

朝朝（AI创业版） on X 万字实战 | FDE 经验分享：小团队如何跑通企业 AI 服务第一单

万字实战 | FDE 经验分享：小团队如何跑通企业 AI 服务第一单



文章导读

这篇文章写给准备做企业 AI 服务的小团队。它不讨论怎样包装一个漂亮的 AI 演示，而是拆解一笔项目怎样从线索进入、经过诊断和报价，最终变成能上线、能验收、能续费的业务系统。文末附有客户筛选表、需求访谈表、报价结构、SOW、验收表、周报、变更单和交接清单框架。

我们今年做企业AI落地之后，目前拿下了几单，客单价最高6位数，这个视频想讲一讲企业AI落地经验。

企业 AI 服务最容易制造一种错觉：Demo 做出来了，项目就完成了一半。

实际情况往往相反。

一个能在会议室里演示的知识助手，也许三天就能搭出来。真正进入企业以后，问题才开始出现：资料能不能用，权限怎样继承，错误答案由谁负责，员工为什么不用，旧流程怎样衔接，系统出了问题谁来处理。

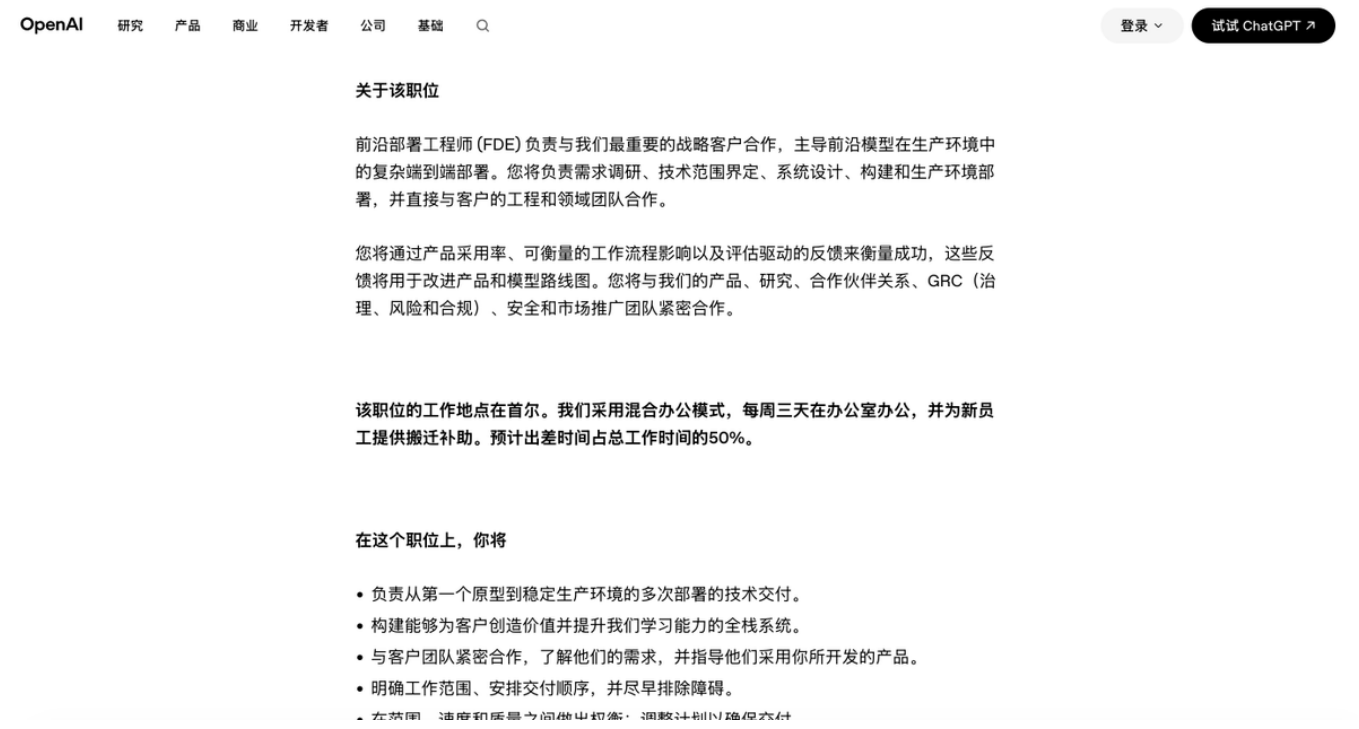
模型只是其中一环。

客户最终购买的重点，也不在于“接入了哪个大模型”。他们关心的是一段原来需要人反复完成的工作，能否在风险可控的前提下变得更快、更稳定，或者成本更低。

这正是 FDE 模式重新受到关注的原因。

FDE 是 Forward Deployed Engineer 的缩写，通常译为前沿部署工程师或前线部署工程师。但如果只把它理解成一个职位名称，很容易低估它的价值。

OpenAI 对 FDE 的岗位描述包括业务发现、技术范围界定、系统设计、建设和生产上线，并以生产采用、可衡量的工作流程影响以及评测驱动的反馈作为成功标准。



这里更重要的词是 **Forward Deployed**：团队必须靠近客户的真实业务，在现场的不确定性里持续做判断。

对于只有 3—5 个人的企业 AI 服务团队，FDE 更适合作为一种 **交付模式**：



这篇文章的角度，也因此和常见的“企业 AI 创业指南”稍有不同。

我们不从注册公司、做官网、准备产品介绍开始，而是从一件更实际的事开始：**一支 3—5 人的小团队，怎样跑通第一笔可验收、可复用、不会把自己拖垮的企业 AI 项目。**

