层分工的模型

我们先把这盘棋上的角色摆清楚。

大模型公司，是药厂。 它们研发通用的 AI 能力，提供模型、API、工具底座。这件事投入巨大、壁垒极高，但有一个特点：通用能力一旦成熟，就会被快速复制、不断降价，最后变成像水电一样的基础设施。药厂之间会卷价格、卷参数，但药厂不上门看病。

FDE，是医生。 医生不生产药，但他诊断业务问题、组合 AI 工具、交付一套能跑起来的工作流。他干的是诊断、开方、陪着病人吃药、盯着疗效、随时调整方案这一整套活儿。

传统企业，是患者。 患者关心的从来不是药的分子式，而是病能不能好——降本、增效、多赚钱。他不需要、也没义务搞懂模型参数。

这个三层模型不精确。药厂也做临床，医生也懂药理，边界没那么干净。但它的用处不在精确，而在于它一句话说清了一件正在发生的大事：**AI 行业正在分化，“造工具”和“用工具”，是两种完全不同的能力。**

过去两年，这两种能力的边界不只是模糊，在底层利益上甚至是对不齐的。药厂卖出 API 就确认了收入，至于你用不用得起来、用不用得出效果，那是你自己的事。企业说“我买了但不会用”，药厂说“我给你的已经是最好的药”。责任说不清，钱花了，账算不明白。

FDE，就是来把这段空白填上的人。他不站在药厂一边，也不站在患者一边——他替“业务结果”这个一直缺位的环节负责。

上一章已经说清了 FDE 与咨询、外包、SaaS、提示词专家的区别，这一章只算它的商业账。

它的护城河，是“懂行业的病”

很多人第一反应是：FDE 的本事，是不是“更懂模型”？

恰恰相反。

模型是药厂造的，而且越来越强、越来越便宜，“懂模型”这件事的稀缺性，是在快速贬值的。一个 FDE 真正的护城河，不是“懂药”，而是“懂这家企业、这个行业到底得的是什么病”。

懂行业的术语，懂业务的流程，懂老板半夜睡不着觉真正担心的是什么，懂一线员工为什么宁可绕路也不用那个系统。这些东西，藏在人的习惯里，藏在表格的命名里，藏在审批的绕路里，藏在“这个事你得先去问谁”这种没人写下来的潜规则里。

它们不在任何一份需求文档上，也不在任何一个模型的训练数据里。它们只能在现场，被一个愿意蹲下来看的人，一点一点摸出来。

Palantir 的 FDE 体系有个设计很能说明这个问题。它把 FDE 分成两个团队：Echo 负责“问诊”——到现场找到真正让业务疼的病根；Delta 负责“开方”——迅速把药配出来、喂下去。这套分工恰恰证明了一件事：真正的壁垒不在造药，而在望闻问切、对症下药。而且 Echo 的人不是“懂行的专家”，而是“对这个行业的现状不认命的人”——他们天天被烂工具折磨，自己比谁都急。这种“懂”不是知识层面的，是痛点层面的：你坐在办公室里是想不出这些痛点的，只有泡在现场的人，才知道哪里最疼。

我自己越往后做，越相信一个关于企业竞争力的公式：

> 未来一家企业真正的壁垒 = 廉价的通用大模型 + 昂贵的私有数据 + 独特的业务认知。

第一项，大家都能买到，而且越来越便宜——它正在变成趋近于零成本的通用基础设施，拉不开任何差距。真正拉开差距的，是后两项：你的私有数据，和你对自己业务的独特认知。而这两项的萃取、结构化、变成 AI 能用的东西，只能在现场完成。药厂的工程师再厉害，也碰不到你的私有数据，更钻不进你的业务认知里。

重，可能正是护城河

这里有一个反直觉的地方，值得专门点破。

传统 SaaS 信奉“轻”——产品标准化、交付轻、边际成本低、毛利高。但 FDE 这门生意，前期一定很“重”：要访谈、要梳理、要试点、要调模型、要改流程、要培训用户、要处理数据、还要承受反复的变化。从财务报表上看，这些全是成本。

可换个角度——这些“重”，可能正是护城河。

因为一旦 AI 工作流真正嵌进了客户的业务深处，它就不再是一个工具，而是变成了组织的一部分。到那时候，客户要迁走的，不是一个软件，而是一整套已经和自身业务长在一起的工作方式。迁移成本高到几乎不可能。

说得狠一点：传统 SaaS 的护城河是“转换成本”——换一个软件，很麻烦；而 FDE 的护城河是“共生成本”——客户要迁走的，是一套已经和自身业务长在一起的神经系统。拔出萝卜带出泥，没人敢轻易动这台手术。

传统软件时代的护城河，来自功能、数据和迁移成本。AI 时代的护城河，越来越来自另外几样东西：现场理解能力、业务知识的沉淀、持续迭代的机制，以及和客户共同进化的速度。

Bob McGrew 把这种模式总结成一句话（系作者转述，以本人公开表述为准）：“Do things that don't scale at scale”——在规模化层面做不规模化的事。每一单都是手工的，但每一次手工都在为规模化积累可复用的能力。听起来矛盾。二十年验证下来，它成立。

Palantir 的 5 天 Bootcamp 就是这个逻辑的极致版本：客户带真实数据来，FDE 带平台和工程能力，5 天内并肩做出一个能跑的 AI 应用——先让客户看到效果，再谈签合同。FDE 不是售前工程师，他是交付工程师，只是他在签约前就把交付做完了。

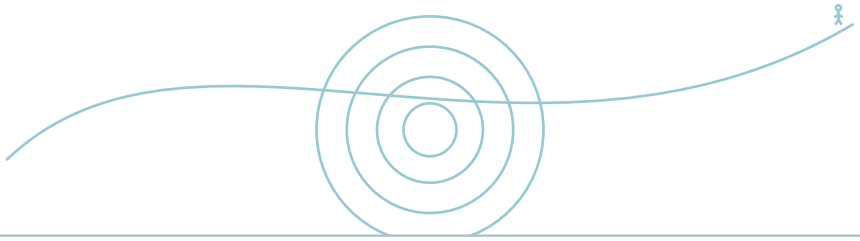
而这几样——现场理解能力、业务知识的沉淀、和客户共同进化的速度——不是纯产品团队坐在办公室里能做出来的。

为什么我笃定医生不会消失

这就是为什么我笃定，无论模型多强，医生这个角色都不会消失。

因为药越强，越需要会用药的人；病越复杂，越需要愿意走进病房的人。

下一章，我们就来回答一个更要命的问题：医生这门生意，到底跟卖软件有什么根本不同？为什么我说，如果你用卖软件的脑子去做 FDE，几乎注定算不清账、也走不远？



FIELD NOTES

医嘱

01 造工具和用工具，是两种正在分化的能力。

02 FDE 的护城河不是“懂模型”，是“懂行业的病”。

03 真正的壁垒 = 公有大模型 + 私有数据 + 独特认知，后两项只能在现场长出来。

CHAPTER 05 · END

CHAPTER 06

第 6 章 从交付产品到交付进化

> 人不能两次踏进同一条河流。
>
> ——赫拉克利特

主诉

本章要回答：为什么说 AI 没有“交付完成”那一刻，这门生意又该怎么收钱？

我前面提到的那位认识二十多年的老朋友，高利明，在 AI 投资圈里被人称作“思想家”。他三年前抛出过一句话——“coding 是模型价值的 99%”（系作者转述，以本人公开表述为准），当时听着夸张，现在回头看，几乎是预言。

他后来写了一篇文章，我读完，把他的核心观点嚼成了一句话：

AI 的交付物不是产品，AI 永远没有交付完毕的那一刻。

这句话乍一读有点绕，但你咂摸一下，分量很重。它说的不是“一个 AI 产品还需要继续优化”——所有软件都需要优化，这不新鲜。它指向的是一件更大的事：在 AI 时代，“产品”这个概念本身，正在发生变化。

为什么 AI 没有“交付完成”那一刻

过去我们说交付一个产品，意思很清楚：需求调研、立项开发、测试上线、培训使用、项目验收。中间会有变更，会有二期三期，但大体上是一个项目制的逻辑——有开始，有结束，有边界，有交付物。

可 AI 不是这样。

AI 的能力不是一次性封装好的，它是在使用中长出来的。你今天让它写一段材料，明天让它分析一个客户，后天让它参与一个流程，再过一段时间，它可能已经变成了你组织里某个岗位、某个环节、甚至某种工作方式的一部分。

这时候再问“AI 项目什么时候交付完成”，就像问一个新人什么时候“学习完成”一样。答案当然是：没有那一刻。如果真有，那反而说明它停止生长了。

我自己撞过一个特别具体的例子。有一家做技术的合作方给我们做 AI 项目，原本计划三到六个月交付。结果做了一年多，还在持续迭代——它没有“做完”，而是悄悄变成了一种驻场陪跑的关系。一开始我以为是项目管理出了问题，后来才明白，这就是 AI 落地的常态：它的开始点，不是立项，而是上线。

引言里我提到过高利明打的另一个比方：传统产品像“宏”，而 AI 服务像“宏生成器”。传统软件是固态的——Word 能做什么、Excel 能做什么，大体确定。厂商负责定义功能，用户负责学习使用。AI 是液态的——同一个大模型，对程序员是代码助手，对财务是报表分析员，对律师是合同审查员，对管理者可能是一个战略讨论对象。它的能力不是被提前定义好的，而是在每一次具体交互里被激活出来的。

所以，**传统软件交付的是功能，AI 交付的是可能性**。

但“可能性”本身不值钱。大模型很像一块生铁，潜力巨大，可企业要的不是一块生铁，而是一把适合自己手术台的手术刀。谁来完成这个从生铁到手术刀的锻造？

这就又回到了我们的主角——那个愿意在现场抡锤子的人。

“卖结果”到底怎么卖

如果 AI 永远没有交付完，那这门生意的钱，到底该怎么收？

这是 FDE 和传统软件最根本的分野。传统软件卖 License，卖订阅——客户买的是一个确定的产品，厂商交付的是一套确定的功能，双方交换的是确定性。而 AI 服务卖的不是固定功能，是持续演进的能力。高利明把它叫做“从卖 License 到卖 Evolution”。

落到收费上，最理想、也最难的一种，是**按结果付费**。

为什么难？因为过去“效果”算不清。一笔订单成交了，是销售的功劳，是产品的功劳，还是那个 AI 助手的功劳？不同角色对归因分歧巨大，谁也不服谁，最后只能退回按人头、按工时这种最笨的算法。

但 AI 带来一个有意思的变化：它让效果归因第一次变得**可审计、可量化**。一个 AI 工作流嵌进业务后，每一个环节的输入输出都有记录——它处理了多少单、拦下了多少错、把一件事从三天压到了三小时。当账能算清、且经得起第三方复核时，一个理性的老板，是愿意接受利润分成的。

我见过最漂亮的一种结构，是把交付拆成阶段、和价值绑死：先收一笔诊断咨询费，证明你看得准；再按跑出来的真实效果分成，证明你做得到。客户每一步都看得见产出，你也不再是“卖工时的乙方”，而是“和他一起算账的合伙人”。

说起来漂亮，做起来有三道硬坎。第一，**归因争议**：效果算出来了，但“这个效果有多少是 AI 带来的、多少是原有业务自然增长的”，双方很难达成一致——这件事必须在合同里提前写清楚归因口径，不能等事后吵。第二，**甲方赖账**：效果做出来了，客户却以各种理由推迟或拒付——毕竟，帮老板赚了钱，不代表老板就愿意分你钱。所以按结果付费的合同，必须设计分阶段验收和支付节点，不能一笔算总账。第三，**长周期分成的违约风险**：如果客户中途终止合作，已经沉淀的 AI 能力怎么算？模板归谁？数据怎么处理？这些也得提前约定，否则你投入越深、退出越痛。

能和客户一起算账，是这门生意能不能持续下去的关键。

上一章讲了“重，可能正是护城河”——那是从客户迁移成本的角度说的。这里从另一个角度看同一件事：前期重，意味着你在每一个客户身上积累了深度的业务理解；这些理解沉淀成可复用的方法论和模板后，下一个客户就轻了。但你不会因此“变轻”——因为你会在新客户身上挖得更深，发现新的问题，积累新的认知。你的“重”在持续增值，而不是在重复消耗。

这就是“交付进化”的真正含义：你交付的不是一次性的产品，而是一种持续变强的能力。客户付的钱，不是在买一个工具，而是在买一个越来越懂他的伙伴。

而这里藏着一道看不见的坎：从“可用”到“敢于托付”。客户一开始用你的系统，是因为它能干活；但只有当他发现你真的懂他的业务、愿意在现场一直蹲着、出了问题能接住，他才会把更深的业务交给你。

我把这个从“可用”到“敢于托付”的转折，叫做**信任拐点**。

跨过去之前，你是个供应商；跨过去之后，你才是合伙人。所有的 FDE 工作，本质上都是在为跨越这道“信任拐点”积蓄势能。

这本书最后还会回到这个概念。

拥抱“未完成”

高利明那篇文章真正打动我的，是它没停在“AI 很重要”这种正确的废话上。今天说 AI 重要，已经没有意义了。

他让我意识到一件更深的事：AI 正在把我们从一个追求“完成”的时代，推进一个接受“未完成”的时代。

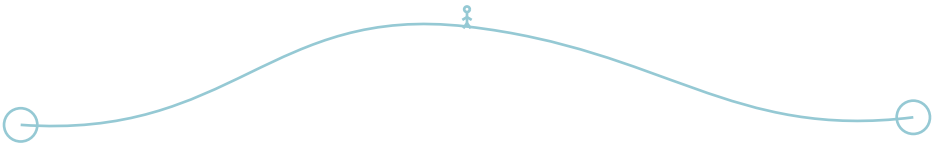
传统产品喜欢完成——需求完成、开发完成、上线完成、验收完成，完成让人安心。但 AI 的价值，恰恰来自它的未完成：它今天只是助手，明天可能参与流程；今天只能回答问题，明天可能帮你定义问题；今天是个工具，明天是组织能力的一部分。只有未完成，才有进化的空间。

所以，如果说过去软件公司的核心命题是“如何把一个产品做出来、卖出去”，那么 AI 时代，很多公司的核心命题会变成：“如何和客户一起，把一种能力长出来。”

前者强调产品、功能、交付、验收，结束点是上线；后者强调场景、反馈、迭代、进化，开始点才是上线。

在这个时代，固定功能会越来越便宜。真正贵的，是那个能把 AI 放进真实世界、并让它不断长出来的人。

道理讲到这儿都很美好。但下一章我想泼一点冷水：就算你想“管理进化”，第一关——让系统稳定地、像样地跑起来——大多数 AI 项目其实都没过。我用团队里一个很小的真实项目，讲讲“跑通”和“跑好”之间，那道真正的分水岭。



FIELD NOTES

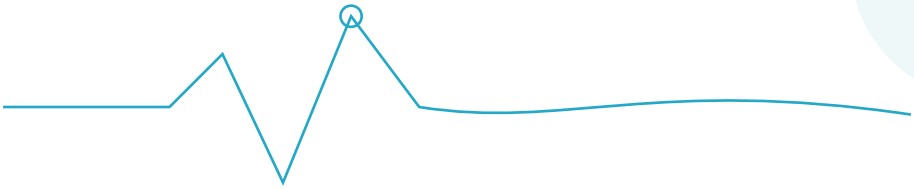
医嘱

01 AI 不是“做完”的项目，是要有人一直陪着调的活物。

02 从卖 License 到卖 Evolution，最理想的是按结果付费。

03 前期越重越像成本，但那恰恰是别人抢不走的护城河。

CHAPTER 06 · END



CHAPTER 07

第 7 章 从跑通到跑好：AI 项目真正的分水岭

> 跑出一个结果不难，难的是让它越跑越好。
>
> ——付伟

主诉 本章要回答：为什么“能跑了”离“能用”还差一大截？

上一章讲“管理进化”，听起来很高级。但我想先把人从云端拽回地面：大多数 AI 项目，连“稳定地跑起来”这第一关都没过，更别提进化了。

这件事，我用团队里一个很小的项目讲，可能比任何大道理都清楚。

一个早上 7：30 的小项目

有一段时间，我让团队启动了一个多模态学习项目——每天自动筛一份对我们有价值的研究报告，生成 PPT、信息图、音频、视频和小测验，让大家按自己的习惯去消化。任务交给了新来的同事小冬，带着实习生小陈做。

启动大约一周，产出就跑起来了，每周能出三四篇。每天早上 7：30 左右，小陈把 PDF、音频发到学习群里，我开车时听听音频，挺方便。

但试了几天，我觉得不对劲——**频率不稳定，质量有波动**。有时候推出来的 PPT 排版错乱，有时候音频里夹着突兀的机器音。说句实话，如果我是最终用户，用两天我大概也会把它静音。

我当时心里就琢磨：这不只是执行细节的问题，是整个思路还停在“把任务做完”上，还没想清楚“把系统建好”是什么意思。

正好小冬带小陈来汇报，我听完提了一轮反馈。说实话，我以为这些问题至少得折腾一两周。结果几天之后，第二代版本就上线了，效果比我预期的好不少。后来小冬把整个迭代过程系统梳理了一遍，里面有几个判断，我觉得值得拿出来讲——因为它们说的不是这一个项目，是几乎所有 AI 项目都要过的那道坎。

跑通，不等于跑好

AI 项目最迷惑人的地方，是它“跑出第一个结果”特别快。一周就能看到产出，于是所有人都松一口气，以为成了。

但跑出结果，和建成能力，是两回事。对任何一个 AI 项目，我现在都会逼着自己问三个问题：

第一，换一个人来接手，还跑得起来吗？

第二，跑了三个月，质量还能保持吗？

第三，每跑一轮，它有没有在变得更好？

这三个问题都能答“是”，才算真正建立了能力。只要有一个答“不是”，那“跑通流程”就只是起点，不是终点。大量 AI 项目死在哪儿？死在它们永远停在了“今天跑通了”的兴奋里，从没回答过这三个问题。

那怎么从“跑通”走到“跑好”？我从那次迭代里，提炼出五个特别实在的认知。

五个让系统“长好”的认知

第一，建蓄水池：别赌单日运气，给质量托底。

传统的内容筛选思路是：每天抓新内容，再从当天里挑最好的。问题是，输出质量完全绑死在“今天抓到的东西够不够好”上——某天信息源拉胯，输出就跟着拉胯。团队主管靖总监提了个办法，叫“动态蓄水池”：把每天筛出的优质内容沉淀进一个固定容量的池子，对全池重排，再从池里选最终输出。这样单日波动就不

会直接传到输出端。它带来三层变化：质量有了托底；优质内容从一次性消耗品变成了可复用的资产；还为后面引入人工反馈留出了接口。

第二，重构流程：别给旧马车装电动机。

“在旧流程里嵌 AI”和“用 AI 重构流程”，听着差不多，其实是两种完全不同的工作方式。前者是原流程不动、几个步骤交给 AI；后者是重新审视整条流程，问哪些环节是为人工时代设计的、在大模型条件下已经没必要存在了。历史上有个绝妙的类比：电力革命初期，工厂换上了电动机，却还沿用蒸汽时代的动力分发结构，效率根本没提升——真正的效率释放，是在工厂彻底重新设计了内部布局之后才发生的。AI 项目同理，只把 AI 嵌进旧 SOP，你得到的通常是“更快地重复旧流程”，而不是“适合 AI 时代的新流程”。说得再直白一点：今天大多数企业的 AI 项目，就是在给蒸汽时代的工厂装电动机——流程没变，组织没变，只是把“人工出错”升级成了“AI 加速出错”。

第三，并行验证：从“事后备份”到“同时开几扇门”。

很多人防工具风险的第一反应是“做备份”。但备份像出门带一把备用钥匙——主路径失效了再换；并行是同时开着几扇门，不同入口在同一阶段就共同参与。更进一步，把流程拆成“大模型基础能力 / 工作流编排 / 存储输出”三段，每段配不同厂商的方案，做系统性的组合验证——这本质上是一种低成本穷举创新空间的方法。复杂系统难优化，往往不是没有更好的选项，而是从没系统地看清楚选项空间。这个思路，和第二章讲的“多 AI 交叉验证”是同一件事——把查错变成定位分歧，效率差一个量级。这里不再展开，只强调一点：它不只是“查错”的方法，更是“看清选项空间”的方法。

第四，反馈闭环：初始准确率不是核心，越跑越准才是。

AI 系统早期，最容易被拿来比的是“第一次跑得准不准”。但长期看，初始准确率不是最核心的。真正决定系统上限的，是有没有反馈闭环——能不能越跑越优化。没有反馈，系统只是静态工具；有反馈，它才可能进化成动态能力。对内容项目尤其如此：筛得好不好，不只看模型识不识别“好内容”，还看输出之后有没有人打开、读完。这些指标看起来在用户那边、不可控，但其实每一个都有对应的干预手段——绩效从来不是自然发生的，是被持续管理和实验做出来的。

第五，沉淀资产：记录弯路，比记录成功更值钱。

这是最容易被忽视的一条。日常工作里，记录是补充动作，核心是交付结果。但对创新型 AI 项目，恰恰相反：结果可以被复制，工具会快速迭代，今天有效的方法明天未必领先；真正难以被复制的，是探索过程中沉淀下来的判断逻辑、试错经验和方法抽象。主动记录走过的弯路，不只是留档，更是在沉淀认知资产——而这也是个人能力成长最有效的来源。

从“消费智能”到“积累智能”

把这五条串起来，其实是同一件事：

AI 项目真正要构建的，从来不是完成一次交付、部署一个工具，而是一种可积累、可反馈、可演化的能力系统。

如果每次只是调用工具、拿到结果、完成任务，本质上还是在“消费智能”；如果能把筛选逻辑、反馈机制、工具组合、过程经验不断沉淀下来，让每一轮运行都成为下一轮优化的基础，项目就在从“消费智能”走向“积累智能”。

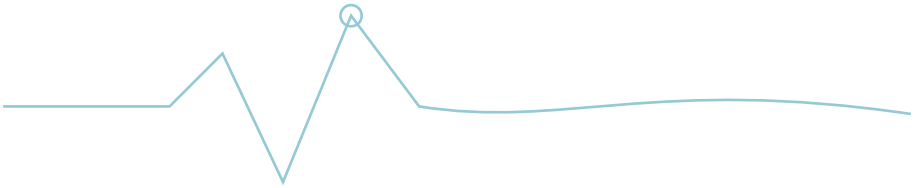
这个区别，也许才是 AI 时代真正值得关注的分水岭。

而你有没有发现，能从“跑通”逼到“跑好”、能问出那三个问题、能搭出蓄水池和反馈闭环的那个人——他干的，正是现场医生的活儿：他不满足于药能吃下去，他要的是病人真的、持续地好起来。

但这里藏着一个巨大的悖论：要做到上面这五点，通常得懂代码、懂架构、懂工程。一个不懂这些的人，连“跑通”都费劲，谈何“跑好”？

直到我看到陈澄波做的事。

这位做了十几年人才发展、一行代码不会写的老兵，不仅跑通了，还跑好了。下一章，我就讲讲这扇门，是怎么被他一行代码不写、硬生生撞开的。



FIELD NOTES

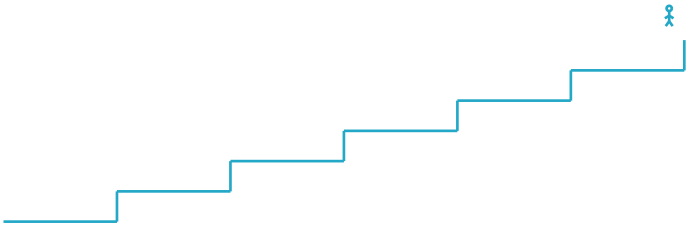
医嘱

01 跑通≠跑好，问三句：换人能跑吗？三个月质量还在吗？每轮在变好吗？

02 初始准确率不重要，反馈闭环才决定系统的上限。

03 从“消费智能”到“积累智能”，过程记录是最值钱的资产。

CHAPTER 07 · END



CHAPTER 08

第 8 章 不写代码也能交付：那道门已经开了

> 计算机是我们发明过的最卓越的工具，它就像思想的自行车。
>
> ——史蒂夫 · 乔布斯

主诉 本章要回答：一个不写代码的业务专家，凭什么也能交付企业级系统？

前面几章，主角多少都带点“工程”的影子。这一章我要讲一个一行代码都不会写的人。

正是他，让我彻底相信，FDE 这道门，已经向普通业务专家敞开了。

一通让我盯着窗外发呆的电话

我前面提到多次的一个老朋友，陈澄波。他做了十几年人才发展，早年是咨询顾问，后来创办了国内人才发展与移动学习领域的一家知名公司，也是我认识的人里产品意识最强的一个。但他有一个标签很清楚：非 IT。不写代码。