为了把流程讲清楚，全文会使用“内部客服知识助手”作为贯穿示例。它只是一个假设场景，用来展示 SOP，不代表真实客户案例，也不会虚构项目结果和收益数据。

一、先想清楚：企业购买的是 workflow 结果

企业客户常见的第一句话是：

“我们也想做AI提效，你们能不能给个方案？”

这句话的信息量几乎为零。

AI 可以用于 **客服、销售、采购、财务、培训、知识管理和研发**。即使同样叫“知识库”，有人只需要内部搜索，有人希望自动生成答复，有人还要求调用审批、更新 CRM 或对外发送消息。

如果服务团队听完就开始介绍模型、Agent、RAG 和工作流平台，很容易把售前会开成产品功能课。

真正应该追问的是：

- 哪一类人正在完成这项工作？
- 他们每天处理多少次？
- 当前最耗时、最容易出错的是哪一步？
- 结果不好会造成什么后果？
- 哪些数据和系统参与其中？
- 哪些决定可以由 AI 辅助，哪些必须由人确认？
- 项目上线后，谁会真正使用？

一家企业说“想做智能客服”，背后可能存在完全不同的目标：减少新人培训时间、缩短平均响应时间、统一产品口径，或者降低重复转人工的比例。

目标不同，方案、数据、评测和报价都会不同。

所以，小团队的第一条服务原则应该是：

不按 AI 功能卖项目，按业务 workflow 卖结果。

这个判断也符合当前企业 AI 落地的变化。McKinsey 的调研指出，在其测试的多项组织实践中，工作流重构与生成式 AI 带来的 EBIT 影响关系最明显；很多企业仍未获得组织层面的财务影响，原因之一正是没有把 AI 放进端到端流程。

对服务团队来说，这意味着项目立项时不要写：

\\\'\\\'\\\'Plain Text

建设一个基于大模型的企业客服知识助手。

\\\'\\\'\\\'

更好的写法是：

\\\'\\\'\\\'Plain Text

在客服人员处理产品使用问题时，
从已审核的知识范围内提供带来源的答复草稿，

降低资料查找时间，统一回答口径，
高风险或证据不足的问题必须转人工处理。

\\\'\\\'