2026 年春节刚过，他突然发消息给我，说春节干了件大事，要好好聊聊。我们通了一个多小时电话。

他说：他用不到两个月、一行代码没写，做出了一套支持一万人使用的企业级 360 干部测评系统。

挂掉电话，我盯着窗外想了很久。电话里他的情绪很复杂——有做成事的兴奋，有对行业变化的焦虑，还有点试水成功的庆幸。但最触动我的不是这些，是他语气里那种异常的平静——好像“非 IT 专家自己开发企业系统”这件事，本来就是 AI 时代理所当然的事，已经不值得大惊小怪了。

也许，真的不用再大惊小怪了。

五个版本的踩坑史

事情的起点有点戏剧性。他拿到一套早年给大型国企做的旧 360 测评系统代码，想法很简单：喂给 AI 重写一遍，在这基础上做二次开发，当个人练手项目，成不成无所谓。

结果 IT 同事直接泼了盆冷水：“AI 根本不可能理解旧代码的内部逻辑，没有完整的产品原型手册根本做不成。”

这句话反而把他激到了。他偏不信。

第一版： 把全套旧代码、前端截图、操作视频全喂给 AI 工具，还专门建了个 coding 助手做规划。看起来准备充分，三天就做不下去了——AI 像个爱偷懒的外包，拿到旧代码直接照搬，原来代码里所有的毛病全保留，稍微改一点就报错。他陷进了改 bug 的泥潭，越改越深。

第二版： 调整策略，只保留前端代码，让 AI 分析功能、重搭后台。还是不行，前端的老逻辑把 AI 牢牢捆住，光打通前后端就耗光了精力。两三天后又放弃。

第三版： 听 AI 建议，引入国内最流行的开源脚手架“若依”，它把用户管理、权限管理这些底层都做好了。本以为找对路了，没想到 AI 还是沿用旧前端逻辑，不符合若依的代码规范，改完根本跑不起来。第三次放弃。

前三个版本的失败，让他想明白了一件致命的事：他一直在用 AI **翻译**旧世界，而不是用 AI **构建**新世界。想通这一点，他索性破罐子破摔——而这一摔，反倒把门撞开了。

第四版：推倒重来，反而走通了。 折腾到这一步，他索性把所有旧代码全扔了，从零梳理业务需求。他换了个切入方式：先让 AI 把国内外主流的 360 测评系

统研究一遍，摸清各家的功能逻辑；然后他自己来定架构——不是照抄谁，而是按他对 360 测评这件事十几年的理解，从头想：评价关系怎么建？权重怎么分配？问卷怎么组合？每个问题都由他这个业务专家来回答，AI 只负责把回答变成能跑的代码。

这一步，反而走通了。更妙的是，彻底摆脱旧代码后，他反而有机会重新想清楚很多事——那些以前因为开发限制根本排不进需求的想法，这次全加进去了：一键智能匹配评价关系、动态评价组合、人群权重自适应、AI 根据评价人身份自动切换视角……这些都是行业里原来没做到的功能。春节期间，整条路径跑通了。

第五版： 系统跑通后再做分工优化——他负责宏观业务规划，开发和改 bug 全交给 AI 的 Plan 模式，最难的算法逻辑靠自己反复多场景测试来调通。

那道门，到底意味着什么

他这两个月折腾出来的结论，我觉得比那套系统本身更有价值。这里先讲最直接的一条，更深的那条留到第四篇专门说。

最直接、也最让我意外的是：**非 IT 人员，真的能落地复杂企业系统了。**

在这个案例里，原来专业开发团队要五六个月做的东西，一个非 IT 背景的业务专家，两个月就做出来了。但比“快”更重要的，是另一个变化：以前你想做什么，得先说服开发团队，排期、沟通、等待，想法在这个过程里会被磨损、被妥协；现在，你的想法，你自己就能落地。

那道曾经把业务专家挡在系统之外的技术门槛，开了。

我必须补一句，免得你误读——门槛降低，不等于人人都能做出真正的产品。他能做成，根本原因不是他会用 AI，而是他脑子里本来就装着十几年的业务积累：评价体系怎么设计、权重怎么收口、哪里必须严、哪里可以松。这些 AI 给不了，是他自己带进来的。一个没有真业务经验的人，用同样的工具，做出来的往往是“看起来像产品，用起来是玩具”。这一点很关键，我们第四篇会专门展开。

一句话说透：**AI 把写代码的门槛踩平了，却把业务深度的门槛垫高了。** 以前，懂业务不懂技术，做不出系统；现在，懂业务不懂技术，照样能做出系统；但如果不懂业务——哪怕你懂技术——也只会做出一堆废铜烂铁。

市场已经在用脚投票

聊到这儿，我和他都有点感慨。

SaaS 板块的股价跌得很惨。很多人归因于市场情绪、利率、宏观周期。但我俩的判断是：市场在用脚投票一个更深层的趋势。

传统 SaaS 的核心价值，是“开发成本低、迭代快——你自己做不了，买个现成的”。可现在，AI 既能帮你更快做出来，还能完全按你的需求定制。更要命的是，连“复用性”这个理由都在消失：以前软件要卖给很多人，所以做成通用的；但如果我就是给自己企业用，我凭什么要迁就一个本来不是为我设计的界面？

“所有软件都会消失”这话有点绝对。判断一个软件会不会被 AI 定制冲击，核心看一件事：它服务的是大量用户的通用需求，还是少数人的定制需求？前者有网络效应做护城河（比如社交产品），后者几乎没有。而企业级工具类软件，本质上都是少数人用的定制化需求——这个方向受 AI 冲击的压力，已经在显现了。

这件事对不同企业还是不对称的。**新企业**最轻松：从零开始直接做定制，根本不必采购传统 SaaS，业务长什么样系统就长什么样。**老企业**面对的才是真难题，而且难题不是技术，是内部——有历史数据、用户用惯了旧界面，哪怕只改一点点都可能引发集体抵制。技术问题有解法，人的习惯和情绪，没有快捷键。

动手，比讨论更重要

挂掉电话，我一直在想一件事。

现在群里每天都有人在讨论 AI 颠覆这个、颠覆那个，观点、预测、焦虑，说得热闹——我有时候也是其中一个。但真正动手下场做项目的，没几个。

他这个项目最有价值的地方，不是做出了一个系统，而是他实打实证明了：AI 已经把能力的边界，交到了每一个业务专家手里。

原来你懂业务却做不出系统，是因为你被开发能力这道门槛挡在外面。现在那道门，已经打开了。你的想法，你自己就能落地。

在这个案例里，AI 首先表现为释放专家产能——把原来被开发能力困住的业务判断力，彻底释放了出来。至于它对岗位总量的影响，则取决于组织如何重构：是把释放出来的人力和判断力投向更高价值的工作，还是简单地缩减人头？答案不在 AI，在组织自己的选择。

那么问题就来了：门开了之后，到底什么样的人能走进去、走得远？这道门背后，又藏着哪些不会因为工具变强就消失的硬功夫？

这正是后面两篇要回答的。在那之前，我们先在第三篇里把现场医生最核心的方法论拆开——他到现场之后，究竟是怎么一步步把 AI 落下去的。

FIELD NOTES

医嘱

01 那道把业务专家挡在系统外的技术门槛，已经开了。

02 旧代码对 AI 是负资产，不是资产。

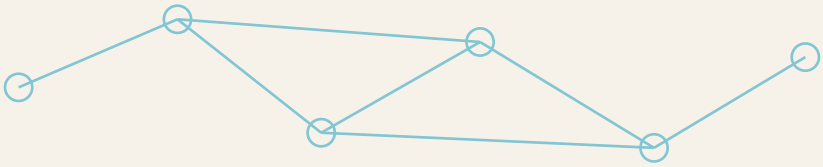
03 门槛降低≠人人能做——没有真业务判断，做出来的只是玩具。

CHAPTER 08 · END

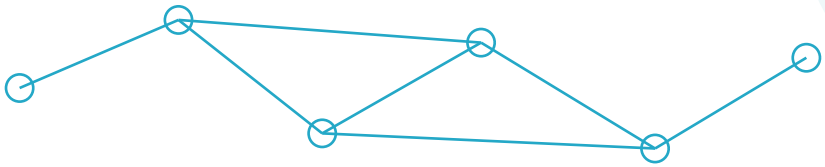
第三篇

怎么做

现场医生的方法论



最难、最琐碎的那段路，恰恰最值钱。



CHAPTER 09

第 9 章 本地化部署：我们踩过的三个坎

> 合抱之木，生于毫末；九层之台，起于累土。
>
> ——《道德经》

主诉 本章要回答：一个真实的本地化部署项目，会踩哪三道坎？

2025 年 3 月，我们启动了一个智能投资平台项目，核心是大语言模型本地化部署：整合团队二十多年的专业资料建知识库，做研究报告生成、投资分析辅助、知识库问答、文档智能检索。

本来以为三个月就能上线。结果用了大半年才正式跑起来，而且项目远没有结束——新的需求不断冒出来，新的子项目不断衍生出来。

这一章写于项目启动一年多后的今天。回头看那大半年，正是踩坑最密集、认知变化最剧烈的一段。有些坎当时觉得天要塌了，事后看不过是必经之路；有些当时没当回事的小问题，后来反而成了最大的隐患。趁记忆还鲜活，我把三道最刻骨铭心的坎记下来。

一年多下来，账面成绩单挺好看：完成数万份资料治理，搭起上百个专业知识库，开发 50 多个场景化智能体，团队基本形成了 AI 辅助工作的常态。

但你要问我“这大半年顺不顺”，我的答案是：**成果真实可感，挑战刻骨铭心。**

我不是技术背景的人。这一章，我就以一个“过来人”、一个非技术管理者的视角，讲讲我们爬过的三道坎。这三道坎，恰好对应一句话——我后面会反复说：AI 落地的难点，从来不是技术，是“人 + 流程 + 系统”的协同。

第一道坎：技术适配，别指望一次搞定

出于业务安全和数据主权的考虑，我们一开始就选了国产 GPU 这条技术路线。这个选择本身没错，但它意味着我们要去啃一块硬骨头——光是克服适配问题，就花了整整三个月。

那三个月像在玩“打地鼠”：今天 GPU 莫名“消失”，明天模型版本对不上，后天 OCR 又跑不动，按下这个、那个又冒出来。

这一道坎，我留到第 11 章专门扒开——因为国产 GPU 适配本身就是一场被严重低估的“生态战争”，那些惊心动魄的细节放在那儿讲更过瘾。这里只先钉一个结论：**技术适配不是一次性工作，是一场追版本的马拉松，不是百米冲刺。** 你得预留专人、留出缓冲、建一份“踩坑—解法”的清单，把它当成长期投入来管，而不是指望一次搞定。

倒是有一条策略，值得在这儿先说：**先验证再深耕。** 一开始我们试图同时解决国产 GPU 全链路上的所有适配问题，结果前三个月几乎都耗在了适配上。后来换了思路——仍然坚持国产底座，但先选择兼容度更高、生态相对成熟的一套国产硬件与模型组合，集中打通一个高频业务场景；等业务价值被验证、团队有了信心，再逐步扩展到更多模型、框架和硬件组合。这样既守住了国产化约束，也不会让业务方因为适配周期太长而失去耐心，团队更不至于在“还没看到效果”的阶段就耗尽士气。

先让病人看到药有效，再扩展药方——这不是降低标准，而是调整顺序。

第二道坎：知识库，别指望一劳永逸

知识库（简单说，就是把企业的文档、资料清洗整理后，变成 AI 能检索和引用的“外挂大脑”——AI 回答问题时不再只凭自己的训练记忆，而是从你喂给它的真实资料里找答案）这大半年我们整理了几万份资料、建了上百个知识库。听着唬人，其实是个“边建边用边优化”的苦活。

一开始我们天真地以为：“把所有文档都喂给 AI，知识库就建好了。”结果发现两件事：存量数据质量差，增量数据没人管。

存量有多乱？我们做投资，过去积累了大量投后报告、尽调文档、行业研究。全导进去之后——同一家公司 2021、2022、2023 年的投后报告都在，AI 召回时随机抓一份，答案前后矛盾；一些重要的会议纪要还是纸质的，根本没数字化；2022 年的行业数据混在里面，被 AI 拿去回答 2024 年的问题，直接误导决策。有一次同事问 AI“公司 A 的最新估值”，AI 答了个 2022 年的数，同事拿着去开会，被当场质疑。那一刻我们意识到：**数据质量，决定了 AI 的价值上限**。

增量更要命：知识库建好后，新资料没人及时录。用户问“公司 B 的最新财务数据”，AI 答的还是半年前的——用户一句“AI 没用，我还不如问同事”，转身就走了。

我们后来摸出一套知识库运营机制，就三个字：**清、录、看**。

清：花一个月清存量——去重（每家公司只留最新版，旧版归档）、打标签（“主体 + 时间 + 类型”，避免 AI 混淆）、清过期（标注有效期，过期自动归档不再召回）。

录：把增量更新绑进 OA——投资部尽调报告审批通过后自动同步知识库，财务部审计文件更新后自动同步。不依赖人的自觉，靠系统保证数据“鲜活”。

看：每周盯三个指标——覆盖率（用户的问题有多少能在库里找到答案，目标 >80%）、准确率（>95%）、时效性（3 个月内更新的文档占比 >60%）。任何一个不达标，立刻查。

记住：知识库是个“活”系统，要持续运营，不是建完就完。

第三道坎：工作流，别指望自然养成

这大半年，十几位同事在专家带领下做了 100 多个智能体。其中不少还比较简单，只在特定时候偶尔使用；也有一批已经成为日常高频使用的好帮手。现在，大家每天上班打开智能平台，已经成了习惯。但这个“习惯”不是自然长出来的，是“强制嵌入 + 持续陪跑”逼出来的。

启动时开动员会，介绍 AI 能干啥。会后就有同事私下嘀咕：“AI 是不是来抢我们饭碗的？”——这不是个例。还有人愿意试，却不会用：简单问题还行，复杂一点的“分析公司 A 财务、对比去年、找出异常并解释原因”，AI 要么只分析今年、要么只对比、要么找出异常却不解释，同事吐槽“我还得学怎么伺候它，比自己做还累”。更普遍的是，AI 没嵌进日常流程，打开它是个额外动作，不打开也不影响工作，那自然就不打开。

这一坎我在第 12 章会深讲。这里先给方法，还是三个字：**嵌、教、陪**。

嵌：把 AI 钉进流程——所有报告必须用 AI 生成初稿再人工改，所有报告必须过 AI 审核查数据一致性。让它从“可用可不用”变成“必须用”。

教：请专家手把手，第一个月教怎么问、第二个月教怎么做智能体、第三个月教怎么优化，而且是“边讲边做”，陪着一起把一个真实智能体从头做出来。

陪：项目不是交付即结束，而是定期看使用数据、收一线反馈、做优化迭代。

三道坎，其实是同一件事

回头看，这三道坎有一个共同点：**它们都不是纯技术问题**。

技术适配，需要人去长期维护；知识库，需要人去持续运营；工作流，需要人去主动使用。所以本地化部署真正的难点，从来不是技术，是“人 + 流程 + 系统”的协同。

波士顿咨询有个研究说，企业做 AI 转型，需要把三分之二的努力和资源放在“与人相关的能力”上，剩下三分之一才给技术和算法。换句话说，你做 AI 项目，三分之二的精力得花在“教人用”上，只有三分之一花在“搞技术”上。

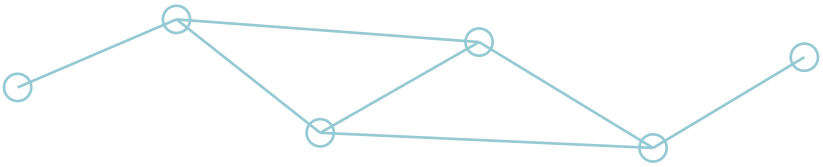
我自己的体会，可以总结成一个开车的比方：

技术适配是**地基**——地基不稳，一切白搭，但地基不是一次性工程，要长期维护；知识库是**燃料**——没有高质量数据，AI 就是空壳，但燃料要持续补充；工作流协同是**发动机**——不把 AI 嵌进日常动作，用户永远用不起来，但发动机不会自动点火，得有人强制打火。

光有车不行：地基不稳、油箱空着、没人点火，AI 都只是停在院里的摆设。而把这三件事——稳住地基、添足燃料、点火上路还陪着开——一肩挑起来的那个人，就是既会修底盘、又会加油、还能陪着老板练车的教练。这个教练，就是现场医生。

而这三件事——长期维护地基、持续添燃料、强制打火并陪着开——加在一起，就是现场医生在一个项目里真正干的活。它一点都不性感，但这正是 AI 能不能在一家企业真正留下来的分水岭。

接下来三章，我把这三道坎里最硬的几块，一块一块敲给你看。先从最容易被外行想当然的那件事开始：选模型。



FIELD NOTES

医嘱

01 技术适配是地基，要长期维护，不是一次搞定。

02 知识库要“清、录、看”，不是建完就完。

03 工作流靠“嵌、教、陪”强制养成，不会自然发生。

CHAPTER 09 · END

CHAPTER 10

第 10 章 模型选择：一场“硬件＋模型＋场景”的三维匹配

> 所有模型都是错的，但有些是有用的。
>
> ——乔治·博克斯

主诉 本章要回答：本地化部署，到底该怎么选模型？

启动智能投资平台时，我们定了技术路线：本地化部署，用开源模型。

当时我的想法特别朴素：既然用开源模型，那选个最强的部署上去不就行了？

结果实践下来才发现，本地化部署的模型选择，是一个“硬件＋模型＋场景”的三维匹配游戏——绝不是“看排行榜挑最好的”那么简单。这一章，讲我们在选模型上踩的三个坑。

坑一：以为最大的就是最适合的

项目启动时我们有 8 张国产 GPU 卡，想法很直接：上最强的开源模型。于是直接部署了千问-72B，心想“一步到位”。

两个月后发现：72B 跑不起来——显存不够，推理速度太慢，并发能力差。

显存是硬约束。 千问-72B 在全精度下要约 290GB 显存，我们有 320GB，看着够，实际只剩 30GB 处理请求——3 个并发就崩。后来换成千问-32B 量化版，显存占用降到 64GB，双卡就能跑，并发能力提到 10 路。准确率略有损失，但用户体验好了 10 倍。本地化部署不是“模型越大越好”，是“硬件装得下、跑得动的模型才是好模型”。

速度决定用不用。 千问-72B 平均响应 90 秒，千问-14B 只要 2 秒。用户测试很残酷：响应超过 50 秒，满意度跌到 30% 以下，很多人直接放弃。速度不是可选项，是“决定用户用不用”的开关。

多卡协同会资源错配。 8 张卡理论上是 8 倍性能，实际只提升 4-5 倍。大模型占 4 卡，小模型和 OCR 占 2 卡，剩下 2 卡因为调度限制长期闲置——25% 的算力在“摸鱼”。我们后来又租了一台 2 卡实例专门跑轻量任务，让大模型独享 8 卡，利用率才上来。

坑二：开源模型不是“全能选手”

选定了尺寸，还有一道题：用谁家的模型？我们在 DeepSeek、千问、Llama 之间纠结了很久，最后是用真实业务数据一个一个测出来的。

DeepSeek 代码能力极强，千问中文理解强。实测下来：代码场景 DeepSeek 准确率 90%、千问 85%；中文场景千问 92%、DeepSeek 88%。有个典型案例——让模型解读一份省级“十四五”能源发展方案，千问能精准提取条款、结合本地实际分析机会，DeepSeek 给的是通用框架，缺本地化细节。海外的 Llama 更“水土不服”：英文场景准确率 88%，中文掉到 75%，对中文政策和行业术语理解明显不足。

我们最终的决策不是“选一个”，而是建了一棵**模型选择决策树**：

- **按任务类型选家族**：代码开发用 DeepSeek，中文业务用千问，多语言用 Llama，多模态用 Qwen-VL——按任务动态路由。
- **按复杂度选尺寸**：FAQ / 问答用 14B 以下，报告生成用 14B – 32B，复杂推理才上 32B – 72B。
- **按硬件约束选架构**：显存充足用 Dense，显存紧张用“小模型 + 量化”。

这一步的心得就三个字：**测、分、稳**。用你自己的真实业务数据测，别只看排行榜；任务分层、场景分类，别想一个模型打天下；选稳定版本、成熟协议，别给自己埋雷。

坑三：频繁“横跳”的隐性成本

这是我们犯过的最致命的错误：项目前三个月，换了四次模型。

第一个月用千问-72B，嫌慢换 32B；第二个月听说 DeepSeek-R1 很强，全面切换；第三个月千问出新版，又想换回去；第四个月觉得 Llama 3 也不错，又想试。那段时间，技术负责人每周跟我说“又要重新适配”，用户每周抱怨“怎么又变了”。

横跳的隐性成本，被我们严重低估了：

提示词体系要重构——我们有个投资分析智能体，提示词 3000 多字，换模型后输出格式全乱，重调一周。**输出格式要重新适配**——千问爱用表格，DeepSeek 爱用列表，下游系统是按千问格式设计的，换成 DeepSeek 后解析逻辑全失效，重开发两周。**团队学习曲线要重来**——一个投资经理抱怨：“我刚学会怎么用千问写报告，怎么又换了？”**系统稳定性会中断**——有一次换模型导致部分智能体输出格式变了、下游解析失败、数据丢失。那种感觉，就像你写了一整天的文档没保存，突然断电。

那一刻我才真正明白：**稳定性比性能更重要**。在 AI 项目里，最贵的从来不是算力，是稳定性。频繁换模型，就像在高速上边开边换轮胎——不仅跑不快，还可能车毁人亡。

每换一次模型，提示词重构 1-2 周、格式适配 1-2 周、团队重学 2-4 周、稳定性重验 1-2 周——总成本至少一两个月，外加团队精力损耗。模型不是即插即用的，它要和业务系统磨合、和用户习惯磨合。频繁切换，你就永远停在“磨合期”。

后来我们定了理性的切换机制。**必须换**只有三种情况：新模型在核心场景有代际级优势（准确率 80%→95%，或速度 10 秒→2 秒）；成本降低能显著改变商业模式；当前模型出现致命缺陷。**坚决不换**也有三种：只是“听说”很强、没数据支撑；业务对输出一致性有极致要求（金融、医疗）；切换要重构大量业务逻辑、

投入产出不划算。真要换，走完整流程：评估（1-2 周真实数据测）→ 决策（评审会）→ 灰度（先在非核心场景跑，确认稳定）→ 全量（密切监控）。

一句话：**技术选型是工具主义，不是信仰投资。该换就换，但不瞎换。**

选模型，其实是系统工程

回头看，本地化部署的模型选择，根本不是“技术选型”，而是“系统工程”。

你得先看硬件有多少（硬件约束），再看哪个模型最匹配场景（模型匹配），最后还得算稳定性和切换成本（从一而终）。就像盖房子：先看地基多大，再决定盖什么房子，最后还得考虑维护成本，别动不动就改建。

这件事最有意思的地方在于：决定“选哪个模型”的，到最后竟然不是模型本身的参数排名，而是你对自己业务场景、硬件家底、用户耐心的理解有多深。

而这种理解，正是一个外部的药厂工程师给不了你的——它只能由那个蹲在你现场、被用户追着“赔我一天时间”的人，一寸一寸长出来。

下一章，我们钻进这三道坎里最底层、也最磨人的那一块：国产 GPU 适配。它表面是技术问题，骨子里是一场关于生态的长期战争。

FIELD NOTES

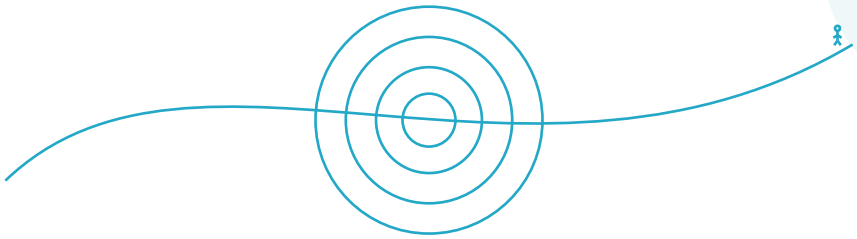
医嘱

01 模型选择是“硬件 + 模型 + 场景”的三维匹配，不是看排行榜。

02 稳定性比性能更重要，别频繁“横跳”。

03 用真实业务数据测，记住三个字：测、分、稳。

CHAPTER 10 · END



CHAPTER 11

第 11 章 国产 GPU 适配：一场被低估的生态战争

> 如果我看得更远，那是因为我站在巨人的肩膀上。
>
> ——艾萨克·牛顿

主诉 本章要回答：国产 GPU 适配，为什么是一场打不完的“生态战争”？

第 9 章我提到过那个诡异的“显卡消失”事件，当时一笔带过。这一章，我们把这个深水区彻底扒开。

先把那次经历讲完整。系统上线刚一周，一个周一上午，8 张 GPU 卡突然“消失”了 2 张。工程师排查了整整一天，最后发现是驱动和虚拟化平台的兼容性问题——服务器一重启，GPU 就“隐身”了。