前一句只描述技术，后一句才开始形成可交付边界。

二、3—5 人 FDE 团队，应该怎样分工

小团队最大的风险在于责任混乱：所有人都在做所有事。

售前承诺没人记录，交付范围没人控制，工程师被客户临时需求牵着走，项目负责人又不敢拒绝。项目越多，团队越忙，利润却越来越薄。

3—5 人团队不需要复制大公司的部门结构，但必须覆盖四类责任。

1\ 业务与交付负责人

负责客户筛选、需求访谈、范围确认、项目节奏和关键决策。

这个人不能只会销售，也不能只会写项目计划。他需要理解客户业务，并能把模糊需求翻译成工程团队可以执行、客户可以验收的任务。

2\ AI / 全栈工程负责人

负责模型、知识检索、Agent、工作流、接口、前后端和部署。

小项目里可以由一名全栈工程师承担，复杂项目再按数据、平台或前端拆分。

3\ 业务分析与采用负责人

负责整理流程、知识和规则，建立测试问题，组织培训，收集真实用户反馈。

很多团队会省掉这个角色，结果做出来的系统技术上能跑，业务人员却不会用，也不愿用。

4\ 安全、数据与领域专家

这类能力不一定全职配置，可以通过顾问或合作伙伴补齐。

涉及人事、财务、医疗、法律、生产控制和敏感客户数据时，不能让普通项目经理代替专业人员做最终判断。

一个最小配置可以是：

人员	核心责任	不应承担的责任
:---	:---	:---
负责人 / FDE Lead	获客、诊断、范围、交付、客户沟通	独自承诺全部技术细节
AI 全栈工程师	原型、集成、部署、监控	替业务方确认知识正确性

| 业务运营 / BA | 流程、内容、测试集、培训、反馈 | 未经授权决定安全边界 |
| 弹性专家 | 安全、合规、领域审核 | 长期承担日常项目管理 |

如果团队有 5 个人，可以增加一名数据/集成工程师和一名客户成功负责人。

人数不是重点，**责任是否有人明确承担才是重点。**

三、获客第一步：不要找“想做 AI”的客户

“对 AI 感兴趣”不是合格线索。

很多人愿意参加分享会、看演示、讨论趋势，却没有明确业务负责人、可用数据和预算。小团队如果把大量时间花在这类客户身上，很容易获得很多热闹的会谈，却没有一笔能进入交付的项目。

更值得优先接触的客户，通常同时具备五个信号：

某项工作高频、重复，已经形成明显负担；

业务负责人愿意参加，不只让 IT 部门了解；

有一批可以用于试点的数据或材料；

愿意安排真实用户参加测试；

对成功结果有初步定义，并愿意为试点 **投入资源**。

投入资源是最重要的。

建立一张线索评分表

可以给每个线索按照 0—2 分评估：

维度	0 分	1 分	2 分
痛点强度	好奇了解	有不便	已影响成本、收入或风险
业务负责人	未出现	偶尔参与	明确负责并推动
数据准备	不知道在哪	可以收集	已有可用样本
使用频率	偶发	每周发生	每天高频发生
验收意愿	只想看 Demo	愿意试用	愿意共同定义指标
预算与时间	完全未知	需要内部申请	有窗口或预算负责人

这张表不是科学模型，只是帮助小团队把有限售前时间放到更可能成交、也更可能交付成功的客户上。

小团队的四个获客入口

入口一：已有关系中的真实问题

第一批客户往往来自信任，而不是广告。

可以从过去服务过的企业、行业朋友、培训客户和合作伙伴中，寻找一个已经反复出现的问题。

沟通时不要问“要不要做 AI”，而要问：

“你们现在有没有一项工作，每周都有人重复查资料、整理信息或回答同类问题？”