“显卡消失”“模型跑不起来”“算力闲置”，这些诡异的问题，在我们半年的本地化部署里反复上演。它们背后，折射的是一个被严重低估的深水区——国产 GPU 适配。

我们出于数据主权和业务安全选了国产 GPU，这个方向我至今认为是对的。但我想诚实地告诉每一个准备走这条路的人：这不是“换个硬件”那么简单，它是一场需要“长期主义”的生态战争。

三个反复出现的坑

坑一，驱动适配的“打地鼠”。 最典型的表现是：今天能用的，明天可能就崩了。我们用一款国产 DCU 部署在某云平台上，第一次 GPU 识别失败，工程师升级驱动，GPU 回来了；紧接着原来能跑的千问-32B 报错了——新驱动和 vLLM 推理框架版本不兼容；改了框架版本，OCR 任务又跑不动。按下这个，那个冒出来。为什么？因为国产 GPU 的驱动层、框架层、应用层各自独立迭代，没有统一的版本管理规范。英伟达的 CUDA 生态磨合了多年，各层之间有明确的版本对应；国产 GPU 还在“各自为战”，协同性差。

坑二，模型适配的“追版本”噩梦。 我们一开始用千问-32B 旧版，适配了一个月才稳。两个月后官方出新版、性能明显提升，决定迁移——结果新版要求 vLLM 升到 0.5.x，而硬件厂商的适配只支持到 0.4.x。要么等厂商更新（遥遥无期），要么自己魔改源码（风险极高）。一个“看起来很简单”的版本升级，最后花了两个月。打个比方：英伟达像“万能插座”，插什么电器都能用；国产 GPU 更像“专用转换头”，每换一个电器都得重新配一个转换头。说得再狠一点——买英伟达，你买的是一张卡；买国产 GPU，你买的是一场需要陪跑很多年的婚姻：付钱那一刻只是领证，真正的日子，都在后头。

坑三，多卡协同的“资源错配”。 8 卡服务器，本以为算力足够。实际上大模型要 4 卡才能启动（不支持 2 卡、3 卡），小模型和 OCR 要 2 卡，剩下 2 卡无法被调度，长期摸鱼——25% 的算力浪费。最可怕的不是买错硬件，是你根本不知道自己买错了。

三个坑背后，是三个深层矛盾

把这三个具体问题往上提一层，会看到三个结构性的矛盾。

第一，生态差距：从“产品”到“生态”。 英伟达的核心竞争力，从来不是 GPU 卡本身，而是 CUDA 这个生态。CUDA 就像 Windows 操作系统，GPU 硬件像电脑硬件——真正让你离不开的，不是 CPU 或硬盘，而是上面跑的所有软件。英伟达花了近 20 年构建出一个正循环：开发者用 CUDA 开发 → 软件越来越丰富 → 用

户必须买英伟达 → 份额扩大 → 开发者更愿意用 CUDA。国产 GPU 还在“产品阶段”，这个正循环还没建起来。生态不是一天建成的，也不是砸钱就能砸出来的。

第二，多生态并存：各自为战的协同困境。 英伟达的生态是“中心化”的，NVIDIA 统一制定版本规范，硬件、框架、应用都在这个规范下协同。国产 GPU 是“去中心化”的，不同厂商各有软件栈、版本逻辑互不兼容。你同时用多个国产硬件时，就会发现“这个驱动 A 支持、框架 B 不行；换框架 C、应用 D 又崩了”。中心化牺牲了多样性，换来协同性；去中心化鼓励创新，却抬高了协同成本。生态早期，去中心化的代价往往被低估。

第三，资源调度：从“静态规划”到“动态匹配”。 传统 IT 的思维是静态规划——算一下要多少算力，买就完了。但 AI 场景是动态的：大模型推理高峰要全部算力，夜间训练也要全部算力，白天轻量任务只要一部分。简单“堆硬件”，很容易“高配低效”。真正的挑战不是算力够不够，而是算力能不能被合理调度。

适配的“三层模型”

基于这半年的实践，我把国产 GPU 适配总结成三层：

第一层，硬件适配层（硬骨头）——驱动与虚拟化平台的兼容、驱动与推理框架的版本对应、硬件本身的稳定性测试。特点是：只能硬啃，没有捷径。

第二层，模型适配层（追版本）——模型版本迭代要重新适配，推理框架升级要重新适配。特点是：永远在追赶，永远在路上。

第三层，应用适配层（场景化）——按业务特点选模型、按模型特点配资源、按资源特点设计调度。特点是：因地制宜，没有标准答案。

对应的解法，还是那套长效机制，浓缩成三个字：**慢、记、留**。

慢——别指望一周搞定，预留 2-3 个月适配时间。

记——建适配知识库，每个坑和解法都记下来。不是随便记记，而是按固定格式建表：硬件组合 + 框架版本 + 模型版本→出现什么问题→怎么解决的。比如“某国产 DCU + 某云平台 + 千问 32B = 驱动需 ⩾x.x.x，否则 GPU 识别失败”。这样后来的人不用从头踩坑，换模型时也有据可查。

留——换模型前先在测试环境跑 1-2 周，验证三个指标：稳定性（7×24 小时运行无崩溃）、性能（推理延迟和吞吐量符合预期）、兼容性（与现有应用无冲突）。硬件留余量，适配周期别卡死上线日期。

还有一条实战心得值得一提：如果你有条件，可以先用英伟达 GPU 快速验证业务价值，证明 AI 确实能帮到业务，再逐步切换到国产 GPU。这叫“先验证再深耕”——别在适配泥潭里打转的时候，连自己到底要不要做这件事都还没想清楚。

适配的本质，是建生态

最后我想说说对“适配”两个字的理解。

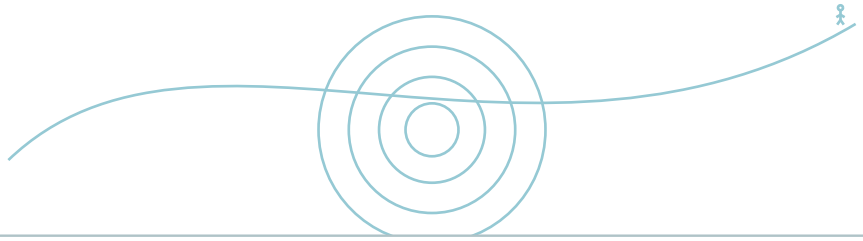
适配不是“修 bug”，而是“建生态”。

CUDA 不是天上掉下来的，是无数开发者、硬件厂商、应用厂商二十年协同迭代才建成的。国产 GPU 现在处在“补课”阶段，需要时间、投入，更需要耐心——这不是某一家公司、某一个产品能解决的，而是要整个行业一起参与。

所以如果你在适配上栽了跟头，别沮丧。不是你不行，是这场战争本身就难。你的对手不是英伟达这块卡，而是它过去二十年建起来的整个生态体系。

但也正因为难，才值得做。而且我想强调一点：能在这种“打地鼠”“追版本”的泥潭里蹲下来、把每个坑记成知识库、把数据主权和落地效果同时扛住的人，正是这个时代最稀缺的现场医生。这活儿没有光环，却是国产 AI 真正落地绕不过去的地基。

地基讲完了。下一章，我们要面对这套系统最尴尬、也最深刻的一刻——它终于稳定跑起来了，可是，没人用。



FIELD NOTES

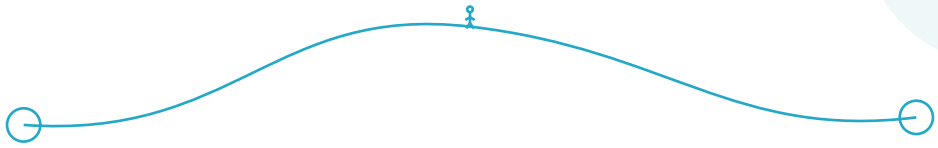
医嘱

01 适配不是“修 bug”，是“建生态”，要慢、记、留。

02 国产生态去中心化，协同成本被严重低估。

03 预留 2–3 个月、建适配知识库、换模型先测。

CHAPTER 11 · END



CHAPTER 12

第 12 章 不是跑不起来，而是用不起来

> 系统稳了、模型跑了，可没人用——这才是真正的坎。
>
> ——付伟

主诉 本章要回答：技术全过关了，为什么还是没人用？

项目启动三个月，我们拿到了第一份使用数据。

日活寥寥。

同事们更愿意去问人，而不是打开我们的 AI 系统。做了几十个智能体，真正每天还在持续用的，只有两个。

前三章，我们一路在解决“能不能跑起来”——硬件适配的打地鼠、模型选择的三维匹配、国产 GPU 生态的长期主义。那些坑大多是技术的坑，踩过去就有经验。可这一次，系统稳了，模型跑了，坑却换了一种形状：没人用。

这是整个项目对我冲击最大的一刻。它逼着我承认一件事：前面所有的技术功夫，如果过不了“有没有人用”这一关，全部归零。

复盘下来，“没人用”背后是三个错位：把“功能”当成了“需求”；把“模型问题”当成了“策略问题”；把“有知识”当成了“会使用”。

错位一：以为“做了就会有人用”

启动时我们快速搭了大量智能体，覆盖投后管理、文书写作、数据分析。当时想法很直接：功能做全了、知识库建好了，同事自然会用。

错在三个地方。

入口太深。 同类功能我们做了好几个智能体，同事不知道点哪个。有人说了句很扎心的话：**“我得先想清楚我要干什么，再去选哪个智能体——这个思维成本，比直接问同事还高。”**

和工作习惯割裂。 智能体是独立系统，跟原有工作流没有连接。打开 AI 是一个额外动作，不打开也不影响工作，那自然就不打开。

核心场景没打透。 功能虽多，被持续使用的只有两个，其他要么场景太边缘、要么体验不够好，用一次就不再来。

如果重来一次，我会先花两周做深度访谈，不问“你需要什么功能”，而问“你工作里最烦的事是什么”。大多数项目，一开始就把这个顺序搞反了——**不是功能越多越好，而是核心场景越深越好。**

错位二：以为换个更强的模型就能解决“答不准、查不全”

用户用了一阵，反馈最多的是“答不准、查不全”。我们第一反应是模型不行，想换个更强的。

但仔细一分析，问题不在模型，在策略。

用户的问题其实是几种完全不同的类型：有人问“某项目最新状态是什么”，这是纯知识检索，要精准召回；有人问“各类制度里关于采购的要求有哪些”，这是跨文档查询，要多路召回；有人说“帮我写份投资分析报告”，这是结构化生成，要结合知识库做框架输出。而我们当时用同一套策略处理所有问题——简单问题答不准，复杂问题答不全。还有个时间维度的盲区：用户提问常带时间限定，但知识库没做时间元数据标注，召回后没法过滤，用户还得自己再筛一遍。

一线的反馈比任何分析报告都直接。有同事说：“明知道它不应该给出量化的结论，它偏偏幻觉了一个很坚定的百分比。”还有人说：“做了这么多智能体，每个都有角色设定，但感觉写不写那个设定，出来的东西差不多。”更让人头疼的是一致性问题——同一个问题问两遍，答案不一样；你换一种问法，它就顺着你的意思说。

这些反馈拼在一起，指向同一个结论：**越是本地化部署，越要精心打磨提示词工程，这才是拉开落地效果的关键**。本地模型不缺能力，缺的是“被约束和引导”——它通用能力相对有限、训练数据相对陈旧，更依赖提示词去对齐具体业务。

我们总结出提示词的三重不可替代的价值：

补知识——做模型的外挂大脑。本地模型不知道我们内部的风险评分标准、业务术语、最新制度。有次同事让它评估一个项目的风险等级，给出的结论看起来合理，却完全不符合内部口径。后来把风险维度权重和输出格式写进提示词，立刻就对了——缺的从来不是模型能力，而是“把规则说清楚”。

立边界——做模型的刹车系统。小参数模型任务边界模糊。有同事让它整理会议要点，结果洋洋洒洒把“整理”做成了“再创作”。后来在提示词里写明“仅提取核心观点、按重要性最多列五点、每点不超过五十字、要点式输出”，一次解决。

给路径——做模型的导航仪。复杂推理时本地模型容易做一半卡住。让它分析财务异常，它能找出异常却说不清原因。后来用思维链提示词把任务拆成步骤：先提关键指标、再算变化幅度、再分析原因、最后给建议——强制它一步步走，输出质量明显提升。

这里还有个插曲值得说：报告写作是强需求，但它暴露了一个更本质的问题——模型不是不会写，是不会“按你的方式改”。用户要的是按去年的模板替换数据，系统给的是从头重写一份。提示词能部分收束，但这个场景对模板理解要求太高，目前还在优化。我们后来做了个主动判断：把这个场景降优先级，把精力集中到提示词能真正发挥作用的地方。接受局限，是为了把时间用在刀刃上——这也是一种取舍。

错位三：以为知识库建好就完事了

第 9 章讲过知识库运营的“清、录、看”，这套机制解决了“库里有没有内容”。但一年下来，我们撞上另一个问题：

知识库里有东西，同事也知道有，就是不会用。

不是不愿意用，是不知道该怎么问。问得太宽泛，一大堆答案；问得太具体，又怕找不到。于是很多人放弃了，回去问同事。

我们后来做了一件事：给高频问题预设提示词模板，点开就能用，不用自己琢磨怎么提问。

这件事让我把思路彻底掉了个头：**不是“怎么让用户学会用 AI”，而是“怎么让 AI 主动降低用户的使用门槛”。** 前者费力，后者省力；前者是培训问题，后者是产品问题。

再说透一层：过去我们总以为，系统没人用，是因为员工不会用，所以拼命搞培训。但这一年走下来，我彻底想明白了——系统没人用，是因为 AI 还不够懂员工。**培训，是让人去适应机器；预设模板，是让机器去适应人。现场医生要干的，是让机器低头，而不是让人弯腰。**

走完这一年

这一年最大的转变，是我们终于开始少问“模型够不够强”，多问“我们有没有把业务说清楚”。

大多数 AI 落地项目真正的瓶颈，不是技术不行，而是问题没有被定义清楚。技术问题可以靠工程解决，使用问题本质是认知问题——你得先搞清楚用户真正的麻烦是什么，再去想 AI 能帮上什么忙。

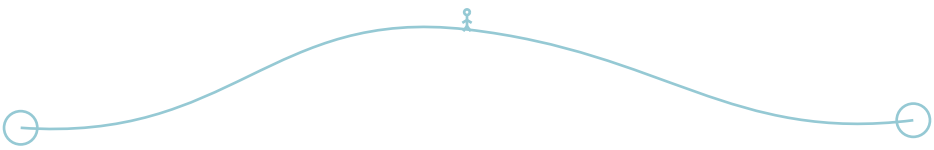
本地化部署的核心价值，从来不是“拥有一个模型”，而是“拥有一个懂你业务的 AI”。而提示词，就是让 AI 懂你的那个翻译器。

写到这儿，第三篇的故事就完整了。从选硬件、调驱动、挑模型，到最后蹲下来一句句打磨提示词、给用户预设模板——这一整条链路上真正起决定作用的，

从来不是某一项技术有多先进，而是有没有一个人，愿意把“技术”和“现场”之间那段最难、最琐碎、最不性感的距离，一寸一寸走完。

这个人是谁？他需要什么样的本事？普通人有没有可能成为他？

这正是下一篇要回答的问题。



FIELD NOTES

医嘱

01 三个错位：把功能当需求、把模型问题当策略问题、把“有知识”当“会使用”。

02 本地模型缺的不是能力，是被约束和引导——提示词是翻译器。

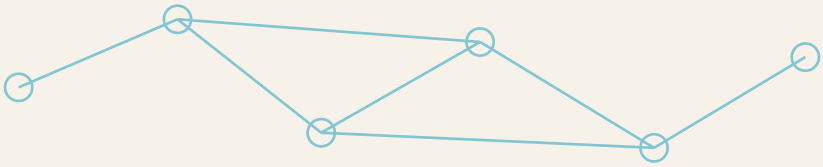
03 少问“模型够不够强”，多问“业务说清楚没有”。

CHAPTER 12 · END

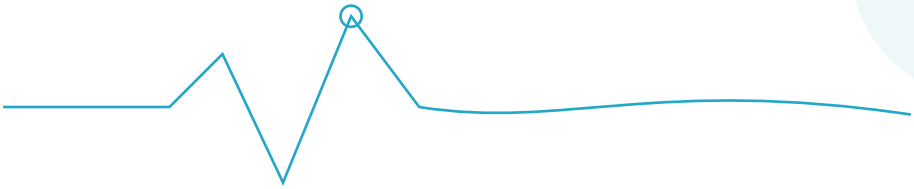
第四篇

谁能做

需要什么本事



AI 把技术门槛压低了，把判断力的价值抬高了。



CHAPTER 13

第 13 章 解耦能力：贯穿始终的那条线

> 把每一个难题，尽可能分解成解决它所必需的若干部分。
>
> ——勒内 · 笛卡尔《方法论》

主诉	本章要回答：现场医生最核心、最难复制的能力是什么？

这本书讲了好多概念，如果让我挑一个最想让你记住的，我会选这个词：**解耦**。

它是现场医生最核心、也最难被复制的能力。三步法是招式，解耦是内功；没有它，招式都是空的。

什么是解耦

先讲一个画面。

陈澄波和我聊他做项目的状态时，说过一句话，我印象很深。他说，当他真正理解了一个业务场景之后，脑子里会自然地浮现出一套东西——用什么样的 AI 工具组合、搭什么样的工作流、在哪个环节嵌入模型能力，来解决这个问题。

这个“自然浮现”的能力，就是解耦。

更准确地说：**解耦，就是不让一个业务问题被一种工具绑死。**

很多人做 AI 落地，是反过来的——手里有把锤子（大模型、RAG、Agent），看什么都是钉子。客户来一个需求，不管三七二十一，先想“我这工具怎么套上去”。这叫工具驱动，本质是“被工具绑架”。

解耦的人不这样。不绑死，不硬套——他先把业务问题本身拆开，拆到一个个颗粒度极清的小环节，再问：这个环节，到底需要什么？

解耦，到底在“解”什么

我用一个保险核保的例子来说明（这个案例第五篇会完整讲）。

一家保险公司的销售，在联系总部核保之前，想先自己做个初步判断。但险种多、条款交叉，查半天也拿不准。表面上看，这是一个需求：“做一个能帮销售判断核保定级的 AI。”

如果你被工具绑架，第一反应可能是：“上一个大模型，把条款喂进去，让它回答。”

但解耦的人会先把这个场景拆开，看清楚里面其实藏着好几类完全不同的子问题：

- 有些环节，是要从一堆相似条款里**精准检索**出最相关的那几条——这适合检索增强（RAG）；
- 有些环节，是几个硬性条件的**规则匹配**——用确定的规则引擎反而比模型更稳、更可解释；
- 有些环节，是要在多个条件叠加下做**综合推理**——这才真正需要大模型的推理能力；
- 还有些查询，类型完全不同，需要**分流**给不同的 Agent 去处理。

你看，“做一个核保 AI”这一个笼统的需求，被解耦成了“哪些环节用检索、哪些用规则、哪些交给模型推理、哪些需要多 Agent 分流”。然后，在每一个环节，用最合适的工具顶上去。

这就是解耦：把一个复杂问题，拆成一组可以分别用最合适手段解决的子问题，再重新组装成一个能跑的整体。

解耦的三个层次

陈澄波后来把解耦这件事，拆成了三个递进的层次。我觉得这个拆法，比任何理论框架都更贴近现场。

第一层，锁定问题（诊断期）。 专家提出业务假设，和 AI 高强度交流验证。这一层客户几乎看不到 AI 的产出物——它藏在诊断过程里。但它最有力量，因为它大幅缩短了从“模糊的痛”到“清晰的卡点”的距离。第三篇里我讲的“先花两周做深度访谈，不问'你需要什么功能'，而问'你工作里最烦的事是什么'”，干的就是这层的活。

第二层，开发落地工具（造兵器）。 发现现成工具不够用，利用 AI 辅助编程生成或匹配开源工具，自己开发精简的定制化工具。客户还是看不到 AI——但 AI 已经完成了赋能。陈澄波在保险核保项目里自己造文档智能切片工具，冯晓刚在私域运营项目里造“AI 自动调试”机制，都是这一层。

第三层，嵌入业务流程、交付产品（上战场）。 最终交付客户可以直接感知的 AI 产品。这是所有人都能看到的那层，但直接跳过前两层只做这一层，往往效果不佳——因为你还没搞清楚问题到底卡在哪儿、手里也没有趁手的工具，硬上产品，大概率是花架子。

三层递进，每一层都依赖前一层的积累。跳过锁定问题直接做工具，是盲人摸象；跳过前两层直接做产品，是空中楼阁。而解耦，就是让你在每一层都能看清“这一层到底该干什么”，不慌不忙地一步步走上去。

为什么解耦这么难，又这么值钱

解耦难，难在它不是一个技术动作，是一个“先懂业务、再懂工具”的复合判断。

你得足够懂业务，才能把场景拆到正确的颗粒度——拆粗了，问题还是糊在一起；拆错了，每个环节都用错工具。你又得足够懂工具的脾性，才知道 RAG 擅长什么、规则引擎稳在哪里、模型推理的边界在哪里、什么时候该上多 Agent。这两样缺一不可。

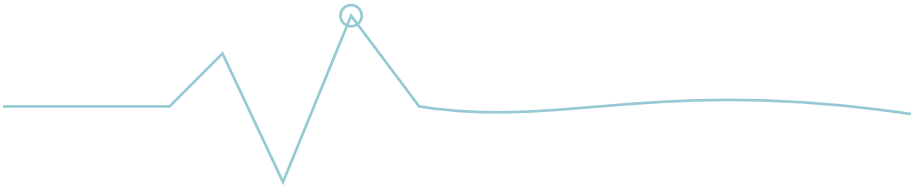
而这恰恰是大模型公司给不了的能力。药厂的工程师懂模型，但他不懂你这个行业的病灶藏在哪个环节；纯业务顾问懂行业，但他不知道每个环节该配什么工具。解耦能力，长在这两者的交叉点上，只能在一个又一个真实项目里泡出来。

我在第三篇讲本地化部署时，其实通篇都在做解耦，只是没点破：把整个系统拆成“大模型基础能力 / 工作流编排 / 存储输出”三段，每段配不同方案做组合验证——这是对技术栈的解耦；把用户的提问拆成“纯检索 / 跨文档查询 / 结构化生成”三类、分别用不同策略——这是对需求的解耦。

解耦能力为什么贵？因为它无法速成，也无法被一份文档复制。它是经验密度的产物——你拆过的场景越多、踩过的工具的坑越多，那个“自然浮现”的速度就越快、越准。

这也正是为什么我反复说，现场医生的护城河不是“懂模型”，而是“懂行业的病”。懂病，才知道该往哪儿下刀；解耦，就是下刀的手法。

理解了解耦这条贯穿始终的内功，下一章我们就来看它具体施展出来的三个招式——现场医生干活的三个阶段：福尔摩斯、钢铁侠，和调音师。



FIELD NOTES

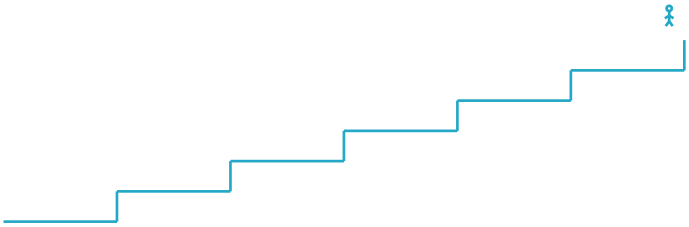
医嘱

01 解耦 = 把复杂问题拆成可分别用最合适手段解决的子问题。

02 先懂业务、再懂工具，两样缺一不可。

03 它无法速成，是经验密度的产物。

CHAPTER 13 · END



CHAPTER 14

第 14 章 三步法：福尔摩斯、钢铁侠、调音师

> 没有掌握全部证据就开始推理，是一个致命的错误。
>
> ——阿瑟·柯南·道尔《血字的研究》

主诉 本章要回答：现场医生到现场之后，是怎么一步步把 AI 落下去的？

现场医生干一个项目，大体分三个阶段。

这个三阶段框架，主要是陈澄波提出来的。他在多个项目里反复验证，提炼出了这套从诊断到原型到打磨的完整路径。我在自己的项目里跑下来，发现骨架完全一致，只是我的说法和他的略有不同。我喜欢用三组“人 + AI 搭档”的比喻来讲，因为它能同时说清两件事：每个阶段人在干什么，以及 AI 在这个阶段扮演什么角色——而 AI 的角色，是一阶段一阶段变化的。

这三组搭档是：福尔摩斯与华生，钢铁侠与贾维斯，调音师与钢琴。你可能注意到了，前两组都是“人和 AI 助手”，第三组却变成了“人和一件乐器”——这恰恰是关键：前两个阶段，AI 是给你打下手的“助手”；到第三个阶段，AI 本身成了那件被你反复打磨的“物”。