入口二：付费诊断工作坊

与其免费做一个没有边界的 POC，不如提供半天或一天的流程诊断。

交付物包括现状流程、问题清单、AI 适用点、数据清单、风险边界、试点建议和粗略预算区间。

客户即使暂时不做项目，也获得了一份可用结果；团队也避免无限免费售前。

入口三：自媒体营销获客

不要只发布“十大 AI 工具”或模型新闻。

更有效的内容是把某个行业流程讲透，例如：

- 经销商售后问题怎样进入知识库；
- 外贸询盘怎样完成客户背调和分级；
- 培训机构怎样从课程资料生成答疑助手；
- 制造企业怎样整理设备故障和维修经验。

内容越接近工作流，越容易吸引真正有问题的人。

入口四：与已有服务商合作

软件实施商、咨询公司、培训机构和行业协会已经掌握客户关系，却不一定具备 AI 交付能力。

小 FDE 团队可以成为联合交付方，但必须提前约定客户归属、售前成本、项目责任、知识产权和回款方式。

四、第一次需求访谈：把“想做 AI”问成可报价的项目

第一次访谈 只负责判断问题是否值得继续，不急着现场给方案。

建议围绕七组问题展开。

1\ 业务目标

- 为什么现在要解决？
- 不做会发生什么？

- 项目优先服务成本、收入、体验，还是风险？

2\ 当前流程

- 谁在什么情况下启动流程？
- 输入从哪里来？
- 中间经过哪些人和系统？
- 最终结果交给谁？
- 哪一步等待最久、返工最多？

3\ 工作量基线

- 每天或每月处理多少次？
- 每次平均需要多久？
- 当前错误、转人工或返工比例是多少？
- 是否存在季节性波动？

如果客户无法提供精确数字，先记录估计范围和数据负责人，不要把口头印象直接写成项目收益。

4\ 数据与知识

- 资料存放在哪里？
- 哪些是正式版本？
- 是否存在冲突和过期内容？
- 哪些数据不能进入外部模型？
- 权限能否从原系统继承？

5\ 风险和例外

- 哪些错误可以接受？
- 哪些错误一次都不能发生？
- 什么时候必须拒答或转人工？
- 是否涉及个人信息、商业秘密或受监管数据？

6\ 用户与采用

- 第一批用户是谁？
- 他们现在使用什么工具？
- 为什么会愿意改用新流程？
- 谁负责培训和收集反馈？

7\ 决策与预算

- 谁是业务负责人？
- 谁审核技术和安全？
- 谁签合同、谁验收？
- 希望在哪个时间窗口上线？

访谈结束后，不要立即发一份几十页的宏大方案。

先用一页纸回传你的理解：

项目理解确认

目标用户

当前流程与痛点

建议试点范围

暂不包含的内容

可用数据与系统

关键风险

需要客户补充的信息

下一步建议

客户确认这张纸以后，项目才进入正式方案阶段。

五、不要一上来做免费 POC，先卖“诊断”

免费 POC 很容易变成小团队的陷阱。

客户不断补充材料和需求，团队不断修改，最后客户说“效果还不错，我们再内部讨论一下”。项目没有预算，没有验收，也没有明确结束时间。

更健康的服务阶梯可以分为四层。

第一层：付费诊断

目标是把模糊问题变成可决策方案。

交付物包括：

- 当前流程图；

- 业务基线和目标；
- 数据与系统清单；
- 风险清单；
- 试点范围；
- 验收指标草案；
- 实施计划与正式报价。

第二层：受控试点

只验证最关键的业务假设，不追求完整系统。

例如客服知识助手第一阶段只覆盖一个产品线、一个部门和一组已审核文档，不接入全部系统，也不直接对外发送答案。

第三层：生产部署

在试点有效后，再补齐权限、日志、监控、异常处理、系统集成、培训和运维。

第四层：持续运营

上线后按月或按季度提供内容更新、评测回归、模型与成本优化、故障处理和新场景扩展。

这四层的价值在于，每个阶段都有独立结果和继续决策的入口。

客户可以在证据不足时停止，服务团队也不会把所有风险压在一张总价合同上。

六、报价：不要按“做一个 Agent 多少钱”来算

企业 AI 项目最难报价的原因，是系统表面看起来相似，背后的交付复杂度却完全不同。

同样是知识问答，50 篇已整理好的公开文档，与几万份包含扫描件、表格、旧版本和部门权限的企业资料，完全不是同一项工作。

小团队可以把报价拆成六个部分。

1\ 诊断与方案

包括访谈、流程梳理、数据盘点、风险分析、架构和验收设计。

2\ 核心建设

包括知识处理、检索、Agent、工作流、前后端和提示词等核心能力。

3\ 系统集成

包括 SSO、权限、飞书、企业微信、CRM、ERP、工单和其他接口。

4\ 测试与验收

包括测试集、评测规则、UAT、回归测试、性能测试和安全检查。

5\ 上线与采用

包括部署、培训、文档、灰度使用、反馈处理和交接。

6\ 风险与持续服务

包括需求变更缓冲、第三方费用说明、模型调用、运维和支持等级。

可以使用下面的公式做内部估算：

```
\`\`\`Plain Text
项目报价
= 预计人力成本
+ 第三方与基础设施成本
+ 项目管理和沟通成本
+ 风险缓冲
+ 合理利润
\`\`\`
```

不要把“模型调用费”当成项目主要价格依据。

客户付费购买的是问题诊断、系统建设、业务适配、风险控制和交付责任。模型只是成本构成之一。

一份可复用的报价表

模块	工作内容	交付物	数量/周期	费用
:---	:---	:---	:---	:---
诊断	访谈、流程、数据、风险	诊断报告	\`[填写]\`	\`¥____\`
试点	最小场景建设	可运行试点	\`[填写]\`	\`¥____\`
集成	账号、权限、业务系统	接口与配置	\`[填写]\`	\`¥____\`
评测	数据集、指标、测试	验收报告	\`[填写]\`	\`¥____\`
上线	部署、培训、文档	生产版本	\`[填写]\`	\`¥____\`
运维	监控、更新、支持	月度服务	\`[填写]\`	\`¥____/月\`

文中不提供“行业统一价格”，因为团队能力、客户规模、数据环境和风险等级差异太大。

真正应该统一的是报价结构，而不是一个看似精确的数字。

报价单必须同时写清四件事

包含什么：明确功能、数据、用户和系统范围；

不包含什么： 例如历史数据清洗、第三方采购、移动端或私有化；

客户提供什么： 数据、账号、专家、接口、测试人员和审批；

变化怎样处理： 什么算需求变更，怎样评估费用和工期。

报价不写边界，项目就会在交付阶段重新谈判。

七、SOW：把售前承诺变成双方都能执行的范围

SOW 是 Statement of Work，也就是工作说明书。

对于小团队，它比一份华丽方案更重要。SOW 负责把销售语言翻译成项目语言。

一份可执行 SOW 至少包含：

项目工作说明书

- ## 1. 项目背景与业务目标
- ## 2. 目标用户与使用场景
- ## 3. 项目范围
- ## 4. 明确不在范围内的事项
- ## 5. 客户提供的数据、系统与人员
- ## 6. 服务方交付内容
- ## 7. 项目里程碑与时间计划
- ## 8. 验收指标与测试方法
- ## 9. 数据、安全与权限要求
- ## 10. 需求变更流程
- ## 11. 上线、培训与交接
- ## 12. 付款节点与终止条件
- ## 13. 知识产权与保密约定
- ## 14. 第三方服务及费用

其中最容易被忽略的是“不在范围内”。

例如，试点只处理中文文本，不包含扫描件 OCR；只覆盖内部员工，不直接面向客户；只从指定知识库读取，不自动修改源文档。

这些内容如果不写，客户往往会自然地认为“以后都可以顺便加上”。

八、交付第一步：先建立基线，不要急着搭系统

项目启动以后，团队最容易犯的错误是立刻开始做原型。

原型很快出现，能带来短期兴奋，却可能验证了一个错误问题。

第一步更重要的工作，是冻结一版可追溯的基线。

业务基线

- 当前流程要经过几步？
- 每月大约处理多少次？
- 平均处理时间是多少？
- 哪些错误或返工最常见？
- 当前满意度或服务水平怎样？

数据基线

- 原始材料有多少？
- 格式和语言是什么？
- 正式版本怎样识别？
- 敏感等级如何划分？
- 谁负责内容正确性？

技术基线

- 系统运行环境是什么？
- 身份和权限怎样管理？
- 能否访问相关接口？
- 日志与监控怎样保存？
- 是否允许调用外部模型？

评测基线

从真实业务中收集一批问题，记录当前人工处理结果、使用来源和风险分类。

如果项目上线前没有基线，上线后就很难证明改进来自哪里。

九、用两周做出第一个“可评测版本”