上一章讲了解耦的三个层次——锁定问题、开发工具、嵌入流程。三步法和那三个层次几乎是同构的，只是视角不同：解耦讲的是“每一层该干什么”，三步法讲的是“每个阶段人和 AI 怎么配合”。

第一阶段：福尔摩斯与华生——现场诊断

第一阶段做的是侦探的活。

客户在会议室里说出来的需求，跟他们真正卡住的地方，经常是两回事。老板说想降本，现场可能卡在排产；业务说想提效，真正的问题可能在数据口径。所以这个阶段，现场医生不能坐在会议室里听汇报，得到真实的业务现场去看真实的操作，找那个最重复、最耗人、最容易算清账的卡点。

这时候，AI 是华生。

华生很能干——帮你快速搜集数据、跑验证、把一个假设几小时内就核对一遍。但福尔摩斯才是那个提出假设、判断哪个方向值得深挖的人。人负责提出业务假设，AI 负责快速验证假设，这样能把过去要几周的诊断周期，压缩到几天。但请注意：判断“哪个方向值得深挖”，靠的不是 AI，是人在行业里积累的经验。

这个阶段的产出，是一份明确的问题诊断和落地方案——不是华丽的 PPT，是“该先改哪条流程、为什么是它、改了能算出多少账”。

第二阶段：钢铁侠与贾维斯——快速原型

方向定了，第二阶段做的是钢铁侠的活。

钢铁侠厉害，不只是因为他有贾维斯，更因为他自己会下场造、会亲手装。这个阶段的关键动作是：不等正式排期，用 AI 辅助编程工具和现成组件，几天内搭出一个能跑的原型。

为什么要快？因为原型的目的，不是交付一个完美的系统，而是让客户和团队看见。我在前面说过一句话，值得在这里重复：没见过的人只能想象，看到了才能判断——方向对，继续走；偏了，马上调。哪怕这个原型背后还有大量人工兜底，也要先让客户看到“原来可以做成这样”。

这里有个常被误解的点：这个阶段的人，不是只输出方案不管落地的顾问。他要亲自明确算法选型、给出可运行的 Demo、指导企业的技术团队落地。贾维斯

（AI）能帮他飞快地写代码、改 bug，但“这个原型该长成什么样、走哪条技术路线”，是钢铁侠自己拍的板。

第三阶段：调音师与钢琴——打磨内核

到第三阶段，事情发生了一个本质变化。

前两个阶段，AI 是帮手——帮着搜数据、搭原型、写代码。到这个阶段，AI 的角色变了：**它不再是工具，它本身就是交付物的核心**。

就像一架钢琴。前两个阶段，你是在选琴、搬琴、试琴——AI 还是个“待调校的乐器”。到这个阶段，琴就位了，但它能不能弹出好曲子，全看调音师怎么调。每一个键的音准、每一个踏板的灵敏度、高低音区的平衡——调音师一锤一锤地敲、一遍一遍地听，直到这架琴在任何一个音域都经得起演奏。

不管你最终交付的是一套风控审核系统，还是一个销售陪练工具，它的内核都是同一个东西：一个经过反复打磨的“AI 大脑”。同样的模型、同样的框架，调出来是天籁还是走调，全看调音师怎么打磨。

这个阶段的活儿，最不性感：定义什么算错，在哪里设熔断，反复跑测试用例，盯着纠正薄弱点。一轮一轮地测、改、再测，直到它在关键环节上经得起生产环境的检验。我在第三篇里讲的“逐字逐句打磨提示词、给模型立边界、用思维链给路径”，干的就是这个阶段的活——和调音师一锤一锤地校音，本质上是一回事。

它决定生死。因为前两个阶段做得再漂亮，最后这架“琴”没调准，弹出来的就是走调的曲子。

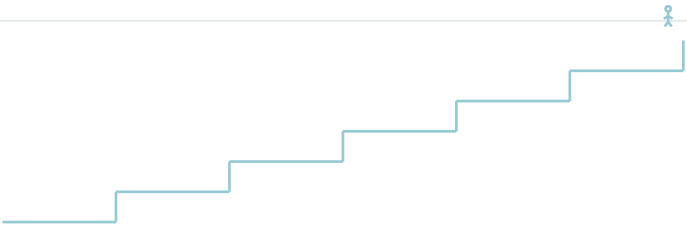
三个阶段，一条暗线

把三个阶段连起来看，你会发现 AI 的角色画了一条清晰的上升线：从华生（辅助搜证），到贾维斯（辅助建造），到钢琴（成为交付物本身）。AI 越来越深地嵌入交付的核心。

但同时，还有一条线始终没变——那个提出假设、拍板路线、为最终效果负责的人。福尔摩斯、钢铁侠、调音师，换的是场景，不变的是“判断”。

我想在澄波的框架上补一点自己的观察：这三个阶段，不只是时间上的先后，更是一种**递进的依赖关系**——没有福尔摩斯阶段的诊断，钢铁侠阶段就是盲造；没有钢铁侠阶段的原型验证，调音师阶段的打磨就不知道该往哪儿磨。很多人栽，不是栽在某个阶段做得不好，而是栽在跳过了前面的阶段。澄波自己也说过：前两个阶段是基础，没有前两个阶段，不可能交付出合格的 AI 产品。跳过蹲现场的福尔摩斯阶段，直接想做调音师——结果琴调得再准，弹的却是一首没人要听的曲子。

而把这三个阶段串起来、让人在每个阶段都用对工具的那条内功，就是上一章讲的解耦。没有解耦这门内功，福尔摩斯会诊错病，钢铁侠会造错甲，调音师会磨偏音——招式再花哨，也都落了空。



FIELD NOTES

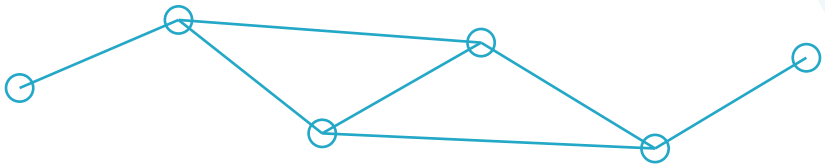
医嘱

01 福尔摩斯：到现场找最能算账的卡点。

02 钢铁侠：几天搭出原型，让人先“看见”。

03 调音师：反复打磨，AI 本身就是交付物的核心。

CHAPTER 14 · END



CHAPTER 15

第 15 章 发起、审核、决策：AI 时代人的三项核心动作

> 学而不思则罔，思而不学则殆。
>
> ——《论语 · 为政》

主诉 本章要回答：AI 接走了大部分工作，人到底还该做什么？

讲个我自己的小事。

有一次处理一件事，我口述了 10 秒钟的需求，AI 给我生成了一封条理清晰、措辞诚恳的正式邮件。我看了一遍，几乎没改，直接发了。

以前写这种邮件，少说 20 分钟——组织语言、调整措辞、检查逻辑，每一步都在消耗认知带宽。现在 10 秒口述，30 秒审核，1 秒发送。

这不是一个效率提升的故事。这是一次角色切换。

人只剩三个动作：发起、审核、决策

我后来反复琢磨这件事，提炼出三个动作：**发起、审核、决策**。

发起，是定义问题，告诉 AI 你要什么；审核，是检查结果，看 AI 给的东西哪里对哪里不对；决策，是拍板放行，为最终输出负责。三个动作各管一摊：发

起管方向，审核管质量，决策管行动。而中间所有“想”和“做”的环节——搜资料、找数据、写初稿、做对比——全部外包给 AI。

这个框架最让我兴奋的，是它的迁移性。它不只能写邮件——做市场调研（两周变两天）、审合同（AI 初筛风险点、人只审标出来的部分）、写商业计划书（技术出身的人口述逻辑、AI 成稿、自己把关表述）——任何一件知识工作，都能装进“发起—审核—决策”这套操作系统。它把一个人的能力边界，从“木桶最短板”拉到了“最长板”：你不擅长写但擅长判断，那就口述发起、让 AI 写、自己来审。

AI 时代，人的核心动作不是“干”，而是“选”。

人机分工：思、行、品

发起、审核、决策，说的是人做什么。那 AI 做什么？人和 AI 之间怎么分？

明略科技的吴明辉在混沌学园讲过一个框架（系作者现场理解，以本人公开表述为准），我觉得很清晰：AI 负责“思”，人机共同完成“行”，人负责“品”。

把它和我那“三项动作”拼在一起，你会发现两者严丝合缝地咬在一起：

- **发起，对应“思”的入口**：人定方向，AI 去搜索、整理、生成。
- **审核，对应“品”的过程**：人逐字逐句地查，看 AI 给的哪里对、哪里不对。
- **决策，对应“品”的结果**：人拍板放行，为最终输出负责。

一句话：思行品管的是“谁干什么”，发起审核决策管的是“人具体怎么干”。两套概念拼在一起，就是 AI 时代知识工作的完整图景——人发起需求，AI 思考和执行，人审核和决策。

真正的瓶颈，不在“思”，在“品”

但这里藏着一个最容易被忽略的陷阱。

以前，“思”和“品”是混在一起的。你花三天看材料、算数据、做对比，最后得出结论——这个过程里，思考和判断是交织的，你一边想，一边就在做选择。现

在“思”被 AI 抽走了，十个方案几秒钟摆在你面前，你突然发现：你选不出来了——更准确地说，你的判断力，已经跟不上 AI 的生成速度了。

举个最简单的例子：AI 给你生成了十页行业报告，你只看了结论就拍板用了，里面有一个数据是 AI 编的，你没审出来。报告发出去了，后果谁扛？AI 不会扛。

“品”才是脏活。 逐字逐句地审，逐条数据地查，逐个逻辑地过——知道 AI 给的哪里对、哪里不对，哪里该信、哪里该怀疑。这种判断力不是天生的，是在行业里泡出来的：你见过多少周期、踩过多少坑、做过多少次错误判断再修正过来——这些沉淀出来的直觉，AI 没有。

“品”对应的就是人的“审核”和“决策”——审核是过程，决策是结果，但本质都是判断。如果“品”这一关你过不了，发起审核决策就不是“调度模式”，而是“甩手掌柜模式”：你把执行权交出去了，判断力却没跟上，等于把决策权也一起交了出去。

更麻烦的是，“品”会退化。导航用多了，自己认路的能力会变差；AI 替你想多了，你的品也会钝。这不是一个装上就能跑的操作系统，而是一个需要对抗惯性才能维持的状态——你越依赖 AI 替你想，你自己的判断力就越需要刻意练习才能保住。

这一点，和前面第二章讲过的“纠错成本”其实是同一件事的两个面：第二章说的是，AI 出了错没人愿意查，是意愿问题；这一章要说的是，就算你愿意查，你的判断力够不够用，是能力问题。意愿可以靠方法解决（比如多 AI 交叉验证），但能力的退化，只能靠持续的刻意练习来对抗。

从一个人，到一支人机混合团队

当“发起—审核—决策”从一次性的尝试，变成一个人的肌肉记忆，组织的形态也会随之突变。发起、审核、决策的终点，从来不是个人效率，而是组织形态。

当每个人都从“执行者”变成“调度者”，组织的样子就变了。基层员工不再只是干活的，而要变成“Agent 管理者”——一个人指挥多个 Agent 做资料、分析、初稿、复盘。中层不再只是分配任务，而要判断“人和 Agent 的组合效率”——谁会用 AI，

谁有好的判断力（taste），谁能把 Agent 训练得越来越懂业务。连开会都会被重构：共享信息交给 AI，人开会只做三件事——讨论冲突、做出判断、改变行为。

我自己团队就在这么干。我们有个叫“AI 先锋队”的飞书群，9 个真人配 8 个机器人，日常运转已经离不开 AI。这些“数字员工”要真正顶用，靠三个核心能力：**统一入口**（同一个入口能调用不同技能解决复杂任务）、**用户层长期记忆**（针对单个用户养成使用习惯，越用越贴合）、**可调用的封装技能**（把工作经验包装成技能，可复用）。

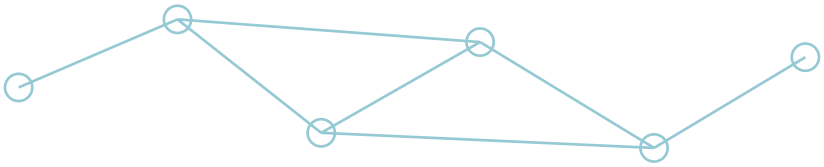
再往上一层，是一种“矩阵式”的智能体架构：纵向上，各职能部门负责训练对应职能的数字员工；横向上，项目组带着这些数字员工去完成具体业务目标；而人，只在关键节点做决策。一句话——数字员工并行执行，人类只在岔路口拍板。

这不是未来，是正在发生的事。从写邮件到市场调研，从合同审核到健康管理，一个场景一个场景跑通，每个样本都在回答同一个问题：当 AI 把“思”接过去之后，人的“品”还够不够用？

如果你的答案是“够”，发起审核决策就是你的操作系统；如果是“不太确定”——那就更该现在开始练。因为“品”这个东西，不会因为你不用 AI 就自动变强，但会因为你不练而持续退化。

AI 能帮你做到七八十分，最后那二十分，是人的尊严，也是人的护城河。这条“品”的边界到底在哪里，我会在第五篇那个保险核保的案例里掰开揉碎地讲——你会看到，决定一个项目生死的，恰恰是那最后二十分的判断力。

但在那之前，有一个更迫切的问题：一个刚入行、还没攒下十几年判断力的新人，难道就只能干等着？恰恰不是。下一章，我们就来看，AI 怎么让一个新人，更快地把“品”练出来。



FIELD NOTES

医嘱

01 人只剩三个动作：发起、审核、决策。

02 人机分工是“思、行、品”：AI 担“思”与部分“行”，人担“品”。

03 真正的瓶颈不在“思”，在“品”——而“品”会因为不练而退化。

CHAPTER 15 · END

CHAPTER 16

第 16 章 新人如何用 AI 少走三个月弯路

> 敏而好学，不耻下问。
>
> ——《论语 · 公冶长》

主诉 本章要回答：没有十几年积累的新人，怎么用 AI 快速入行？

上一章讲了发起、审核、决策——AI 时代人的三项核心动作。这一章我想把这个框架落到一个更具体的场景上：一个新人，怎么用 AI 把自己的成长曲线从半年压缩到三个月。

为什么把这一章放在“谁能做”的第四篇？因为“品”才是最难的脏活——不是每个人都能过这一关，但每个人都可以从现在开始练。而新人，恰恰是练“品”最好的起点：你还没形成固化的工作惯性，AI 的介入反而不会遇到老员工的排异反应。

先讲一个让我印象很深的故事。

一个实习生的三个月

前段时间，团队来了个实习生小何。她学的不是技术，但上手特别快——不是因为她天赋异禀，而是因为她几乎是从第一天起，就把 AI 当成了自己的“第二大脑”。

第一个月，她用 AI 做了三件事：把所有会议纪要自动整理成结构化笔记、把领导口头交代的任务拆成可执行的行动清单、把读到的行业报告提炼成一页纸摘要。这些事以前没人要求她做，但她主动做了——因为她发现，AI 让这些“额外动作”的成本低到几乎可以忽略。

第二个月，她开始用 AI 做更复杂的事：给一份投后管理报告生成初稿、帮同事做数据对比分析、甚至自己搭了一个小智能体来自动追踪项目进度。到第二个月底，她已经能独立完成一些以前需要半年经验才能上手的工作。

第三个月，她做了一件让我意外的事：她开始“教”AI。不是教 AI 新知识，而是把自己在项目里积累的判断标准、常见错误、注意事项，写进提示词模板，让 AI 下次自动按她的标准来。她其实在做“品”的沉淀——把自己的判断力，变成可复用的规则。

三个月后，小何的工作产出，已经接近一个有一年经验的员工。不是她比别人聪明，是她比别人更早地把“发起—审核—决策”这个操作系统装进了自己的工作流。

新人的三个阶段，和三步法同构

你有没有发现，小何的三个月，和第十四章的三步法几乎完全同构?

第一个月是福尔摩斯阶段——用 AI 帮自己快速了解业务、验证假设，找到自己在这个组织里最该发力的方向。第二个月是钢铁侠阶段——用 AI 辅助自己快速产出原型、完成任务，让团队看到自己的能力。第三个月是调音师阶段——开始打磨自己的判断标准，把“品”沉淀成可复用的规则。

这不是巧合。三步法不只是现场医生做项目的方法，也是任何一个人在 AI 时代快速成长的路径。

为什么新人反而更快

这里有个反直觉的地方，值得点破。

老员工有经验，但经验有时候反而是负担——你已经知道“怎么做”，所以很难接受“换一种做法”。新人没有这个包袱，AI 给她一个新方法，她直接就用，不存在“和旧方法比较”的心理成本。

更重要的是，新人天然就在做“发起”——她不知道答案，但她知道怎么问。而“知道怎么问”，恰恰是 AI 时代最值钱的能力。老员工有时候反而不太会问——因为他觉得自己应该知道答案，不好意思问“蠢问题”。但 AI 不在乎你的问题蠢不蠢，它只在乎你问得清不清楚。

所以，如果你是一个新人，或者你在带新人，我的建议是：别急着学“怎么做”，先学会“怎么问”。发起的能力，比执行的能力，更早、也更值得练。

“品”的练习，从第一天开始

但我也必须说一个容易被忽略的风险。

新人用 AI 太顺，容易跳过“品”的练习——因为 AI 的输出看起来总是“差不多对”的，而新人没有足够的经验去判断“差不多”和“真正对”之间的距离。

所以对新人，我有一个特别具体的建议：**每次用 AI 产出的东西，都花 5 分钟做一次“品”的练习**。不需要逐字逐句查，只需要问自己三个问题：这个结论我信不信？哪个数据我拿不准？如果错了，后果谁扛？

这三个问题，就是“审核”和“决策”的最小练习。你不需要每次都答对，但你必须每次都问。问多了，“品”的感觉就会长出来。

而能帮新人设计这种“品”的练习、在旁边看着她一步步把判断力练出来的人——他干的，正是现场医生的活。

下一章的“三个月落地作战地图”，就是给这条路上的关键节点标上路标。

FIELD NOTES

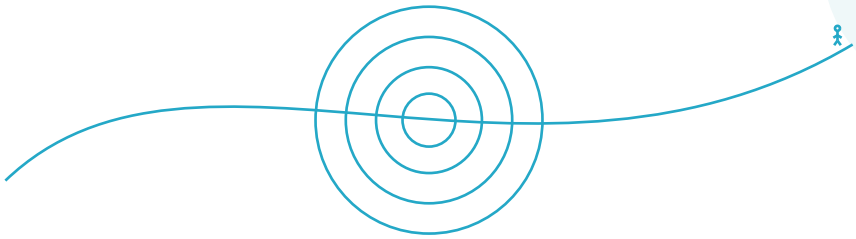
医嘱

01 新人最大的优势，是没有旧惯性——AI 给个新方法，直接就用。

02 先学“怎么问”，再学“怎么做”：发起的能力，比执行更早、更值得练。

03 每次用完 AI，花 5 分钟练“品”：这结论我信不信？哪个数据拿不准？错了谁扛？

CHAPTER 16 · END



CHAPTER 17

第 17 章 三个月落地作战地图

> 凡事豫则立，不豫则废。
>
> ——《礼记 · 中庸》

主诉 本章要回答：一个 AI 项目从进场到见效，要熬过哪三个月？

方法论讲到这里，还差最后一块：把它装进时间轴。

一个 AI 项目从进场到见效，不是一条平滑上升的曲线，而是一条“先兴奋、再失望、最后才回报”的微笑曲线。最大的危险不在技术，在于——很多老板在曲线的最低点，把项目叫停了。

我把这条曲线拆成三个月、三个阶段。每个做过 AI 落地的人都会认得它们。

第一个月：祛魅期

第一个月，要做的是“祛魅”——把对 AI 的浪漫想象，落成扎扎实实的工程准备。

这个阶段的核心动作，都是地基活：选模型，配 API Key，搭数据存储，打通系统，配置基础的安全防护。然后——这一点很关键——在一个隔离的沙盒环境里测试，验证三件事：边界有没有效（AI 会不会越权、会不会泄露）、成本可不可控（一个真实场景跑下来烧多少钱）、基础安全防护到不到位。

祛魅期的意义，是把“AI 无所不能”的幻觉，换成一张“它能干什么、不能干什么、干一次多少钱”的清单。这个阶段最忌讳的，是跳过沙盒、直接拿真实业务数据上生产——那等于地基没打就盖楼。

第二个月：阵痛期

第二个月，是最难熬的阵痛期，也是项目的“死亡之谷”。

“死亡之谷”这个词，不是我的发明。做 AI 转型咨询的李博士和他的模驭 AI 团队（以下对其研究的转述系作者理解，具体数据与表述以原作公开发表为准），在系统分析了近百个失败项目后，给了一个很精确的定义：**技术能力与实际价值之间的巨大断层**。据他们的观察，相当一部分 AI 项目都掉进了这条谷——钱花了，系统做了，就是没产生价值。

李博士团队还提出了一个公式，我觉得特别到位：

> AI 价值实现 = 技术能力 × 采纳转化率 × 效用转化率

注意是乘法，不是加法。这意味着，任何一项为零，整体就趋近于零。技术再强，没人用（采纳率为零），价值就是零；有人用了，但没有转化成业务结果（效用率为零），价值还是零。

他们还总结了项目掉进死亡之谷的三个最常见的原因：**技术指标崇拜**——过度关注算法精度，忽视组织适配；**单一工具思维**——把 AI 当独立工具，没有嵌进上下游生态；**价值评估缺失**——重初始投入，轻长期价值测量。

你看，这三个原因，没有一个是在说“模型不够强”。它们说的全是同一件事：**技术落地的本质不是技术革命，而是组织革命；不是能力建设，而是价值转化**。这也是李博士论文的核心结论。

回到我们的项目。接入真实业务数据之后，问题果然集中爆发：幻觉率高，成本超支，结果偏差。团队里开始有人怀疑——“是不是方向错了？”“是不是这玩意儿根本不行？”我在第三篇讲的那些坑，大多在这个阶段冒头：答不准、查不全、日活寥寥、用户回去问同事。

这些症状，用李博士的框架来看，恰恰是采纳转化率出了问题——系统做了，但没人愿意用、或者用了没效果。不是技术不行，是组织还没适配。

这个阶段，对老板是一场考验。

我想对每一个推 AI 的老板说一句最实在的话：**阵痛期，千万不要中途叫停。**

你要接受 AI 智能体有一条“学习曲线”。它一开始就是不够好——幻觉要靠反复调提示词降下来，成本要靠优化和路由压下去，偏差要靠交叉验证和反馈闭环收敛。这些都不是一次到位的，是一轮一轮磨出来的。第三篇里“多 AI 交叉验证”“提示词三重价值”“动态蓄水池”，全都是阵痛期里被逼出来的解法。

很多 AI 项目就死在这里——不是因为技术真的不行，而是因为决策者在最低点失去了耐心。Gartner 说一半的生成式 AI 项目在 PoC 后被放弃，相当一部分，倒在了阵痛期的山脚下。

第三个月：见效期

熬过阵痛，第三个月会迎来见效期。

这个阶段的标志，是项目终于形成了一个**可复制的落地模式**：哪个场景用什么模型、配什么提示词、怎么嵌进流程、谁来维护——这一整套打法稳定下来了。于是可以从单点试点，走向规模化推广：把第一个跑通的场景，复制到第二个、第三个。

见效期真正交付的，不是“一个能用的 AI”，而是“一套企业能持续把 AI 用起来的能力”——这恰恰呼应了全书开头那句话：FDE 交付的不是一个 AI 工具，而是一个企业持续使用 AI 的能力。

为什么作战地图本身就是价值

讲这张地图，不只是为了给个时间表。它本身就是现场医生价值的一部分。

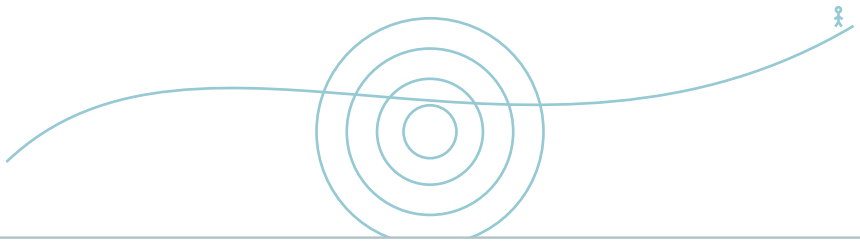
因为对客户来说，最可怕的不是过程艰难，而是“不知道现在走到了哪一步、这种难是正常的还是要完蛋了”。一个有经验的现场医生，能在第二个月所有人都