小团队追求速度没有问题，但第一版的目标应该是 **可评测**，而非看起来完整。

以内部客服知识助手为例，第一版只需要完成：

\\\'\'\'\'Plain Text

员工输入问题

- 系统从指定资料中检索
- 返回答复草稿和来源
- 证据不足时明确拒答
- 用户可以反馈是否有帮助

\\\'\'\'\'

暂时不需要做：

- 覆盖所有产品线；
- 自动对外发送；
- 复杂多 Agent 编排；
- 大量视觉包装；
- 每个系统全部打通。

第一版结束时，至少应该产生四个东西：

可运行的测试环境；

一组真实测试问题；

每个问题的模型结果和引用；

错误分类与下一轮修改计划。

OpenAI 的 Evals 能够把评测标准和数据源组织成可重复运行的评测，并在不同模型或参数配置上比较结果。

无论使用哪套平台，小团队都应该保留同样的思想：每次改模型、知识切分、Prompt 或工作流，都要重新跑一遍固定测试集。

没有评测集的 AI 项目，只能靠演示时的运气判断质量。

十、验收：不要写“回答准确、体验良好”

“回答准确”“系统稳定”“用户体验良好”都不能直接验收。

每个指标必须回答四个问题：测什么、用什么数据、谁来测、达到什么条件算通过。

企业 AI 项目可以从五个维度设计验收。

1\ 业务结果

例如平均资料查找时间、首次响应时间、重复工单比例或新人独立上岗时间。

2\ AI 质量

例如答案正确性、引用有效性、拒答能力、冲突识别和高风险问题处理。

3\ 技术质量

例如响应时间、可用性、并发、错误率、日志完整性和恢复能力。

4\ 安全与治理

例如权限隔离、敏感信息处理、操作审计、数据删除和人工审批。

5\ 用户采用

例如试点用户覆盖、实际使用频率、有效反馈率和培训完成情况。

一张验收表可以这样设计：

验收项	测试数据	测试方法	目标值	责任人	结果	
:---	:---	:---	:---	:---	:---	
高频问题正确性	已确认问题集	人工盲评	\ `[双方约定]\ `	业务专家	待测	
来源可追溯	同一问题集	检查引用	\ `[双方约定]\ `	项目组	待测	
无答案处理	无依据问题集	检查拒答	\ `[双方约定]\ `	风险负责人	待测	
权限隔离	多角色账号	越权测试	不得泄露	安全负责人	待测	
响应性能	压测脚本	P95 统计	\ `[双方约定]\ `	技术负责人	待测	
真实采用	试点用户日志	周度统计	\ `[双方约定]\ `	业务负责人	待测	

这里故意不填统一目标值。

医疗建议、内部资料搜索和营销文案对错误的容忍度不同。服务团队不能拿一个通用的“准确率 90%”套在所有项目上。

NIST 的 AI 风险管理框架也强调，应结合具体使用场景、风险容忍度和资源来治理、映射、测量和管理风险。

验收指标必须在开发前约定。

等项目完成以后再谈验收，双方很容易对“做完了没有”产生完全不同的理解。

十一、交付过程中，用三张表控制失控

第一张：每周项目周报

项目周报
本周完成
下周计划
指标变化
当前风险
需要客户决定的问题
需要客户提供的材料
范围与工期变化

周报要把阻塞和决策提早暴露，无需用它展示团队多努力。

第二张：风险登记表

风险	发生概率	影响	负责人	缓解动作	状态
:---	:---	:---	:---	:---	:---
正式资料版本不清	中	高	客户内容负责人	确认权威来源	处理中
接口权限未开通	高	中	客户 IT	明确开通日期	待处理
高风险问题误答	中	高	双方	增加拒答和人工确认	测试中

第三张：需求变更单

需求变更单
变更内容
提出原因
对范围的影响
对工期的影响
对费用的影响
对验收的影响
双方确认

小团队要保护的是变更过程：任何变化都能被看见、评估和确认。

口头说一句“这个应该很简单，顺便加一下”，不能直接进入开发。

十二、上线之后：交付结束，采用刚开始

系统上线后没人用，是企业 AI 项目最安静的一种失败。

没有报错，没有投诉，也没有业务价值。

所以生产上线必须同时准备采用计划。

选择种子用户

先找愿意尝试、业务经验较强、能给出具体反馈的一小组人。

不要第一天就强制全员切换。

把 AI 放进原来的工作入口

员工每天在飞书、企业微信、CRM 或工单系统里工作，就尽量让 AI 出现在这些入口，而不是要求大家额外打开一个很快被遗忘的网站。

建立反馈闭环

反馈不能只有点赞和点踩。

至少区分：没有找到资料、找到错误资料、答案理解错误、权限不应该开放、流程仍然太慢。

安排固定复盘

上线初期每周查看真实使用问题，确认应该修改内容、检索、规则、界面，还是培训方式。

AI 的价值来自 workflow 改变，不来自安装完成。企业调研也反复显示，治理、流程重构和采用机制会影响 AI 能否产生业务结果。

十三、怎样把一次项目变成可复制服务

如果每个项目都从空白开始，小团队永远只能靠堆人增加收入。

FDE 模式的另一半，是把现场有效的做法带回团队，沉淀成工具、Playbook 和组件。OpenAI 的 FDE 岗位说明也明确提到，将有效模式固化为其他人可以复用的工具、手册或构建模块。

每个项目结束后，至少沉淀五类资产。

1\ 行业问题库

客户最常见的流程、数据、风险和反对意见。

2\ 诊断与方案模板

访谈问题、流程图、数据清单、架构和 SOW。

3\ 技术组件

身份权限、知识处理、评测、日志、反馈和常用系统连接器。

4\ 验收数据集

经过脱敏和授权处理后保留问题类型、错误分类和测试方法，而不是擅自带走客户数据。

5\ 运营手册

内容怎样更新、模型怎样替换、成本怎样监控、故障怎样处理。

最终形成的是一套可以快速适配新客户的交付底座，无需追求一个包打天下的“万能平台”。

十四、验收之后，怎样自然进入续费

续费不应该在合同快到期时才开始谈。

上线阶段就应约定谁负责持续运营，以及哪些工作属于新的服务范围。

常见的持续服务包括：

- 知识更新和质量巡检；

- 固定评测集回归；
- 模型与 Prompt 优化；
- 成本和性能监控；
- 权限与日志审计；
- 新部门或新流程扩展；
- 用户培训和运营报告；
- 故障支持与服务等级。