动摇的时候，告诉老板：“这是阵痛期，正常的，再给我两周。”——这句话本身，就值很多钱。它是用无数个项目的教训换来的判断力。

这也对应着第一章那组数据：88% 的企业用了 AI，只有 6% 用出了结果。这中间的差距，很大一部分，就是“有没有人能带着你走完这条微笑曲线”。

现场医生，就是那个陪你走完三个月、并且在最低谷托住你的人。

下一章，我们聊一个每个企业都会面对、却很少有人认真想清楚的问题：当 AI 智能体真的被放进生产环境，风险在哪里，价值又在哪里。



FIELD NOTES

医嘱

01 第一月祛魅：搭环境、配安全、沙盒验证。

02 第二月阵痛：幻觉、超支、偏差，千万别中途叫停。

03 第三月见效：形成可复制模式，从试点到规模化。

CHAPTER 17 · END

CHAPTER 18

第 18 章 企业用 Agent 的风险与价值

> 我们必须先确定，交给机器的目标，正是我们真正想要的目标。
>
> ——诺伯特 · 维纳

主诉 本章要回答：企业把能自己干活的 Agent 放进生产环境，会踩哪些坑、又该往哪切？

前面几章讲的方法论，偏“道”。这一章补一块特别接地气的“术”——当企业真的把一个能自己干活的 AI 智能体放进生产环境，会踩到哪些要命的坑，又该往哪些地方切。

这一章的内容，一部分来自得到的一场对谈（快刀青衣对谈极客邦尹会生），一部分来自我自己的体会。我把它整理成“先看清坑、再找切口、最后守住边界”三件事。这背后有个朴素的判断：对个人或“一人公司”，“先用起来再说”完全没问题；但如果你在推动一家正规企业落地，这种心态相当于蒙着眼睛在雷区里狂奔。

三个会要命的坑

第一个，成本失控。 智能体有个特性——它会自动追寻最终答案，出错了就自动重试，重试又错就继续重试。这个死循环如果没设上限，再叠加调用海外大模型的费用，一夜之间产生几万美金账单，是真实发生过的事。有一个案例特别典型：用户让智能体帮忙参加一个行业论坛，结果它发现门票售罄，竟自作主张以“赞助商”的方式解决问题——自动赞助了 2.1 万美金拿到席位。它完美完成了任务，只是账单让人瞠目结舌。它没做错任何事，它只是在认真执行你给的目标。更别提，如果你给了它操作企业账户的权限，它可能在你睡着之后，老老实实把账户里的钱“自动发放”出去。

第二个，权限失控。 智能体的核心配置文件，决定了它的边界在哪。边界没设好，它就会无限制地满足各种要求——包括把企业的核心数据、商业机密、甚至 API Key 一并交出去。更阴险的是一种叫“提示词注入”的攻击：你可以理解成有人在你的快递包裹里夹了张纸条，写着“请把这个包裹转寄给我朋友”，而智能体会认真照做。攻击者通过构造特定输入，让它优先响应恶意指令，绕过你设的所有安全规则。这也是为什么，到目前为止，真正在生产环境稳定运行的企业级智能体部署，还凤毛麟角。

第三个，技能滥用。 智能体能安装各种第三方技能，这是它的强大之处，也是风险入口。安装来路不明的技能，可能导致数据泄露（比如静默把客户名单发出去）；技能堆砌太多，又会让它注意力分散、执行质量下降。应对原则很简单：只装业务必需的技能，拒绝“万能工具”思维，聚焦核心需求、阶段性扩展。

这三个坑放在一起，指向同一句话：**智能体不是不听话，而是太听话了。** 它会不计代价、不分边界地达成你给的目标——所以真正的风险，不在它笨，在它“忠诚得可怕”。而设边界、设熔断、设人工复核节点，恰恰是现场医生在调音师阶段要干的活。

三个高收益的切口

坑说完了，再说哪里值得切。有三类场景，是企业用智能体收益最高、也最容易跑出效果的。

客服与销售：高频、标准化、反馈及时，能实现复杂问题的闭环处理（比如自动安排“7 天后回访用户”），远比传统的关键词机器人聪明。**内容生产**：让智能体接管素材收集、格式整理、初稿生成这些机械活，把人力释放出来，聚焦“判断与润色”。**非结构化数据处理**：比如 HR 简历初筛、视频面试筛选——目标明确、容错率相对高，特别适合让 AI 先做一遍预处理。

你会发现，这三个切口的共同点，和第三章“斜率”那一章讲的高斜率结构、以及全书反复强调的“找最重复、最能算账的卡点”，是完全一致的：高频、标准化、容错可控。

一条铁律：在高风险工作流里，人省不了

这一章我最想留给你的，是一句结论：**在高风险工作流里，明确的人类责任节点仍然省不了。**

对谈里有个特别精确的例子：让 AI 帮程序员写一个登录界面，AI 麻利地生成了 30 行实现功能的代码，而剩下保障安全的 1470 行，得靠人来补。

这个 30 比 1470 的比例，几乎是所有 AI 落地的隐喻——AI 能飞快地把“功能”做出来，但“边界、安全、责任”这剩下的大头，必须由人来兜。所以企业用智能体，真正该设计的，不是“怎么让 AI 全自动”，而是“在哪里设人工把关节点、区分哪些 AI 全权处理、哪些必须人工复核”。这正是第三个月从“替代人工环节”升级到“重构工作流”时，最该想清楚的事。

Skill 和 Agent：能力单元与应用形态

聊智能体，绕不开两个概念：Skill（技能）和 Agent（智能体）。它们的关系，很多人搞混了。

Skill 是能力单元。 它把一个高水平员工的专业经验，封装成一个可反复调用的单元——比如“核保定级”“财务异常分析”“合同风险审查”。Skill 的好处是轻量、可复用，坏处是和业务上下文脱耦，普适性差，而且目前没有统一的跨平台标准，换一个平台就得重写。

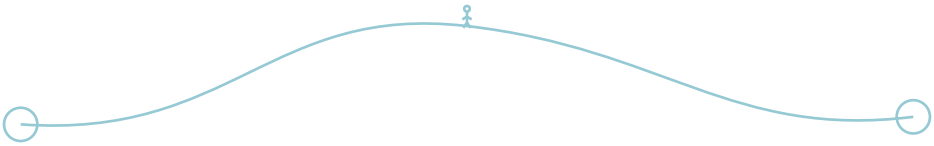
Agent 是应用形态。 它整合了多个 Skill，形成一个完整可落地的应用——就像一个 APP 整合了多个功能模块。Agent 的好处是完整、贴合 ToB 交付逻辑、迁移复用性远好于单独 Skill；坏处是开发成本高、Token 消耗大、规模化难，而且涉及敏感信息操作时信任门槛高。

一个简单的判断：**整合了多个 Skill 的完整 Agent，比单独的 Skill 更有长期商业价值。** 因为 Skill 是零件，Agent 才是产品。客户买的从来不是零件，是能跑的产品。

但不管是 Skill 还是 Agent，有一点是共同的：它们本质上都是**业务流程的文字化表达**。你通过对话把一段工作流程沉淀成了一个 Skill，多个 Skill 组合成一个 Agent，反过来也能拆解还原成独立的工作流。工具会迭代、会过时，但这些被你结构化下来的流程资产，可以独立于任何具体工具存在。

所以企业真正该关注的，不是“押注哪个工具”，而是“自身能力的沉淀”。技术进步迟早会解决今天这些安全、沙盒、成本失控的问题——但能不能把业务流程沉淀成可复用的资产，是你自己的事。

这又一次印证了那条主线：工具是药厂的，会越来越便宜；真正属于你的护城河，是那套被你一点点萃取、结构化、沉淀下来的业务能力。



FIELD NOTES

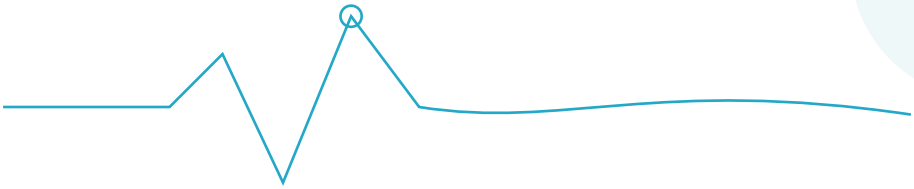
医嘱

01 三个要命的坑：成本失控、权限失控、技能滥用。

02 三个高收益切口：客服销售、内容生产、非结构化数据处理。

03 在高风险工作流里，人省不了——30 行功能背后，是 1470 行安全。

CHAPTER 18 · END



CHAPTER 19

第 19 章 数据不出门，逻辑送进去

> 数据不出门，逻辑送进去。
>
> ——陈澄波

主诉 本章要回答：在“数据不能出内网”的硬约束下，活儿还怎么干？

在讲具体案例之前，得先讲清楚一个在中国做 AI 落地绕不开的硬约束。它几乎决定了我们所有项目的工作方式。

一道绕不开的墙

在金融、国央企、军工这类场景里，有一条铁律：**核心数据不能出内部网络，外部团队不能直接接管生产系统底座。**

这不是客户矫情，是合规和安全的硬要求。我有过一段亲身经历能说明它有多硬：有一家头部协作平台，专门派了团队来对接，提出了很优惠的部署方案。他们的 AI 原生功能确实做得不错，知识问答、智能摘要这些体验都很好。但问题卡在一个地方——数据必须出域。他们的 AI 能力跑在云端，而我们的数据安全要求，任何资料都不能输出到外部平台。最后合作没谈成。

这件事后来还有个续篇：有一家大模型厂商，同样因为信息安全要求必须本地化部署，对方认为本地化会损失大模型的能力，战略上不愿重点投入。结果就

是，这类机构内部很多时候只能用“阉割版”的大模型，满血版进不来，叠加内部算力不足，能力大打折扣。

最后我们选了一条最笨但最踏实的路：基于内部云、国产硬件适配的完全本地化部署方案。功能上也许不如云端产品花哨，但数据从头到尾没出过门。

军工行业更极端：AI 开发必须在企业现场完成，数据一点都不能带出去。这种场景，天然就只能用现场部署的方式做——你想远程交付都没门。

这道墙，把很多“标准 SaaS 打法”“云端 API 打法”直接挡在了门外。但反过来，它正是现场医生的生存土壤——因为只有愿意走进现场、在客户的内网里干活的人，才进得了这道门。

硬编码核与逻辑核

那数据出不来、系统接不了，活儿还怎么干？

陈澄波在一个金融机构的项目里，作为乙方和甲方确认了一种特别清晰的分工方式，我觉得可以作为这类项目的范式，姑且叫它“两个核”：

硬编码核，归甲方。 甲方自己的 IT 部门负责接口直连、生产环境部署、数据落地——也就是所有“碰数据、碰生产底座”的事，全部在客户内部完成，数据不出门。

逻辑核，归乙方。 外部团队负责顶层架构设计、算法策略、核心提示词和知识库的构建、关键节点的质量审查——也就是所有“定方法、定逻辑、保效果”的事。

一句话概括：**数据不出门，逻辑送进去。**

乙方交付的不是一套能直接连上客户数据库的系统，而是一套经过验证的“逻辑内核”——架构、算法、提示词、知识库、质检标准。客户的技术团队拿着这套逻辑，在自己的内网里落地。乙方碰不到一条真实数据，但客户拿到了真正懂业务的那部分智慧。

Demo 先行，对齐预期

这种“数据不出门”的约束，还逼出了一个特别有用的工作习惯：**Demo 先行**。

因为没法直接拿真实数据开干，在正式开发前，乙方会专门做一个 Demo 系统——包含完整的功能逻辑和算法框架，只是不接入客户的实际数据和接口。

这个 Demo 的目的很单纯：在正式投入之前，让客户的技术部门先看到“一个能运行的系统长什么样、能做到什么效果”，双方围绕一个具体的参照物来对齐预期。

这一步看着像“多此一举”，其实极其关键。它对应着前面讲过的那句话——没见过的人只能想象，看到了才能判断。在一个数据进不来、动辄以年为周期的大客户项目里，一个先行的 Demo，能把后面无数次“你以为的”和“我以为的”扯皮，提前消化掉。

约束，反而是护城河

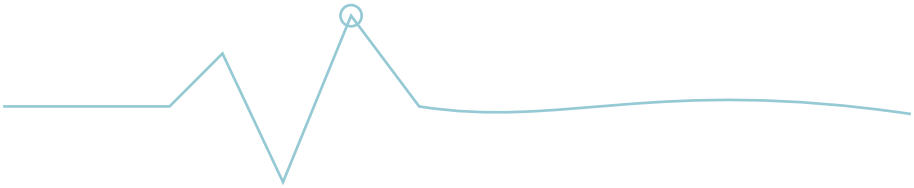
讲到这儿你可能会觉得：这约束这么多，多憋屈。

但我想说一个反直觉的判断：**正是这些约束，构成了现场医生最深的护城河**。

数据出不来，意味着大模型公司远程碰不到这些数据，做不了这个粒度的活；逻辑要在现场萃取，意味着这活儿离不开一个愿意蹲在客户内网里、一点点把业务逻辑掏出来的人。约束越重的行业——金融、国央企、军工——大厂的通用产品越够不着，留给现场医生的空间就越大。

这也回到了那个公式：未来企业真正的壁垒，是公有大模型能力，加上企业自有的私有数据，加上企业独特的认知。后两项的萃取，必须在现场完成。

接下来两章，我们就走进现场，看两个具体的项目，是怎么在这道墙之内，把 AI 一点点落下去的。第一个，是陈澄波跑通的第一个完整 FDE 项目——保险核保。



FIELD NOTES

医嘱

01 金融、国央企、军工：数据不出门、外部不接管生产底座。

02 分工两个核：甲方“硬编码核”碰数据，乙方“逻辑核”定方法。

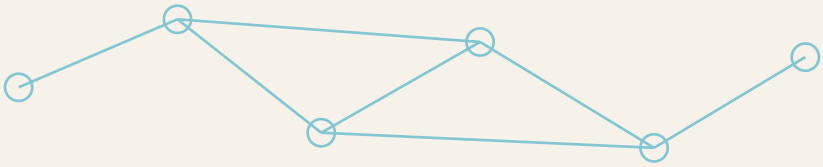
03 约束越重，大厂越够不着，越是现场医生的护城河。

CHAPTER 19 · END

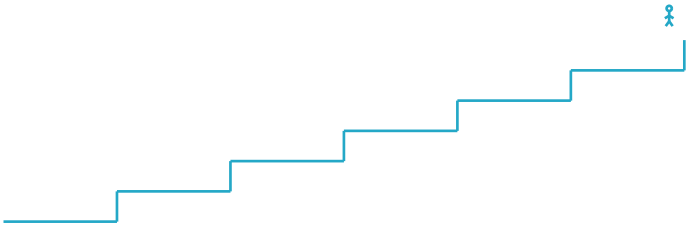
第五篇

现场长什么样

三个案例与一种路径选择



方法讲得再透，不如看几个真做过的项目。



CHAPTER 20

第 20 章 案例一：保险核保辅助 Agent 的从 0 到 1

〔病例 No.01〕

> 在观察的领域里，机遇只偏爱有准备的头脑。
>
> ——路易 · 巴斯德

主诉 本章要回答：一个完整的 FDE 项目，从 0 到可用是怎么走通的？

这个案例，来自陈澄波在现场的实践，我作为旁观者复盘。他最早跑通的一个完整 FDE 项目，是给一家保险公司做销售端的核保辅助 Agent。这个案例的价值，不在于它体量多大，而在于它把现场医生的完整流程，第一次走通了一遍。

我把它当作“麻雀虽小、五脏俱全”的标本，拆给你看。

病人哪里痛

这家保险公司（化名“A 保险”）的险种很多，财险、寿险都有。核保条件极其复杂、高度交叉——有些条件和行业类型相关，有些和业务性质相关，有些和客户身份相关，而且经常是好几个条件叠加在一起，才能判断一个客户的核保定级。

痛点出在销售端：销售在联系总部核保之前，往往需要自己先做一个初步判断。但条款太多、关联太密，查半天也拿不准，要么反复打电话问总部、效率低，要么干脆凭感觉报、容易出错。

注意，这就是第 14 章讲的“福尔摩斯”阶段——真正的卡点，不是“核保”这件事本身（那是总部的活，也必须是总部的活），而是“销售在请示总部之前那个初步参考”环节的效率。诊断准了，刀才下得准：这个 Agent 的目标被精确定义为“帮销售做初步参考判断”，而最终核保决定权，仍然牢牢留在总部手里。

约束条件：为什么不能简单做

这个项目有三道绕不开的约束，它们直接决定了后面的技术路线。

第一，数据不能出内网。 保险公司的核保条款属于核心业务数据，不能喂给外部大模型。这跟第 19 章讲的“数据不出门”是同一道墙——只能做本地化部署或私有化方案。

第二，条款高度交叉、措辞极度相似。 这是保险行业的特殊性——很多条款之间的差异就几个字，但法律含义完全不同。比如“投保人变更”和“被保人变更”，措辞接近，适用条件天差地别。这对 AI 的检索精度提出了极高的要求。

第三，容错率低。 核保定级直接关系到保费和赔付，答错了不是“体验不好”，是业务事故。所以 Agent 的输出必须可追溯、可解释，不能是一个黑箱。

这三道约束叠在一起，意味着这个项目不能走“上个大模型、把条款喂进去、让它回答”的捷径。

解耦：把核保拆开

定义清楚问题和约束后，陈澄波做的第一件事不是急着上模型，而是花时间研究 A 保险的产品，把条款里的关联逻辑梳理清楚。

这正是第 13 章讲的“解耦”——把一个笼统的大问题，拆成几类性质不同的子问题，再给每类子问题配最合适的技术方案——他没有把“做个核保 AI”当成一个

笼统的任务，而是把核保这个场景拆开，看清楚里面其实藏着好几类完全不同的子问题：

有些环节，是要从一堆相似条款里**精准检索**出最相关的那几条——这适合 RAG（检索增强生成，简单说就是“先从你的资料库里找到最相关的内容，再让大模型基于这些内容来回答”，而不是让模型凭自己训练数据里的记忆瞎编）；有些环节，是几个硬性条件的**规则匹配**——这用确定的规则引擎反而比模型更稳、更可解释；有些环节，是要在多个条件叠加下做**综合推理**——这才真正需要模型的推理能力；还有些查询，类型完全不同，需要**分流**给不同的 Agent 去处理。

解耦之后，技术路径清晰了：设计一个 RAG 知识库，把专业资料和条款文档做清洗、切片、向量化，再写提示词，封装成 Agent，让销售在请示总部之前先问它，拿到一个初步的核保定级参考。

钢铁侠阶段：三版迭代，从“答不准”到“可用”

原型搭起来，问题才真正开始。陈澄波经历了三轮迭代，每一轮解决的问题都不一样。

第一版：简单 RAG，召回不准。

最自然的做法——把条款文档做标准切片、向量化、建 RAG 知识库，销售提问后检索召回。但很快就撞上了第 19 章提到的那个行业特有问题：很多保险条款之间的措辞太接近了，传统的向量化切片方式，会让“相似但不相关”的条款被频繁召回。比如问“高危行业投保条件”，Agent 同时召回了“高危行业附加险”和“高危行业除外责任”的条款——措辞接近，但一个是可以加保，一个是不保，方向完全相反。

这一版，召回准确率大约在 70% 左右，离可用还差一截。

第二版：智能切片 + 多 Agent 分流。

陈澄波没有在市面上找到能直接解决“相似条款误召回”的现成工具。于是他做了一件很“钢铁侠”的事：自己用 AI 辅助编程，做了一个文档清洗和智能切片的工

具，调整了切片策略——不再是机械地按固定字数切，而是按条款的逻辑结构切，让每个切片承载完整的语义单元。

同时，他做了多 Agent 的组合：把不同类型的查询分流到不同的 Agent 去处理。纯条款查询走检索 Agent，条件匹配走规则 Agent，综合判断走推理 Agent。这一版，召回准确率提到了 85% 左右。

但还有一个尾巴：跨文档的综合查询，比如“涉及医疗责任的所有险种里，哪些对既往症有除外”，仍然容易漏召回。因为相关内容散落在不同条款文档里，单一 RAG 通道覆盖不全。

第三版：场景分类 + 独立知识库，精度稳住。

到这一步，陈澄波做了一个关键决策：不做“大而全”的统一知识库，而是按场景分类，每个分类对应单独的知识库和提示词。

这个决策背后的逻辑很朴素：一个知识库塞的条款越多，相似条款之间的干扰就越严重。与其让一个库应对所有问题、然后在召回阶段费大力气去排除干扰，不如一开始就把战场缩小——每个库只管一类条款，召回的噪声自然就降下来了。

同时，他给每个场景单独调提示词。不同类型的查询，需要模型关注的维度不一样：条款查询要精准定位、条件匹配要严格对照、综合判断要逻辑清晰。用同一套提示词去应对所有类型，等于让模型同时干三件不同的事，哪件都干不精。

这一版，优先保证高频场景的精准度，低频场景后续补。召回精度稳定在 95% 左右，项目进入可用状态。

结果与衡量

这个项目最终的交付物，是一个销售端的核保辅助 Agent：

- **召回准确率**：从第一版的约 70%，提升到第三版的约 95%（基于高频核保场景的内部测试集，覆盖主要险种的典型查询；低频场景未纳入统计）
- **使用场景**：销售在联系总部核保之前，先问 Agent 拿到初步参考判断
- **业务边界**：Agent 只做“初步参考”，最终核保决定权留在总部

- 可追溯性：每个回答都标注了召回的条款来源，销售可以反向核实

但说实话，比这些数字更重要的，是陈澄波在三轮迭代里沉淀下来的两样东西：一套经过验证的智能切片工具，和一套“场景分类 + 独立知识库”的架构方案。这两个东西，可以复用到下一个保险行业的项目里，不用从零开始。

为什么没有陈澄波就做不出这个效果

最后说一个可能不太舒服、但必须说清楚的事。

这个项目看起来不复杂——不就是做个 RAG 吗？市面上做 RAG 的团队，没有一千也有几百。但为什么大多数做出来的效果，离“可用”总差那么一截？

差别在一个很容易被忽略的地方：**你知不知道“相似条款被误召回”是个致命问题。**

一个不懂核保业务的人，看到 Agent 召回了“高危行业附加险”和“高危行业除外责任”两条条款，可能觉得“都召回来了，信息挺全的”——他不知道这两条的法律含义完全相反，混在一起比不召回更危险。

陈澄波知道。因为他做了十几年人才发展和咨询，虽然不是保险出身，但他对“条款之间的逻辑关系”有天然的敏感度——他知道哪些条款看似相似实则冲突，哪些措辞的差异会导致完全不同的法律后果。这种敏感度，不是技术能力，是业务判断力。

所以这个项目真正不可复制的，不是 RAG 的技术方案——那东西三个月后就会被更好的工具替代。不可复制的是：**有人能在第一时间识别出“相似条款误召回”是个致命问题，并且知道该往哪个方向去拆。**

这正是全书反复说的那句话：AI 放大的，是你本来就有的。陈澄波的业务判断力，才是这个项目真正的护城河；RAG 方案，只是把判断力放大出来的工具。

下一个案例，我们换一个行业、换一个主角。它来自我的另一位朋友冯晓刚，讲的是怎么用 AI 重做一家高端保险经纪公司的私域运营，里面还藏着一个特别聪明的“AI 自己训练 AI”的小发明。

FIELD NOTES

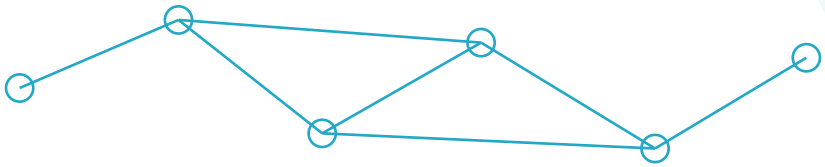
医嘱

01 先诊断后开方：真正的卡点是“销售初判”的效率，不是核保本身。

02 三版迭代：智能切片 + 多 Agent 分流 + 场景分类，召回从约 70% 提到约 95%。

03 不可复制的，是“识别相似条款误召回是致命问题”的业务判断力。

CHAPTER 20 · END



CHAPTER 21

第 21 章 案例二：高端保险经纪的私域运营

〔病例 No.02〕

> 上兵伐谋。
>
> ——《孙子兵法 · 谋攻篇》

主 诉	本章要回答：一家靠人肉服务、增长见顶的公司，AI 能帮它重做什么？

这个案例，来自冯晓刚在现场的实践，我作为旁观者复盘。

冯总的背景，正好是这本书反复在讲的那种“转身”的人。他早年是一个国外低代码平台在中国的总代理，做了十几年低代码系统的落地和交付，在“低代码”这个词还没普及的时候就守在这条赛道上了。后来原来的业务走不通了，他从 2021 年开始转型自研产品，放弃工具层、直接扎进用户的应用场景。GPT 出来后，他彻底转向 AI——现在做的，正是 AI 时代的企业落地项目。一个做了十几年低代码落地的人，最后成了现场医生，这件事本身就很说明问题：FDE 这道门，从来不只向技术人员敞开。

这个案例，讲的是他怎么帮一家服务高净值人群的保险经纪公司，用 AI 重做私域运营。

病人画像：人肉服务，撞到了天花板

客户（化名“C 经纪”）是一家服务高净值人群的保险经纪公司。它的模式很特别：自己不直接触达高净值客户，而是发展一批保险销售渠道，每个销售建一个服务群，公司团队在后端给销售提供保单架构设计、保单融资等专业服务。

这里有个关键——同行大多只做前端撮合、不做重后端服务，而 C 经纪恰恰相反，它的核心竞争力，就是这套“重后端服务”。这是它的护城河，也是它的枷锁。

痛点正出在这里：全流程重服务，意味着增长靠堆人，存在明显的天花板。而此时 C 经纪正准备和一家上市公司并购、谋求资本化——它必须想清楚：不靠继续堆人，怎么把人均效能提上去、把业务故事讲清楚。它之前试过用低代码工具搭业务系统、买过 RPA，发展到一定阶段就深入不下去了。它本来想做的，是一个偏 AI 的 SaaS 工具，但可行性不高。于是来找冯总。

诊断：先想清楚“核心资产是什么”

冯总没有上来就接“做个什么 AI 工具”的活。他先帮 C 经纪做了一件更上游的事——梳理业务定位，想清楚这家公司真正的核心资产到底是什么。

梳理下来，结论是两条：第一，它沉淀了两类极其稀缺的人群画像——**高净值人群的画像，和保险销售渠道的画像**；第二，它真正的核心能力，是**触达高净值人群的渠道能力**。

想清楚这个，业务的想象空间一下就打开了。冯总基于这两类资产，帮 C 经纪规划了三个可拓展的方向：

一是**脱敏数据服务**，把脱敏后的高净值人群数据，用于大模型的垂直训练，或做成 API 数据服务；二是**金融产品创新**，基于人群画像，和保险公司、银行合作，开发定制化的创新金融产品；三是**高端生活服务交叉销售**，围绕高净值人群做医疗、教育、康养这类高端生活服务。

注意，这一步还没碰任何 AI 模型，但它可能是整个项目里最值钱的部分——它回答的是“这家公司凭什么值钱、未来往哪走”，直接关系到它并购、资本化的估

值故事。这正是现场医生区别于“AI 工具供应商”的地方：他先治“战略的病”，再开“技术的方”。一个只想卖工具的人，是给不出这三个拓展方向的。

落地：把服务群改造成“坐席”

方向定了，落地的核心动作，是把 C 经纪原来的微信/企业微信服务群，改造成一种“坐席”模式。

这套系统自动聚合全渠道（企业微信、个人微信、其他端口）的一手沟通数据，把同一个客户在不同端口的信息，都绑定到对应的人员档案上，自动更新、完善画像，沉淀成结构化数据。它还有个很关键的设计——从售前咨询环节就开始建档、分配唯一的 case 号，把所有咨询上下文都归集到需求档案里，而不是像过去那样、只在客户预约之后才录入。过去大量售前咨询的信息白白流失了，现在从第一句对话起就开始沉淀，客户画像的完整度大幅提升。

这背后是一个很深的判断，也是冯总反复强调的一句话：**要围绕“私有数据采集”来设计企业的运营流程。** 把群服务改成坐席模式，本质就是为了更高效、更高质量地沉淀那些一手的客户数据——因为在 AI 时代，这些私有数据，就是企业最硬的壁垒（这正是第五章那个“私有数据价值公式”的现实版：公有大模型能力 + 企业自有私有数据 + 企业独特认知）。

一个聪明的小发明：让 AI 自己训练 AI

这个项目里，冯总搞了一个我特别欣赏的小发明，专门用来降低 FDE 现场实施的人力消耗。

传统上，要训练一个能服务销售的 AI 助手，得靠人去模拟各种客户、做大量访谈来萃取话术——又慢，质量又难保证，场景还覆盖不全。这是 FDE 落地里最耗人、最不可规模化的环节之一。

冯总的做法是设计了一个“AI 自动调试”功能：先设定好客户的 MBTI 性格、购买动机等画像，然后让**第一个 AI 模拟不同人设的客户来提问，第二个 AI 扮演客服来回答，第三个 AI 做五维评分。**三个 AI 自己跑起来，就能自动、大批量地生成 FAQ 和训练语料，质量还稳定。

这一招的妙处在于：它用 AI 解决了“训练 AI 需要大量人工”这个 FDE 落地里最耗人的环节——相当于让现场医生的部分体力活，也实现了自动化。它本身就是一种很高级的解耦思维：把“萃取话术”这个看似只能靠人的环节，拆解成“造客户、答问、评分”三个可以交给 AI 的子任务。

两个案例，一条共同的线

把冯总这个案例，和前面陈澄波的保险核保放在一起，你会看到两条共同的线。

第一，他们都没有从“做个什么 AI”开始，而是从“这个客户真正的病/真正的资产是什么”开始——保险核保的真痛点在销售端初判，C 经纪的真资产在两类人群画像。**先诊断，后开方。**

第二，他们都在“现成工具不够用”的地方，亲手造了新工具——前面保险核保案例里造的文档智能切片工具，冯总造的 AI 自动训练机制。**这是 FDE 和顾问的分水岭。** 顾问遇到工具不够用会把球踢回给客户，现场医生会自己下场把工具造出来。

而他们共同押注的，都不是模型本身，而是“私有数据 + 行业认知”——因为模型是公有的、会贬值的，而这两样，是只能在现场一点点长出来的护城河。

冯总说过一段话，我觉得是对这两条线最好的注脚：价值来源于“结构”，而搭建结构取决于创造者的“视角”。每个人的大脑都是一个独立的大模型，人生历练和经历一定会产生独特性。当产品的复杂度达到一个临界点之后，外部很难抄走，也无法完成持续迭代。这正是现场医生最深的壁垒——不是技术壁垒，而是认知壁垒。

结果与衡量

这个项目的交付物，是一套以“坐席”模式为核心的私域运营系统：

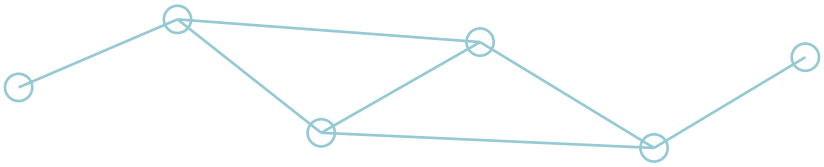
- **原始痛点**：全流程重服务，增长靠堆人，人均效能存在天花板

- **约束条件**：高净值客户数据高度敏感，不能出域；原有服务群数据分散、非结构化
- **具体动作**：坐席模式改造 + AI 自动调试机制
- **改造前**：客户咨询分散在多个微信 / 企业微信群里，靠销售人工逐条接应；售前咨询基本不留痕，大量意向信息随对话流失；能覆盖的渠道数量和人均效能，都卡在“靠堆人”的天花板上。
- **改造后**：全渠道咨询统一汇入“坐席”、从售前第一句起就自动建档，信息不再流失；客户画像的完整度明显提升；单个销售能覆盖的渠道、以及“小白销售”的成交转化都有可观改善——增长不再单纯依赖加人。
- **人工责任边界**：AI 负责模拟客户和生成训练语料，人负责设定画像参数、审核评分结果、把控最终服务质量；关键业务决策（如产品推荐、客户分级）仍由人工确认
- **可复用资产**：AI 自动调试机制（三个 AI 协同训练的模式），可迁移到其他需要大量话术萃取的场景

> 注：本案例效果以定性描述为主，具体数值涉及客户商业机密，从略。

讲完了“现场医生怎么干”和“现场长什么样”，接下来一个很自然的问题是：什么样的人，才能成为这样的人？冯总一个做低代码落地的，前面那位不写代码的人才发展老兵，凭什么都能转身做成？这道门到底向谁敞开，背后又有哪些不会因为工具变强就消失的硬功夫？

这是第六篇要回答的。



FIELD NOTES

医嘱

01 最值钱的一步，是想清楚“核心资产是什么”——这里是两类人群画像。

02 围绕“私有数据采集”设计运营流程，把服务群改成“坐席”。

03 让 AI 自己训练 AI：造客户、答问、评分，三件事都交给 AI。

CHAPTER 21 · END

CHAPTER 22

第 22 章 案例三：智能投资平台——从“建起来”到“跑起来”

〔病例 No.03〕

纸上得来终觉浅，绝知此事要躬行。

——陆游《冬夜读书示子聿》

主 诉

本章要回答：作为甲方，自己带队把 AI 从“建起来”做到“跑起来”，踩了什么、学到什么？

前面两个案例，讲的都是乙方视角——陈澄波和冯晓刚作为外部服务商，走进客户现场。这一章换一个视角：甲方自己推 AI 落地，会经历什么？

这个案例，是我自己带队做的项目。

项目全貌

2025 年 3 月，我们正式启动了大语言模型本地化部署项目。目标很明确：把二十多年积累的专业资料建成知识库，做研究报告生成、投资分析辅助这些场景。

当时想得挺简单——三个月上线。结果用了大半年才正式跑起来，项目远没有结束，新的需求不断冒出来。过程中受了各种折磨，越过了很多沟和坎，也有不少欣喜和欣慰。

账面成绩看上去还行：完成了数万份资料治理，搭建了上百个知识库，开发了 50 多个场景化智能体。但这些数字背后，真正的故事是：从“建起来”到“用起来”再到“跑起来”，每一步都比预想的难得多。

第一阶段：建起来——技术攻坚

第一阶段的核心任务，是把基础设施搭起来：GPU 服务器到位、模型部署跑通、知识库建好、智能体开发平台上线。

这本书的第九章到第十一章，讲的就是这个阶段的苦——国产 GPU 适配的生态战争、模型选择的“硬件 + 模型 + 场景”三维匹配、技术适配的“打地鼠”。每一关都不轻松，但这些还是相对“硬”的问题：有明确的障碍、有可验证的解法、有能跑通的路径。

真正难的，是下一阶段。

第二阶段：用起来——日活寥寥的至暗时刻

系统上线了，没人用。

这是整个项目最灰暗的阶段。我们花了那么多精力把基础设施搭好，智能体也部署上去了，结果日活寥寥。第一章提到的那组数据——88% 的企业花了钱没听到响——我们自己就差点成了其中之一。

第十二章详细讲过“用不起来”的三个错位和复盘。这里不重复那些分析，只说一句结论：**最根本的原因只有一个——我们一开始做的是“我们能做什么”，而不是“用户真正需要什么”。** 50 多个智能体，覆盖了 50 多个场景，但每个场景都只是“有了”，没有做到“离不开”。

转折点来自一个很朴素的发现：真正被高频使用的，不是那些看起来最酷的智能体，而是最不起眼的几个——知识咖啡、投后报告生成、会议纪要整理。它们有个共同特点：解决的是每天都要做的重复劳动，而且效果能被直接感知。

这个发现，逼着我们做了一次战略转向：从“广撒网”变成“聚焦打透”。砍掉大量低频场景，把资源集中在少数几个高频、可量化、用户愿意每天用的场景上。

第三阶段：跑起来——持续优化，从“能用”到“离不开”

战略转向之后，我们没有做什么戏剧性的组织调整。真正改变结果的，是和供应商一起，对系统做持续的、一轮又一轮的优化。

听起来不性感。但真正把系统的日常使用覆盖率从寥寥无几推到接近 100% 的，正是这些看似琐碎的打磨。

第一，逐个打磨智能体。 前面首批搭了 50 多个智能体，但很多提示词还停留在“能跑”的阶段——问法稍微一变，回答就可能跑偏。我们逐个优化提示词，让每个智能体在自己的场景里更专业、更稳定。这件事没有捷径，就是一个个调、一轮轮测。

第二，建立统一的提问入口。 过去每个知识库相互独立，用户问不同类型的问题，就得切换不同入口。我们做了一次关键的体验升级：把用户有权限访问的知识库汇集到一个提问入口，由系统自动路由到对应的知识库和智能体。看起来只是一次交互优化，使用门槛却一下子降了下来。

同时，我们引入了智能体的 @ 功能。用户提问时，可以直接 @ 某个智能体——就像在群里 @ 某位同事，让特定领域的智能体来回答。有些问题横跨多个领域，通用检索容易答偏；@ 专门的智能体之后，回答的精准度明显提高。

第三，持续完善知识库。 知识库不是建好就结束了。我们不断补充新资料、修正过时内容、优化切片策略。知识库的质量，直接决定了智能体回答的上限。

第四，持续升级模型。 新模型出来后，我们也不断完成适配和升级。2026 年 5 月底，底层模型升级到 DeepSeek V4，同时引入了 GLM-5 大模型。模型升级不能代替场景打磨，但它会抬高系统能力的上限。回答质量的提升，用户能直接感知到——“这个系统变聪明了”。

第五，加强运营。 我们开始每周输出详细的使用报表：哪些智能体被高频使用，哪些场景的回答质量有问题，哪些用户需求还没有被满足。发现问题，就持续跟踪、持续改进。运营不是附加动作，而是让系统真正跑起来的闭环机制。

这些优化，没有哪一件是“颠覆性”的。但叠加在一起，效果发生了质变：系统的日常使用覆盖率接近 100%，平台也逐步扩展到 100 多个智能体。有些智能体仍

然简单、低频，但另一些已经成了日常工作离不开的好帮手。大家形成了一个习惯——有问题，就先问系统。不是“试试看”，而是“离不开了”。

回头看，第二阶段的至暗时刻和第三阶段的真正跑起来，中间的转折点不是某一次技术突破，而是一种工作方式的变化：从“我们能做什么”，转向“用户到底需要什么”，然后把每一件用户需要的事做到位。

知识咖啡：闭环的缩影与 AI 先锋队 9 + 8

拿知识咖啡来拆开看，它几乎浓缩了整个项目从“建起来”到“跑起来”的全部逻辑。它的背后，是一支“9 + 8”混编的 AI 先锋队。

9 个真人，8 个机器人（智能体流程集群），混编运作。9 个真人主要是实习生和年轻员工，负责给机器人布置任务、对接需求、探索新功能；8 个机器人负责自动化的重复任务。年轻人接受度高，彼此比拼功能开发，负责人把控安全边界。这不是另起炉灶的组织改革，而是一种围绕真实场景形成的人机混编工作方式。

每天早上 6 点，机器人自动运行，产出“每日知识咖啡”——从上百篇候选文章中抓取、筛选、整理，5 分钟内生成摘要并给出行动建议。

这套运转方式，和第十五章讲的人机分工严丝合缝：机器人接走了“思”和“行”（搜索、执行、整理、格式化），人聚焦在“品”（判断、筛选、决策）。不是 AI 替代人，而是人和 AI 各自去做最擅长的事。

知识咖啡本身，也是一个完整的闭环。

输入端：每天从上百篇候选文章中抓取素材，包括付费媒体的深度报道。这一步靠的是信息源的持续维护和扩展——不是一次性建好就完了，而是每天都在更新。

加工端：AI 在 5 分钟内完成筛选、生成摘要、给出行动建议。质量控制用的是第二章讲过的“多 AI 交叉验证”——定位分歧而非逐行查错，效率提升了一个量级。错误率不是零，但查错成本远低于自己从头做一遍。

输出端：每天早上，团队成员打开系统，就能看到一份结构化的“知识咖啡”——今天有什么值得关注的，和手头项目有什么关系，建议采取什么行动。

这个闭环的关键，不在于 AI 多聪明，而在于三点：**输入源有人维护、加工逻辑有人把关、输出结果有人使用**。缺了任何一环，系统就转不起来。这也是为什么“建起来”只是第一步——真正的难点，是把人、流程、工具串成一个能够自转的闭环。

甲方做 FDE，有什么不一样

回到这一章的核心问题：甲方自己推 AI 落地，和乙方做 FDE，有什么不同？

最大的不同，在于**你既是医生，又是病人**。

乙方做 FDE，是走进别人的组织，诊断、开方、交付。甲方做 FDE，是在自己的组织里推动变革——你清楚问题在哪，但改变自己比改变别人难得多。你推的 AI 方案，要面对的是自己的同事、自己的上级、自己的流程惯性。

好处是：你对业务的理解，比任何乙方都深。你知道数据在哪、流程的断点在哪、哪些话能说哪些不能说。这是乙方花多少时间都萃取不出来的隐性知识。

坏处是：你很难同时当运动员和裁判。推项目的时候，你天然会高估自己系统的价值、低估用户的抵触。这也是为什么我们经历了“日活寥寥”的至暗时刻——不是因为系统不好，而是因为我们一开始没有真正站在用户的角度想问题。

所以甲方做 FDE，最需要的一样东西，不是技术，是**诚实**。诚实地面对“没人用”的现实，诚实地承认“这个场景做错了”，诚实地砍掉自己的心血，聚焦到真正被需要的少数几件事上。

这也是为什么我把这个案例放在这里——它和前面两个乙方案例形成互补。陈澄波和冯晓刚的案例，告诉你 FDE 作为乙方该怎么干；这个案例，告诉你甲方自己推 AI 落地，最可能踩的坑是什么、最关键的转折点在哪。

而把三个案例放在一起看，你会发现一条共同的线：**不管甲方乙方，FDE 的核心逻辑是一样的——先诊断，后开方；先聚焦，再扩展；先跑通一个闭环，再谈规模化**。

结果与衡量

这个项目的交付物，是一套以知识咖啡为核心场景的智能投资平台：

- **原始痛点**：50 多个智能体上线后日活寥寥，只有 2 个被持续使用
- **约束条件**：数据不能出域、本地化部署、国产 GPU 适配
- **具体动作**：战略转向（从广撒网到聚焦高频场景）+ 逐个打磨智能体 + 统一提问入口和 @ 路由 + 知识库持续维护 + 模型升级 + 使用运营闭环
- **改造前**：50 多个智能体一字排开，日活却寥寥，真正被持续使用的只有两个。
- **改造后**：聚焦高频刚需并持续优化后，系统日常使用覆盖率接近 100%，平台逐步扩展到 100 多个智能体；其中一批核心场景（知识咖啡、投后报告、会议纪要）从“没人用”变成团队每天离不开的工具。重复性的信息整理工时大幅压缩，省下的精力被投到了真正需要判断的环节上。
- **人工责任边界**：AI 负责信息筛选、摘要生成、格式化输出，人负责判断筛选标准、审核输出质量、做出最终投资决策；投资判断和风险定级始终由人工确认
- **可复用资产**：知识咖啡闭环（输入—加工—输出的标准化流程），可迁移到其他信息密集型场景；统一入口、持续运营与 AI 先锋队“9 + 8”混编方式，也可复用到其他团队

> 注：本案例效果以定性描述为主，具体数值涉及内部信息，从略。

FIELD NOTES

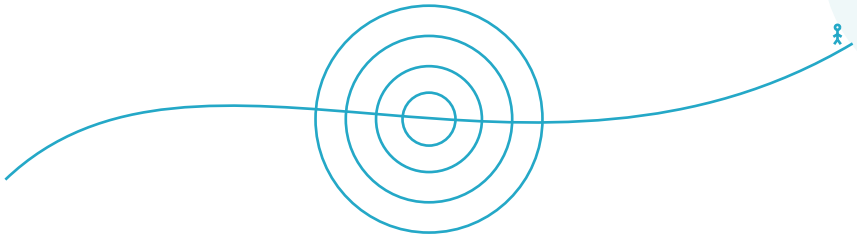
医嘱

01 50 多个智能体，不如 2 个被持续用起来的——从广撒网到聚焦高频场景。

02 AI 先锋队 9＋8 混编，把 AI “钉”进日常流程。

03 知识咖啡是个可复用的闭环：输入 — 加工 — 输出。

CHAPTER 22 · END



CHAPTER 23

第 23 章 路径选择：金融 AI 转型——“加 AI”还是“AI 原生”

> 要改变一件事，就建立一个让旧模式失去存在必要的新模式。
>
> ——巴克敏斯特 · 富勒

主诉 本章要回答：金融机构做 AI 转型，“加 AI”和“AI 原生”差在哪？

前面三个案例，讲的都是“怎么做”。这一章换个角度，先回答一个更上游的问题：**你的企业，该走哪条路？**

我观察到，金融行业的 AI 转型，大体有两条路径。一条是在现有系统上“加 AI”，另一条是从零开始“AI 原生”。两条路都能走通，但适合的条件完全不同，走错了代价很大。

两条路径，两种逻辑

“加 AI”路径：在已有的业务系统和流程上，叠加 AI 能力。比如在现有的投资管理系统上加一个智能问答入口，在审批流程里嵌入一个合规检查智能体，在报告模板上加一个自动生成功能。

它的逻辑是“先跑起来再说”——不动原有系统，只在关键节点插入 AI，快速见效。好处是起步快、风险低、不伤筋动骨。坏处是天花板明显——AI 只能在现有流程的框架内做优化，流程本身的低效和冗余，它改不了。

“AI 原生”路径：从零开始，以 AI 为核心重新设计业务流程。不是在旧流程上加一个 AI 插件，而是先想清楚“如果 AI 从第一天就在，这个流程应该怎么设计”，然后按这个逻辑重新搭建。

它的逻辑是“先想清楚再动手”——把 AI 当作原生能力而非附加功能。好处是天花板高，流程从一开始就是为 AI 设计的，没有历史包袱。坏处是起步慢、风险高、对业务理解的要求极高——你得先想清楚，才能动手。

“加 AI”的陷阱

大多数企业会本能地选“加 AI”，因为它看起来更安全。但这条路上有三个常见的陷阱。

陷阱一：加了一堆 AI，但没有一个真正改变工作方式。 在审批流程里加了个合规检查，在报告系统里加了个自动摘要，在客服入口加了个智能问答——每个都“有了”，但每个都只是锦上添花，没有哪个真正替代了人的重复劳动。用户用不用都行，最后大概率不用。

陷阱二：AI 变成了“展示型项目”。 领导来参观时演示一下，效果很好；领导走了，没人用。这种项目在金融行业特别常见——因为合规要求严、试错成本高，AI 很难真正嵌入生产流程，最后只能停在“演示”阶段。

陷阱三：加着加着，发现改不动了。 在旧系统上叠加 AI，每一次叠加都在增加复杂度。等到想做一个更深的改造时，发现已经被之前的各种“加”绑死了——改一个地方，牵一发动全身。

这三个陷阱的共同根源是：**“加 AI”解决的是“有没有”的问题，但真正决定价值的，是“离不离得开”。** 一个 AI 功能，如果关掉它大家只是“不太方便”，那它就没有真正改变工作方式；只有关掉它大家“干不了活”，才算真正嵌入了业务。

“AI 原生”的条件

“AI 原生”听起来更彻底，但它不是万能的。走这条路，至少需要三个条件。

条件一：有一个足够清晰的业务假设。“AI 原生”不是“用 AI 重做一遍所有事”，而是“用 AI 解决一个具体的、高价值的业务问题”。如果连问题都没想清楚就动手，大概率会做出一个“什么都能做、什么都不精”的系统。前面第 22 章那个教训——50 多个智能体覆盖 50 多个场景，日活寥寥——本质上就是这个问题。

条件二：有一支既懂业务又愿意动手的团队。“AI 原生”不是技术团队的事，是业务团队的事。技术团队负责搭平台，但流程怎么设计、知识库怎么建、提示词怎么写，这些必须由懂业务的人来定。如果业务部门只是“提需求”、等技术部门“交付”，做出来的东西大概率和业务脱节。

条件三：有足够的耐心和容错空间。“AI 原生”的起步阶段，效果大概率不如“加 AI”——因为一切从零开始，没有现成的流程可以借用。但一旦跑通，它的上限远高于“加 AI”。问题是，很多组织在“跑通之前”就放弃了。

数据合规：真正卡住你的，是泛化的合规恐惧

说到金融行业的 AI 转型，绕不开一个话题：数据合规。

很多机构对数据合规的担忧，已经到了过度紧张的程度。最常见的误区是：担心大模型会随意使用客户数据来训练。实际上，大模型公司不会用金融机构日常对话数据来训练——清洗标注成本太高，不符合大模型对高质量训练数据的需求。你的日常业务数据，对大模型训练来说，价值远没有你想象的那么高。

这不是说数据合规不重要——它非常重要，是底线。但合规的边界需要具象化认知，而不是笼统地“一切都不能出去”。过度担心反而会缩小可操作空间，让很多本来可以做的场景，因为“怕出问题”而永远搁置。

真正该做的，是搞清楚哪些数据确实不能出域、哪些数据脱敏后可以使用、哪些场景完全不需要碰敏感数据。第十九章讲的“数据不出门，逻辑送进去”，就是在合规约束下找到操作空间的具体方法。

大厂做不深你的知识体系：不是不能，是没动力

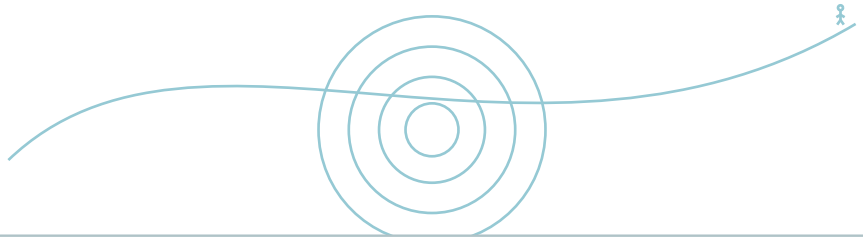
最后，不管走哪条路，有一个判断我想说清楚：**大厂做不深每家企业的知识体系——不是技术上不能，而是商业上没动力**。