月度复盘不要只汇报调用次数。

更值得展示的是：用户在哪些工作中使用，哪些问题仍然失败，人工处理是否减少，业务指标是否变化，下个月准备解决什么。

如果没有 **持续价值**，运维费会被客户视为额外成本。

如果每月都能看到问题被发现、知识被更新、流程继续改善，续费会成为自然结果。

十五、小 FDE 团队最容易踩的十个坑

1\ 什么客户都接

结果是行业、系统和需求 **高度分散**，所有项目都无法复用。我们现在主做外贸客户。

2\ 免费 POC 没有结束条件

团队投入不断增加，客户却没有决策压力。不要做太多免费的概念验证。

3\ 按模型能力承诺结果

模型更新、数据质量和业务规则都会影响最终表现。

4\ 售前承诺没有进入 SOW

交付时双方依赖不同记忆解释范围。

5\ 只找 IT，不找业务负责人

系统能够运行，却没人推动使用和验收。

6\ 没有基线就承诺降本增效

项目结束后无法证明节省了什么。

7\ 用演示问题代替真实测试集

上线后遇到用户的自然表达和例外情况，质量迅速下降。

8\ 把所有问题都交给 Prompt

数据、权限、流程和产品问题被包装成一句提示词问题。

9\ 不敢做需求变更

为了维护关系不断免费增加范围，项目利润被逐渐吃掉。

10\ 交付完不做沉淀

下一位客户到来时，团队又从第一次访谈开始重新发明流程。

十六、完整 SOP：从一条线索到项目续费

最后，把全文压缩成一条可以直接放进团队手册的流程。

\\\`Plain Text

阶段 0：线索筛选

→ 评估痛点、负责人、数据、频率、验收和预算

阶段 1：首次访谈

→ 确认目标、流程、基线、数据、风险、用户和决策链

阶段 2：付费诊断

→ 输出流程图、数据清单、风险清单、试点范围和验收草案

阶段 3：方案与报价

→ 拆分诊断、建设、集成、评测、上线和运维费用

阶段 4：签订 SOW

→ 明确范围、不包含项、客户责任、里程碑、验收和变更机制

阶段 5：项目启动

→ 确认团队、沟通机制、账号权限、数据和技术环境

阶段 6：建立基线

→ 记录当前业务表现、数据质量、技术条件和测试问题

阶段 7：可评测试点

→ 用最小范围验证核心假设，建立错误分类和改进计划

阶段 8：生产建设

→ 补齐权限、日志、监控、异常处理、系统集成和安全要求

阶段 9：UAT 与验收

→ 按业务、AI、技术、安全和采用五类指标测试

阶段 10：灰度上线

→ 种子用户、培训、反馈、问题修复和逐步扩大范围

阶段 11：项目交接

→ 代码、配置、账号、文档、操作手册、故障流程和未决风险

阶段 12：持续运营

→ 质量回归、知识更新、模型优化、成本监控和新场景扩展

阶段 13：案例沉淀

→ 经客户授权后整理可公开成果，内部沉淀模板、组件和 Playbook

\\\'\\\'\\\'

附：八份可以直接建立的项目模板

模板一：客户筛选表

客户名称：

业务痛点：

发生频率：

业务负责人：

数据状态：

试点用户：

验收意愿：

预算与时间：

主要风险：

下一步：继续 / 观察 / 放弃

模板二：需求访谈表

业务目标：

当前流程：

工作量基线：

输入与输出：

参与人员：

现有系统：

数据与权限：

高风险错误：

必须转人工的情况：

项目负责人：

验收负责人：

模板三：方案一页纸

目标用户：
要解决的工作流：
建议试点范围：
解决方案：
暂不包含：
关键指标：
关键风险：
客户需要提供：
预计里程碑：

模板四：报价表

诊断与方案：¥____
核心建设：¥____
系统集成：¥____
测试与验收：¥____
上线与培训：¥____
持续运营：¥____/月
第三方费用：按实际发生 / 包含额度____
付款节点： ____

模板五：验收表

业务指标：
AI 质量指标：
技术指标：
安全与治理指标：
用户采用指标：
测试数据：
测试方法：
验收人员：
通过条件：
未通过后的处理：

模板六：周报

本周完成：
指标变化：
下周计划：
当前风险：
客户待办：
需要决策：
范围变化：

模板七：变更单

变更内容：

提出原因：

范围影响：

技术影响：

工期影响：

费用影响：

验收影响：

双方确认：

模板八：交接清单

代码与版本：

部署与环境：

模型与第三方账号：

数据与权限：

配置与密钥交接方式：

测试与验收报告：

用户操作手册：

管理员手册：

故障与回滚流程：

已知问题：

持续服务边界：

最后：小团队真正要建立的是交付信用

企业 AI 服务正在变得越来越容易开始。

模型、Agent 平台、知识库和开发工具都在降低原型门槛。一支小团队可能几天就做出过去需要多人协作的系统。

但企业服务从来不只比较谁做 Demo 更快。

客户会观察你是否理解业务，是否敢写清边界，问题出现时能否定位，需求变化时能否控制，上线以后能否帮助员工真正使用。

FDE 模式最值得小团队学习的，也正是这种端到端责任：从问题进入，到系统上线，再到业务结果被测量。

第一单不必很大。

选一个高频、边界清楚、数据可用、风险可控的工作流，认真跑完诊断、报价、交付、评测、验收和交接。

当你能把这条链路稳定完成，再把过程沉淀成模板和组件，小团队才真正拥有了可以复制的企业 AI 服务能力。

如果你正在准备第一单，先拿文末八份模板，把手上的客户线索完整走一遍。