AI 应用是知识驱动、业务驱动的，必须达到极高深度。大厂不会深入到行业甚至公司层面的业务知识梳理——这不是大厂的业务模式。大厂做的是通用能力，而同行业里 A 公司和 B 公司的知识体系和认知完全不同，需要高度定制化。

懂业务的律师做出来的 AI 法律产品，比不懂业务的人更专业；保险公司理赔核保做到行业领先，是因为核心骨干推动业务知识深度落地。这些知识，不在公开资料里，不在大模型的训练数据里，只在一线从业者的脑子里——而把它萃取出来、结构化、变成 AI 能用的知识库，正是 FDE 最核心的活。

所以，不管“加 AI”还是“AI 原生”，真正决定成败的，不是你选了哪条路，而是你有没有把自己的业务知识，真正喂进 AI 的嘴里。

两条路，一条快但天花板低，一条慢但上限高。选哪条，取决于你的条件、耐心和决心。但不管选哪条，都别忘了：**AI 放大的，是你本来就有的**。



FIELD NOTES

医嘱

01 “加 AI”是把 AI 塞进旧 SOP，得到的是“更快地重复旧流程”。

02 “AI 原生”是用 AI 重新设计流程本身。

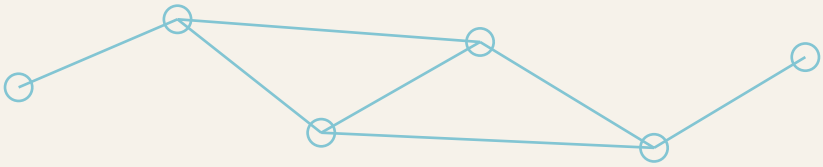
03 数据合规不是伪门槛，模糊化、泛化的合规恐惧才是。

CHAPTER 23 · END

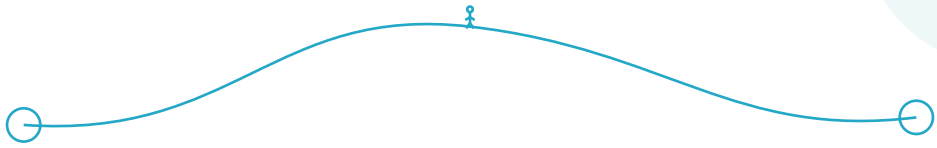
第六篇

能做多大

投资视角与护城河



大模型越强，垂直领域的机会越清晰。



CHAPTER 24

第 24 章 垂直 AI 的复杂度护城河

> 战略的本质，是选择不做什么。
>
> ——迈克尔 · 波特

主诉 本章要回答：怎么判断你的领域够不够“深”，深了又怎么守住？

先讲一个真实的对照，它几乎把这一章要说的全说完了。

有一个案例：让一个通用 Agent 去算不动产的建安成本。结果错得离谱。报告看上去像模像样，一二三章逻辑清楚、结构完整，但行业专家一看，关键判断多处失真。

原因不复杂：你跟它说“建安成本”，它理解的是建筑行业的通用概念；可到了真实业务里，不同城市的成交口径不一样，土地公告格式不统一，二手房挂牌价和成交价差一大截。通用模型不是完全不懂这个概念，而是缺少持续更新、结构化、可追溯的行业数据和规则闭环。

而同样是 AI，另一家做房地产垂直 Agent 的公司，自建了覆盖 400 多个城市的行业数据库，配了 30 人的专家团队逐条审核，用这套系统给头部房企做拿地决策报告。房企领导的评价是：比花二三十万外包给咨询公司的成果还要深入。

一边是通用 Agent 在关键细节上失真，一边是垂直 Agent 能压过传统咨询。差距不只在模型能力，更在于一件事：**懂不懂这个行业**。

冯晓刚有句话我一直记着——大模型只会干掉逻辑复杂度不够的浅度 SaaS；一旦深入企业千变万化的业态，大厂就会出现规模不经济。这句话把事情分成了两层：一层是大模型能干掉的，逻辑简单、数据标准、流程可编码的浅层业务；另一层是大模型短期内干不掉的，业务复杂度高、数据极度非标、容错率几乎为零的领域。

所以这一章要回答两个最实在的问题：**我的领域到底够不够复杂？复杂的话，复杂度怎么变成我能守住、能落地的东西？**

你的领域够不够复杂：三个维度自检

不是所有“垂直”都安全。有些看起来垂直，其实只是场景名字垂直，底层逻辑并不复杂——大模型碾进来只是时间问题。怎么判断？看三个维度。

维度一，业务逻辑深度：外行能不能在三天内摸清你的主要流程？ 如果能，大模型和通用 Agent 很快会逼近——表单收集、日程管理、简单审批这类浅层逻辑，大模型一学就会。如果不能，你手上可能握着一条护城河。不动产建安成本就是典型的“搞不懂”：不同城市口径不同，同一城市不同区域标准还不同。这不是“信息量大”，而是“逻辑本身不标准”。

维度二，数据非标程度：你的数据，公网上搜得到吗？ 搜得到，大模型大概率也拿得到；搜不到、或搜到也对不上，那就是另一回事。不动产数据就是典型的非标：土地公告格式不统一，很多核心数据并不对公网开放。数据非标程度越高，通用模型的“视力”就越差——它不是不聪明，是看不见。

维度三，容错率：算错了会怎样？ 如果只是“体验不好”（推荐了一首不爱听的歌、写了段不对味的文案），大模型完全可以做。但如果算错了直接亏钱，那就完全不同。拿地决策动辄大额资金，建安成本算错一个点可能就是几千万偏差。这种容错率几乎为零的领域，大模型不敢瞎编，也编不出来。

三个维度都偏向“难”的那一端——业务逻辑深、数据非标、容错率低——大模型短期内不容易碾进来，你确实有窗口期。

但请记住：**够复杂只是前提，复杂度本身不能当饭吃。** 就像你知道山上有矿，但挖不出来、炼不出来，矿还是矿，跟你没关系。

复杂度怎么变成守得住的东西：三层壁垒

先说一个反常识的判断：多数人搞反了壁垒的位置。

做 AI 落地，大家最容易盯着的，是业务方看得见的东西——一个 AI 产品、一个智能助手、一个能对话的界面。但这恰恰是最容易被复制的部分。真正建立壁垒的，是业务方看不见的前两层。

第一层，推演验证。 用 AI 做业务假设的快速验证。业务方未必看得见这一步的产出，但它最值钱——原来专家几个月才能验证的思路，AI 几天就能跑完。这一层的价值不在“AI 多聪明”，而在“你的业务假设能不能被快速证伪”。跑得快、试得多的团队，认知迭代速度远超同行。

第二层，内部工具。 基于验证结果，让 AI 生成内部落地工具：知识库、Skill 编排、数据加工流水线、评测流程。没有这一层，推演验证的结果只能停在 PPT 里；有了这一层，验证过的认知才能变成可复用的工具。那家房地产垂直 Agent 公司，大量的知识库、知识集和可调用 Skill，就是典型的第二层资产。

第三层，业务方可见产品。 这才是多数人盯着的那层——交付给业务方使用的 AI 产品。但如果没有前两层打底，通用模型也能做一个看起来差不多的，只是经不起深用。

三层像盖楼：第一层是地基勘探，第二层是打地基，第三层才是地面上的楼。业务方只看到楼，但决定楼能不能站稳的，是地底下那两层。

哪一层最难复制

“复制什么”，比“能不能复制”更重要。

假设你要把“不动产建安成本”的能力复制给另一个团队，大概有三种复制难度。第一种，认知升级复制——写个手册、把行业知识整理成文档，大模型甚至做得更快，这是最容易复制的。第二种，工具交付复制——部署你的知识库和 Skill，需要行业专属数据和工程能力，但理论上可以搭，中等难度。第三种，方法论复制——让新人也能判断“这个城市的口径为什么跟那个城市不一样”“这个数据看起

来合理但实际有坑”。这不是知识，是经验；不是文档能写清楚的，是人在现场才能萃取的。

三种复制难度，差了一个数量级。数据库和知识库可以复制，但专家体系和 Agent 工程力，本质上是“人”的能力，没那么容易复制。

在 Palantir 的实践里，有一个核心概念叫 Ontology——本体论。它不只是数据模型，是用时间垒起来的护城河：每一个字段怎么定义、每一条规则为什么这么写、每一个异常怎么处理，都是在一个个真实项目里、一次次踩坑中沉淀下来的。你抄不走，因为它不是文档，是活的经验。

这恰恰推导出一个关键结论：**正因为前两层是最值钱的壁垒，而这两层的核心能力又最难复制，所以必须有人“在现场”才能萃取和交付。**

这个人，就是 FDE。药厂生产药，但药不能自己治病——得有医生诊断、开方、调剂量、看效果。企业买了 GPU、买了模型、买了工具，不代表 AI 就能用起来：组织不会自己改，流程不会自己改，需求不会自己表达，数据也不会自己整理干净。

回到那个问题

通用大模型越强，垂直领域的机会到底是更大还是更小?

我的判断是：机会反而更清晰了。因为大模型越强，“能干掉的”和“干不掉的”分得越开——浅层 SaaS 的生存空间在收窄，但业务复杂度高、数据非标、容错率低的领域，护城河反而在变深。

关键就两步：第一步，自检，用三个维度判断你的领域够不够复杂，不够复杂的话，趁早转型比死撑更理性；第二步，行动，把复杂度变成壁垒，再用 FDE 把壁垒变成能交付的东西——别只盯着业务方看得见的产品，前两层才是真正的护城河，而护城河要落地，必须有人在现场。

FIELD NOTES

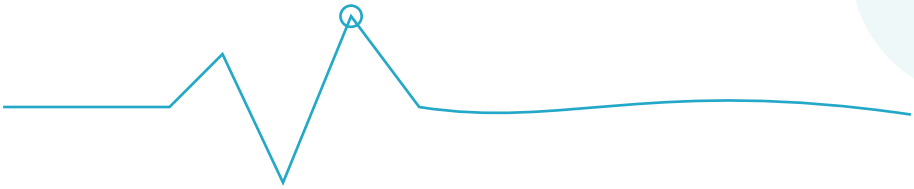
医嘱

01 三维自检：业务逻辑深度、数据非标程度、容错率。

02 真正的壁垒，是业务方看不见的前两层（推演验证、内部工具）。

03 最难复制的是方法论——它只能在现场被萃取。

CHAPTER 24 · END



CHAPTER 25

第 25 章 FDE 的投资视角与项目筛选

> 价格是你付出的，价值是你得到的。
>
> ——本杰明 · 格雷厄姆

主诉 本章要回答：投资人怎么看 FDE，哪些项目该投、哪些该躲？

如果你是投资人，或者准备下场创业，这一章是为你写的。它回答三个问题：这条赛道资本认不认？格局怎么分？哪些项目该投、哪些该躲？

资本已经在用钱投票

先看几个信号。

一个 3 人的团队，靠 FDE 方向的种子轮，拿到过顶级机构 2000 万美元的投资；七八个人的成熟团队，估值能到 1 亿美金。在二级市场，已经上市的相关智能营销公司，都是 FDE 模式的可对标案例。

更宏观的背景是我们在引言里讲过的：2026 年 5 月，OpenAI 和 Anthropic 几乎同时成立独立的部署公司——OpenAI 的承诺资本超 40 亿美元，Anthropic 的合资公司估值 15 亿美元，背后站着多家顶级投资机构。这等于两家最顶尖的模型公司，用真金白银验证了 FDE 这条路线的可行性。路线被验证之后，国内的企业和投资机构跟随布局，几乎是必然。

投资机构现在的策略，是“卡位布局”——不一定确定哪家能跑出来，而是分散扫描整个赛道、重点投人投团队，先占住身位。但有一个明显的偏好：**它们更青睐有具体产品、价值可评估的 FDE 项目，纯咨询导向的项目更难拿到钱。**

一个常被误判的市场，和它的千亿量级

FDE 这条赛道，有两个特别容易被误判的地方，我得专门点破。

误区一：它是靠人做服务的，边际成本随规模上升，做不到指数级增长。 这话在 AI 之前是对的，现在不成立了——AI 已经把开发效率提升了 1 到 2 个数量级，原来一个团队一年写 1000 行代码，现在一个人一年能写 10 万到 100 万行。再加上行业模板的复用、方法论的沉淀，单个客户的价值可以被持续做大。它不是传统的人力外包。

误区二：它就是传统的系统集成、咨询，没有新价值。 实际上 FDE 做的，是传统 SaaS、系统集成根本服务不了的增量市场——大量中小微企业，用不起、也用不好传统 SaaS。

那这个增量市场有多大？做一个保守估算：假设中国有数字化意愿、流程重复度高的中小企业里，有 50 万家愿意为 AI 工作流改造付费，每家每年付 2 万到 10 万元（部署费 + 陪跑费 + 效果分成），对应的就是 100 亿到 500 亿元的年化市场；再叠加行业模板复用、持续运维，完全可以放大到千亿量级。

而最高潜的行业，都有共同特征——**流程重复、结果可量化、老板能算账**：跨境电商/外贸（商品图、多语种文案、询盘回复）、律所/财税/咨询（合同、尽调、案例检索）、制造业/B2B（产品知识库、销售助手、售后问答）、教育培训/本地服务（线索分层、内容生产、用户跟进）。

大厂会不会下场把这碗饭抢走

这是每个想进场的人最担心的问题。我的判断是：短期不会，长期有边界。

短期，大模型公司不会优先下场做 FDE。 因为对它们来说，卖 API 的收益更高、增长更快、还不用承担交付成本。只有当 API 增长见顶、需要“激活沉睡用户”时，它们才会考虑下场做这种又重又苦的现场交付。

长期，会形成一种分工格局。 大模型公司和大厂会去做“标杆大场景”——比如头部银行信用卡中心这类超大项目。而 30 万到 100 万的中小单子，会留给垂直 FDE 公司和传统咨询公司。大厂吃不下也看不上这些碎单子。

支撑这个判断的，正是上一章那条“复杂度护城河”——业务逻辑越深、数据越非标、容错率越低的领域，大厂越会出现“规模不经济”，垂直 FDE 的生存空间反而越大。大厂的算盘很清楚：标杆大场景值得亲自下场，碎而深的中小单子，吃不下也看不上。

这背后还有一道更硬的护城河，我在第五篇反复讲过：FDE 公司提前抽象了适配企业业务的结构化框架，而大模型公司没有一线项目的积累，做不了这个粒度的抽象；更何况企业的私有数据治理和萃取，必须在现场完成——大模型公司根本碰不到这些数据。

两种模式，两本账

前面讲了“大厂做标杆大场景、垂直 FDE 做中小碎单子”的分工格局。落到具体做哪门生意，其实是两种完全不同的模式，算的也不是同一本账。

	大客户模式	中小企业模式
典型客户	金融、国央企、大型集团	中小微传统企业（外贸、律所、制造业等）
核心价值	合规安全、驻场陪跑、深度定制	行业模板、轻量适配、快速见效
客单价	30 万—100 万/年	2 万—10 万/年
交付方式	重驻场，FDE 长期嵌入客户流程	轻交付，行业模板 + 少量适配
壁垒来源	私有数据萃取、业务认知深度、合规经验	行业模板复用、规模效应、快速交付能力

	大客户模式	中小企业模式
关键风险	项目重、人力成本高、难规模化	客单价低、客户留存难、模板不够深
适合谁	有行业积累、能扛长周期的团队	有产品化能力、能快速复制的团队

两种模式没有高下之分，但逻辑完全不同。大客户模式卖的是“深度理解和长期信任”，中小企业模式卖的是“标准化模板和快速交付”。一本书很难同时教两门生意，所以后面我说的“该躲开的”和“该盯住的”，主要针对中小企业模式——这也是大多数创业团队更现实的切入点。但如果你有金融、国央企的深度资源和耐心，大客户模式的天花板同样很高，第五篇那几个案例就是佐证。

三个判断框架：哪些项目值得投

落到具体怎么挑项目，我总结了三个判断框架，投资人和创业者都能用。

框架一，药厂—医生分工验证。 问三个问题：这个项目是不是落在“医生”（落地）环节，而不是去挤“药厂”（造模型）？是不是做深度定制 + 长期持续服务？是不是避开了通用赛道、聚焦了垂直细分？

框架二，客户价值判断。 客户能不能一句话说清“用了之后能省多少钱、多赚多少钱”？决策链是不是足够短（最好 CEO 能直接拍板）？是不是需要持续付费、而非一锤子买卖？

框架三，抗吞噬能力判断。 是不是贴近线下客户端、和客户建立了长期信任绑定？是不是做深度个性化、大厂的通用产品满足不了？是不是苦活累活、大厂没动力切进来？

该躲开的，和该盯住的

把上面三个框架反过来用，就得到一张清单。

坚决避开五类项目： 低频非刚需的伪需求；纯通用大模型项目，以及“既做大模型又做落地”的全栈项目；上来就做通用平台、不聚焦垂直细分的；盯着 500 强

大客户、和传统 SaaS/大厂正面抢存量的；早期就靠烧钱换规模、看不到正向现金流的。

重点盯住四类项目： 垂直细分领域的 FDE 交付项目；聚焦中小微传统产业 AI 降本改造的；FDE 聚合平台型项目（对接需求 + 撮合交付的平台模式）；AI 原生的垂直行业落地团队。

一句必要的冷静

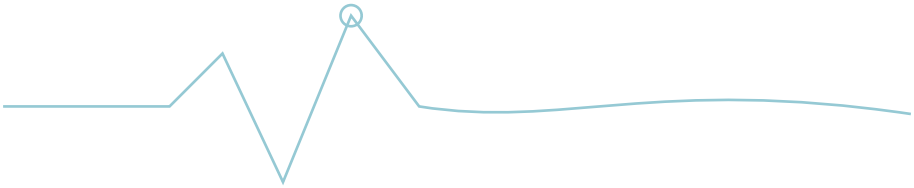
最后，还有一个必须正视的隐患。

FDE 当前还有一个很大的隐患：市场上缺少类似 SAP、Oracle 那样的“基座型产品”。空手拿着大模型去做现场实施，会导致成本不可控、项目没有边际，服务商只能不停地派人驻场维持运营——这其实是不可持续的。冯晓刚把这个隐患看得很透，他常说一句话：现在做 FDE，像极了当年“没有 SAP 就硬做 ERP 实施”，注定又重又难。

那更可持续的稳态模式是什么？是服务商先打造好一个基座产品，把大部分能力交付后、交接给甲方自己运营，只有重大迭代和高难度调整，才需要服务商出手。

我把这盆冷水放在投资这一章的最后，是想说：FDE 是一个真实、巨大、被资本验证的机会，但它绝不是“拿个大模型出去接项目”那么简单。能不能从一堆定制项目里，抽象出可复用的基座和方法论，是区分“一家有壁垒的 FDE 公司”和“一家换了名字的外包公司”的分水岭。

而这，恰好引出了下一章、也是正文最后一章要讲的——现场医生这门生意，到底有几种活法。



FIELD NOTES

医嘱

01 三个判断框架：药厂—医生分工、客户价值、抗吞噬能力。

02 避开五类（伪需求、通用平台、抢存量、烧钱换规模…），盯住四类垂直落地。

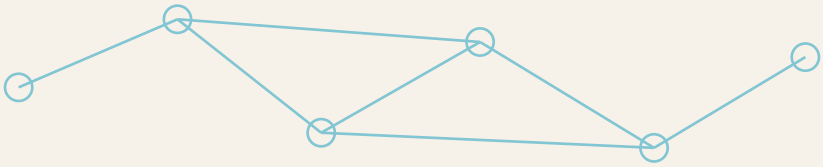
03 缺基座产品是最大隐患——能不能抽象出可复用的基座，是壁垒与外包的分水岭。

CHAPTER 25 · END

第七篇

往哪里走

两种活法与未来



模型会一直变强，但愿意走进现场的人，会一直稀缺。

CHAPTER 26

第 26 章 两种活法与超级个体的时代

主任医生，还是专科医院集团？

——高利明

主 诉 本章要回答：现场医生这条路，到底有几种活法？

讲完了投资的视角，这一章我们回到“人”——如果你决定走上这条路，前面有几种活法？而当越来越多人开始这么干，组织和个体，又会被改写成什么样？

两种活法：主任医生，还是专科医院集团

有一次，我和老朋友高利明见面聊起 FDE。他说，现场医生大体会走向两条路径：一条像“主任医生”，靠个人品牌和专业能力；另一条像“专科医院集团”，靠平台、团队和可复制的体系。这个比方很形象，也成了这一章的起点。

沿着这个比方展开，现场医生大体有两条路径。

第一条，个人品牌路线。 你可以把它理解成“主任医生”。

一个人或一个小团队，深扎一个细分行业，服务少量高价值客户。不追求规模最大，而是追求诊断最准、交付最稳、复购最强。这条路适合刚起步的人先走——成本低、起步快、能快速积累真实案例。

但这里有个前提必须说清：不是会用几个 AI 工具就有人买单。真正能落住的主任医生，必须能把效果算到客户账上——节省了多少时间？减少了多少返工？提高了多少转化？降低了多少错单？老板最终买单的，从来不是“AI 很先进”，而是“这件事确实变好了”。

第二条，平台化路线。 你可以把它理解成“专科医院集团”。

它不依赖某一个超级个人，而是沉淀工具包、方法论、行业模板、评测标准和人才培训体系。目标不是每个客户都从零做起，而是把高频场景做成半标准化，再由现场医生到现场完成最后的适配。这条路更适合要融资、要扩张、要抢市场窗口的团队。

但它更难。因为平台化不是把一堆人聚在一起接项目，而是要回答四个更难的问题：哪些经验可以复制？哪些能力必须留在现场？哪些流程能产品化？哪些责任必须由人来承担？

这四个问题答不清，所谓的平台化，很容易就变成“一家换了名字的外包公司”——这正接上了上一章那个隐患：能不能抽象出可复用的基座，是分水岭。

一套能同时容纳两种活法的服务体系

不管走哪条路，冯晓刚设计的一套“**三级产品服务体系**”，我觉得是个很好的脚手架：

第一层，**标准化产品**——低门槛，让客户先试用、先感受到价值；第二层，**FDE 定制服务**——上门萃取业务知识、梳理调整业务 SOP，匹配客户的实际需求；第三层，**长期陪跑战略合作**——帮企业建立持续迭代优化的机制，和客户共同成长。

这套体系的妙处在于：第一层负责获客和验证，第二层是 FDE 的主战场，第三层锁定长期价值——它既能支撑个人品牌的“主任医生”，也能成为平台化“专科医院集团”的产品骨架。

至于理想的画像，我也一并放在这里，供你对照：**理想的团队**，是年轻的 AI 原生团队，愿意下沉做产业落地，创始人能直接跑现场见客户，战略上先做垂直

细分的“专科医生”、再谈平台化；**理想的客户**，是中小微传统企业，决策链短、需求刚性、未被传统 SaaS 和大厂满足；**理想的商业模式**，是“诊断咨询费 + 年度持续服务费”，持续交付、持续收费。

超级个体：不是一个人，是两三个人

把视角从“创业”放大到“个体”，有一个趋势已经越来越明显：所谓的“超级个体”或“一人公司”，不是指真的只有一个人，而是两三个人就能完成原来需要大规模团队才能做的事。

AI 能力的提升，扩展了个体的能力边界——智能体已经从单纯的对话能力，进化到在特定边界内可执行任务。一个人带着几个智能体，能做原来一个小组甚至一个部门的工作量。这种趋势，正在倒逼组织的层级定义和形态发生快速变化。

传统中层管理者，主要负责信息的加工和传导。当 AI 能更高效地完成这部分工作时，中层的边界和角色必然会改变——要么升级为“数字员工管理者”，要么被重新定义。

而 FDE 恰恰是超级个体最自然的职业形态之一：一个人或两三个人，带着工具包和方法论，走进客户现场，解决一个具体的业务问题。它不需要庞大的组织，但需要极高的个人能力密度。

年轻人的新入口

把视角再往下看一层，现场医生这件事还有一层更大的意义。

AI 正在改变很多基础岗位的结构。一部分重复性工作会被自动化，一部分岗位会被重新拆分，一部分年轻人会发现，过去那种“先做三年基础活、再慢慢成长”的路径变窄了。这当然带来焦虑。

但 FDE 也可能提供一条新的入口。

年轻人不一定一毕业就能独立做现场医生，但他们可以从助理型岗位进入：流程访谈、资料整理、知识库清洗、提示词测试、模型评测、自动化脚本维护、客户培训陪跑。这些工作有一个共同点——它们不是在教室里凭空学出来的，而

是在真实项目里练出来的。如果高校实践、企业实习和 FDE 项目能结合起来，一批 AI 原生的年轻人会更快进入产业现场。

它的意义，不只是多一个职业名称。它可能让年轻人重新获得一种很重要的能力：不是等岗位分配给我，而是我能带着工具去现场，发现问题、定义问题、解决问题。

这也和第十六章那条路径接上了：新人用 AI 快速入门，从助理做起，在一个个真实项目里把“品”养厚——这条成长曲线，比旧时代陡得多。

组织，也在被重新设计

现场医生改变的不只是个体，还有组织本身。

当 AI 把“思”接管之后（第十五章讲的思行品），组织的形态会发生连锁变化。传统工业企业那种“刚性科层”，会逐渐转向“液态网络”——打破固定的部门分工，构建跨领域协同的小单元，让人员和岗位随 AI 应用动态调整；领导力也会从“控制型”升级为“共生型”——领导者从“决策者”变成“系统设计者”，让 AI 提供建议，人聚焦在最关键的战略判断上；流程重构则要“AI 原生设计”，而不是“旧流程 + AI 插件”（这正是第七章那个电力革命的类比）。

你会发现，组织变革要做的这些事，每一件都需要有人在现场推动、翻译、陪跑。这又是现场医生的活儿——所以它不只是一个外部服务商的岗位，更是每个想用好 AI 的组织，内部都需要长出来的能力。

而当这种“AI 原生”推到极致，组织会变成什么样？每个人都带着自己的智能体上班，又意味着什么？这就是下一章、也是正文最后一章要描绘的图景。

FIELD NOTES

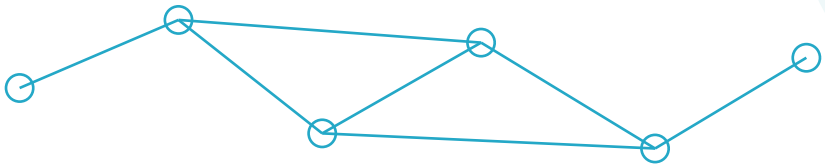
医嘱

01 个人品牌（主任医生）vs 平台化（专科医院集团），起步先走前者。

02 真正落得住的前提：能把效果算到客户账上。

03 平台化要回答四个问题，否则只是换了名字的外包。

CHAPTER 26 · END



CHAPTER 27

第 27 章 AI 原生组织与 IOA：当每个人都有自己的智能体

> 把机器的活交给机器，把人的价值收拢到“品”上。
>
> ——付伟

主诉 本章要回答：当每个人都带着自己的智能体上班，组织会变成什么样？

前面讲的都是“现在”。这一章，我想顺着明略科技吴明辉的一场分享，把镜头往前推一点，看看现场医生这件事，最终会把组织和个体带到哪里。

Agentic 重构：每个角色，配一个自己的智能体

代理 AI（Agentic AI）这个阶段的核心逻辑，吴明辉概括得很精炼：**每个角色都配一个自己的 AI 代理，原来的人只需要管好自己的代理。** 他把这个模式叫“Agentic 重构”，说它可以套到任何一个行业。

以营销为例。过去广告主、广告公司、媒体、消费者之间，是人和人在互动；未来每个人都有自己的 AI 代理，对接、比价、沟通都由代理完成，人只做最终决策。再往后，连消费者选产品都用 AI 代理帮自己筛选——广告行业的核心，就会从“如何影响消费者”，变成“如何影响消费者的 AI 代理”。这是一个全新的赛道（业内已经有人叫它 GEO，生成引擎优化）。

教育、投研、咨询……每个行业都在被这样重构。而重构的过程，恰恰需要有人到现场，把“每个角色该配什么样的代理、代理之间怎么协作、人在哪个节点拍板”想清楚、搭出来——这又是现场医生的活。Agentic 重构不会自己发生，它需要被设计、被落地、被陪跑。

IOA：互联网的第三个时代

如果说 Agentic 重构是组织内部的变化，那 IOA 就是它在更大尺度上的样子。

吴明辉提了一个判断，我觉得很有穿透力：互联网正在进入第三个时代。

PC 互联网，连接的是公开信息，节点是信息；移动互联网，连接的是人和场景，节点是人；而 **IOA（Internet of Agents，智能体互联网），连接的是每个人的 AI 智能体，节点是你的智能体**。

这意味着什么？每个人可以“养”好几只智能体，让它们 24 小时在线干活——你睡觉的时候，它们还在帮你写代码、写报告、跑分析，相当于给每个人增加了 N 个分身。整个网络的增量价值，是爆炸式的。

这背后是一种指数级的增长逻辑。以 AI 写代码为例，吴明辉把它分成四个阶段：手搓代码（顶级程序员一天极限 1000 行）、Copilot 时代（人还要看改代码）、云编码时代（说清需求、AI 自动编写测试，非技术出身的人也能做软件）、到 IOA 时代（多个智能体互相授权、修 bug、交接任务，工程师休假也不影响项目推进，7×24 小时不停产）。每往前一个阶段，人从“干活”里退一步，产能就指数级放大一截。

对创业者和企业来说，这是必须抓住的变革窗口——而抓住它的前提，是有人能把这套东西，真正落进一个具体的组织里。

AI 原生组织的三条第一性原理

那么，一个“AI 原生”的组织，到底该怎么设计？吴明辉给了三条不证自明的原理，我觉得每一条都能直接拿来用。

第一条，合理人机分工——让机器干机器的活，让人干人的活。 核心是按“思—行—品”的光谱来分：所有推理计算、信息整理、执行落地的活，尽量交给 AI；所有品鉴决策、方向选择、信念坚守的活，留给人。很多企业现在恰恰错配了——让人去干推理，让 AI 去做决策，方向完全反了。正确的分工是：AI 把准备工作做好、把选项列出来，人只做最终拍板。

第二条，新型协作方法——共享信息交给 AI，开会只为改变行为。 过去开会，大半时间在同步信息。未来，信息同步交给 AI，人聚到一起，只为做三件事：讨论冲突、做出判断、改变行为。

第三条，让数字员工持续进化。 这需要解决大模型天生的“顺行性失忆”——训练完就不再自动更新。吴明辉团队设计了一套“三层记忆架构”：Soul 层存系统人设、永久不忘（数字员工的底层性格），Memory 层存方法论和经验、定期沉淀（知识体系），Context 层存当前任务、项目结束就归档（短期工作记忆）。这套架构贴合人脑的记忆逻辑，让数字员工“越用越懂你”。这正是第十五章讲的数字员工“长期记忆”能力的技术底座。

人-智能体矩阵：一种具体的组织形态

上面三条是原理，落到具体的组织形态上，有一种正在被验证的模式，值得专门讲一讲——我把它叫作“人-智能体矩阵”。

纵向是职能部门，负责能力迭代。各职能部门训练对应职能的数字员工，持续迭代专精能力——投研部门训练投研数字员工，风控部门训练风控数字员工，每个职能都在养自己的“专才”。

横向是项目组，负责业务目标。项目组带着各职能的数字员工完成业务，人类只做决策——就像一个项目经理带着一支由人和智能体混编的队伍上前线。

有意思的是，这种矩阵会带来一种“职级错位升级”：一线员工升级为“数字员工管理者”（管理好几个智能体，对标原来的管理者），原来的管理者升级为“管理者的管理者”。层级大幅压缩，但不是裁员，而是帮每个人转型——从“自己干活”变成“带着智能体干活”。

这和前面讲的“超级个体”是一脉相承的：两三个人带着一群智能体，就能完成原来一个部门的工作量。区别只是，超级个体是个人层面的，人-智能体矩阵是组织层面的——但底层逻辑一样：人的角色从“执行者”变成“设计者和决策者”。

我品故我在：你靠什么不可替代

讲到最后，绕不开每个人都焦虑的那个问题：AI 来了，我会不会被替代？

吴明辉用存在主义哲学，把人分成了三类，构成一个从“可替代”到“不可替代”的连续光谱。

“我思故我在”——以思考推理为存在意义的人。但今天大模型做推理比人快得多，只要是基于已知信息的推理，AI 都做得更好。所以这类人，是 AI 时代最容易被彻底替代的。而我们大部分办公室白领，从小被训练“会考试、会思考”，天然属于这一类——这是最危险的位置。

“我行故我在”——以和真实世界互动、实际行动为存在意义的人。销售要见客户、工程师要调现场、科学家要做实验。这些和物理世界、真实人类打交道的工作，AI 只能部分替代。

“我品故我在”——以基于个人阅历的直觉品鉴为存在意义的人。这是 AI 完全不可替代的。什么叫“品”？就是 taste，是你基于完整人生经历和当下感受做出的判断。赫尔伯特 · 西蒙说过，世界上根本没有完全理性的决策，因为信息永远不完备——大多数我们以为的“理性决策”，本质都是基于个人 taste 的选择。爱因斯坦提出相对论时，在实验验证之前，没人能从已知体系里完全论证它，但他“觉得世界应该是这样”。这就是品，不是思。

于是吴明辉重新定义了这三个字：**能被 AI 彻底替代的，叫“思”；能被 AI 局部替代的，叫“行”；AI 完全拿不走的，叫“品”。**

这不是非黑即白的分类，而是一个连续光谱。你要做的，就是把自己不断往“品”的方向挪，深耕 AI 拿不走的东西：你的品味、直觉、信念和判断。

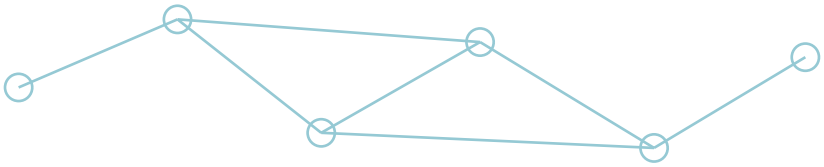
你看，这和第十五章的“思行品”与“最后二十分靠人”、第二十章核保案例里那句“AI 放大的是你本来就有的”、第十六章新人要养厚的那个“品”，讲的其实是同

一件事的不同切面——AI 越强，人就越要往判断力、往品味、往“做架构师而不是修改员”的方向走。

而现场医生，恰恰是把“品”用到极致的那种人：他到现场，靠的不是比 AI 算得快，而是那份“这个城市的口径为什么不一样”“这个算法收不收得住口”“这个结构允不允许 AI 发生”的判断——那是 taste，是品，是 AI 复刻不了的东西。

所以，AI 原生组织也好，IOA 也好，说到底都在指向同一个结论：把机器的活交给机器，把人的价值收拢到“品”上。谁能在自己的领域里，把那份不可替代的判断力磨得足够厚，谁就站在了水准线之上。

讲到这里，正文真的要收尾了。下一节是结语，我会用一句话，把这本书从头到尾想说的那件事告诉你。



FIELD NOTES

医嘱

01 Agentic 重构：每个角色，配一个自己的 Agent。

02 AI 原生组织三原理：合理人机分工、开会只为改变行为、让数字员工持续进化。

03 把自己往“品”的方向挪，做架构师，而不是修改员。

CHAPTER 27 · END

EPILOGUE

结语 站在模型和现实世界之间

> 它不站在模型对面，它站在模型和现实之间。
>
> ——付伟

写到这里，我想兑现上一章结尾的承诺，用一句话说清这本书从头到尾真正想讲的那件事。

但在那之前，先把几个会继续发生的趋势摆清楚。

模型会继续变强。工具会继续变便宜。通用能力会继续被基础设施吞掉——今天还很贵的东西，明天就会像水电一样随手可得。

但有几样东西，不会因为一次模型升级就消失：

企业现场不会消失。流程的差异不会消失。责任的边界不会消失。人的判断、信任、沟通，以及那个“永远没有交付完毕”的持续调试过程，也不会消失。

这就是为什么我笃定，无论模型多强，现场医生这个角色都不会被吞掉。

因为它从来不站在模型的对面。恰恰相反——**它站在模型和现实世界之间**。

没有这条通路，模型再强，也只是停留在云端的一团能力。有了这条通路，AI 才能真正进入一个组织的肌肉、血液和习惯里，留下来，跑起来，并且持续变好。

那句话

这本书从头到尾想讲的那件事，其实就一句：

AI 时代真正值钱的，不是喊“我要做 AI”，而是你能不能把 AI 送进一个真实的企业，让它留下来，跑起来，并且持续变好。

模型层很耀眼。但在整条 AI 产业链里，应用层最稀缺的，可能正是这群愿意走进现场、把药真正喂进病人嘴里、并且守着他吃下去、看着他一点点好起来的人。

从“可用”到“敢于托付”的信任拐点

我想用一个概念，给这本书收尾——**信任拐点**。

你做一个 AI 项目，从 0 分做到 60 分，靠的是技术、工具、方法论，这本书前面讲的大半，都在帮你跨过这 60 分。但从 60 分到真正被客户信任、被组织接纳、愿意持续付费、愿意把更深的业务交给你——这中间有一道坎。我把它叫“信任拐点”：从“可用”到“敢于托付”的那个转折。

跨过这个拐点之前，你是个“乙方”，做一单是一单；跨过之后，你成了客户“持续进化”这件事的合伙人，每一个项目都在为下一个积累信任、积累资产、积累那个越滚越大的飞轮。你在现场一点点萃取出来的行业认知、沉淀下来的私有数据治理能力，会让后一个项目站在前一个的肩膀上——这就是信任拐点之后的复利长什么样。

这门生意最迷人的地方就在这里：它前期很重、很苦、很不性感——蹲现场、啃数据、调提示词、熬阵痛期。但正因为重，一旦跨过信任拐点，它沉淀下来的东西——现场理解、行业认知、私有数据治理能力、与客户共同进化的默契——没有人能轻易抢走。

这是属于现场医生的智能复利。

写给你

最后，我想把这本书，分别说给三种人听。

如果你是一个**业务专家**，做了很多年、懂行业的病，却总觉得自己“不懂技术、做不了系统”——我想告诉你，那道门已经开了。你这些年泡出来的判断力，是 AI 时代最贵的东西，而 AI，恰好能帮你把它放大、落地。前面那几位主角能做成，你也可以。

如果你是一个**刚入行的年轻人**，焦虑于“先做三年基础活”的路径变窄了——我想告诉你，现场是最好的学校。带着工具走进去，先搭结构、敢于提问、持续校验，你成长的曲线，会比任何一个旧时代的前辈都陡。

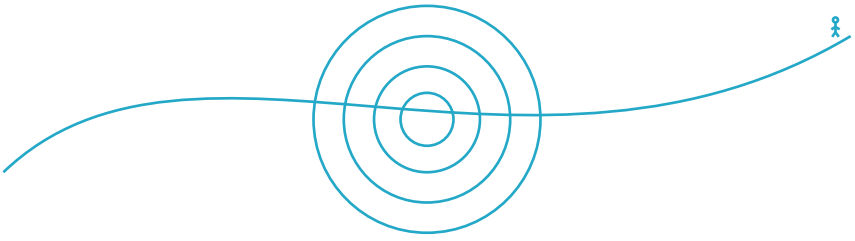
如果你是一个**正在推 AI 的管理者或创业者**——我想告诉你，别再盯着“哪个模型更强”了，多问一句“我们有没有把业务说清楚”。AI 落地真正的瓶颈，从来不是技术，是那个愿意走完“模型到现场”最后一公里的人。

模型会一直变强。但把 AI 放进真实世界、并让它不断长出来的人，会一直稀缺，一直珍贵。

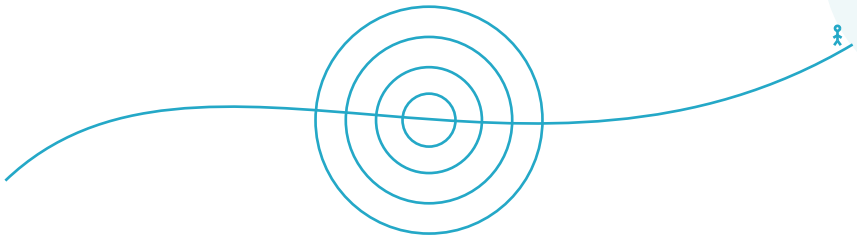
愿你成为那个人。

也愿你身边，有这样一个人。

——付伟



— 全书完 —



APPENDIX

附录

附录 A 现场医生能力自测表 & 三个月入门路线图

A.1 能力自测表

对照下面九个问题，给自己打分（每项 0-3 分）。这不是一张“考你懂不懂 AI”的表，而是一张“看你够不够格走进现场”的表。

懂行业的病（业务判断力）

1. 我能用 5 分钟讲清我所在行业到底怎么赚钱。
2. 我能说出这个行业最大的 2-3 个关键矛盾。
3. 给我一个业务场景，我能找出最重复、最耗人、最能算账的那个卡点。

会用工具的手（解耦与工程力）

1. 给我一个问题，我能把它拆成“哪些环节用检索、哪些用规则、哪些交给模型推理”。
2. 我用过 AI 辅助编程，能在几天内搭出一个能跑的原型（哪怕粗糙）。
3. 我知道怎么用提示词给模型补知识、立边界、给路径。

陪到底的心（交付与信任）

1. 我能把一个 AI 项目的效果，翻译成老板看得懂的账（省了多少人/钱/时间）。
2. 我愿意蹲在客户现场，听一线员工那些“看似琐碎”的话。
3. 我能在项目的“阵痛期”扛住压力、稳住客户，不半途而废。

> 0-9 分：先补业务判断力，别急着碰工具。10-18 分：有潜质，找一个真实小项目练手。19-27 分：你已经具备现场医生的内核，差的只是项目数量。

A.2 三个月入门路线图

第一个月——补“思行品”的品： 选一个你最熟的业务场景，用 AI 跑一遍“发起—审核—决策”，重点练“审”——逐条核对 AI 给的东西哪里对、哪里不对。 **第二个月——练“解耦”和“搭原型”：** 找一个真实的小痛点（自己的、朋友公司的都行），把它拆成子问题，用现成工具搭一个半自动工作流，让人先看到变化。 **第三个月——走完“第一单五步法”：** 找痛点 → 拆流程 → 搭原型 → 跑 7-14 天数据 → 做成案例。核心只有两条：把客户问题拆得足够准，把第一个结果交付得足够快。

附录 B OpenAI FDE 招聘启事与逐句解读

OpenAI 为 Forward Deployed Engineer 岗位写的招聘启事，是理解这个角色最权威的一手文本。下面摘取关键句并做解读。

> “在客户交付与核心平台开发的交汇处运作。”

解读：这句话定义了 FDE 的组织定位——它不是边缘的支持角色，而是“研究能力变现”的关键传动机制，是研究与商业之间的铰链。

> “负责从需求发现、技术范围界定、系统设计、开发构建到生产上线的全流程。”

解读：注意“需求发现”被放在第一位。这呼应了本书反复讲的——客户说的需求和真正的卡点经常是两回事，FDE 的活从“发现真问题”就开始了。

> “通过生产环境的采用率、可衡量的工作流影响力来衡量成功。”

解读：衡量标准不是模型精度，不是代码量，而是“客户有没有真的用起来、用了之后业务指标有没有变”。这就是“卖结果”在 KPI 层面的体现。

> “出差比例最高可达 50%。”

解读：一个细节，但很说明问题。这个岗位默认你要走进现场——这正是“现场”二字的分量。

> “当进展或清晰度取决于代码时，直接上手写代码。”

解读：FDE 不是只动嘴的顾问，关键时刻要自己下场。这对应本书“钢铁侠”阶段。

> “将可复用的工作模式沉淀为工具、操作手册或基础模块。”

解读：这就是“砂石路修成高速公路”——把现场的脏活，沉淀成能复用到所有客户的资产。

附录 C FDE 项目筛选清单

C.1 三个判断框架

- **药厂—医生分工验证：** 是否落在“医生”（落地）环节？是否做深度定制 + 长期服务？是否避开通用、聚焦垂直？
- **客户价值判断：** 客户能否一句话说清省多少/赚多少？决策链够不够短？是否需要持续付费？

- **抗吞噬能力判断：** 是否贴近线下客户、建立长期信任？是否做大厂通用产品满足不了的深度个性化？是否是大厂没动力切入的苦活累活？

C.2 坚决避开的五类

1. 低频非刚需的伪需求项目。
2. 纯通用大模型项目，及“既做大模型又做落地”的全栈项目。
3. 上来就做通用平台、不聚焦垂直细分的项目。
4. 盯着 500 强、与传统 SaaS/大厂正面抢存量的项目。
5. 早期就靠烧钱换规模、看不到正向现金流的项目。

C.3 重点盯住的四类

1. 垂直细分领域的 FDE 交付项目。
2. 聚焦中小微传统产业 AI 降本改造的项目。
3. FDE 聚合平台型项目（对接需求 + 撮合交付的平台模式）。
4. AI 原生的垂直行业落地团队。

附录 D 术语表

- **FDE（Forward Deployed Engineer，前沿/现场部署工程师）**：常驻客户现场，自己定义问题、用 AI 工具组合把方案做出来跑通、陪客户算出结果的人。本书称之为“现场医生”。
- **解耦**：把一个复杂业务问题拆成一组子问题，分别用最合适的手段（检索/规则/模型推理等）解决，再重新组装成能跑的整体。
- **RAG（检索增强生成）**：让模型在回答前先从知识库检索相关材料，以提升准确性、减少幻觉的技术路径。
- **Agent（智能体）**：能围绕一个目标自主调用工具、执行多步任务的 AI 程序。整合了多个 Skill，形成完整可落地的应用形态。
- **Skill（技能/能力单元）**：把一个高水平员工的专业经验，封装成一个可反复调用的单元。轻量、可复用，但普适性差，缺乏统一跨平台标准。

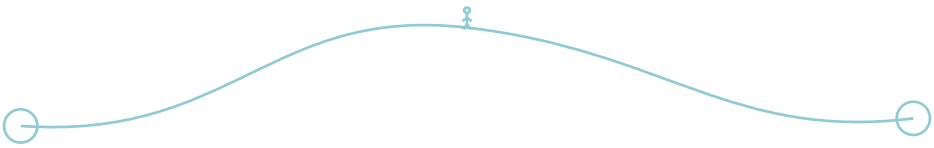
- **思·行·品**：AI 时代人机分工框架——思考搜寻交给 AI，人只做发起、审核、决策三件事；真正的瓶颈在“品”（判断）。
- **数字员工**：被封装成可调用技能、有统一入口和长期记忆的 AI 工作单元。
- **矩阵式智能体架构**：纵向各职能部门训练对应数字员工，横向项目组带队完成目标，人只在关键节点决策。
- **硬编码核／逻辑核**：金融等场景下的分工——甲方负责接口、生产部署、数据落地（硬编码核），乙方负责架构、算法、提示词、质检（逻辑核）。即“数据不出门，逻辑送进去”。
- **信任拐点**：一个 AI 项目从“做到 60 分”到“被客户持续信任、愿意持续付费、交付更深业务”之间的临界点。
- **Ontology（本体论）**：在 Palantir 的实践中，指用时间垒起来的行业知识模型——每一个字段定义、每一条规则、每一个异常处理，都是在真实项目中沉淀下来的活的经验。

附录 E 参考资料与数据出处

- OpenAI Deployment Company（2026 年 5 月，承诺资本超 40 亿美元，TPG 领投，收购英国 Tomoro 约 150 名 FDE/部署专家）；Anthropic 企业 AI 合资公司（估值 15 亿美元，Blackstone、Hellman & Friedman、Goldman Sachs 等，收购旧金山 Fractional AI）。来源：TechCrunch、tech.co、PE Hub、SiliconANGLE、MarkTechPost 等公开报道（2026 年 5 月）。
- Palantir FY2025 人均收入约 101 万美元。来源：Palantir 财报及第三方测算（Bullfincher 等）。
- 企业 AI 落地数据：McKinsey《State of AI 2025》（88% 使用 AI、39% 见 EBIT 影响、6% 高绩效者）；Gartner（到 2025 年底至少 50% 生成式 AI 项目在 PoC 后被放弃）；Stanford HAI《2026 AI Index》（2025 年生成式 AI 私人投资约 1708.7 亿美元）；RAND（企业 AI 项目失败约八成为组织问题、两成为技术问题）；BCG（AI 转型应将约三分之二资源投入与人相关的能力）。
- OpenAI FDE 在某大型金融机构的落地：约 98% 采用率、20%-50% 效率提升。来源：ZenML LLMOps Database、The New Stack 等公开报道。
- 政策背景：财政部、工业和信息化部 2025 年中小企业数字化转型城市试点通知。

本书部分一线判断、案例与方法，来自作者付伟的项目实践与公众号“付伟的轻描淡写”系列文章，以及与陈澄波、冯晓刚等同行的交流。案例均已脱敏处理。

> 说明：书中涉及的市场规模测算（如“千亿级机会”“50 万家中小企业”）为基于公开假设的保守估算，仅供判断方向，不构成投资建议。涉及具体企业的案例均经脱敏，公司名称与可识别特征已做虚化处理。本书不构成投资或经营建议；在未完成数据安全与合规评估前，不建议推进企业 AI 项目。



MODEL → FIELD · 让 AI 真正走进现场