

# 未来企业的 AI 基础设施

企业为什么应该像当年建设 ERP 一样，  
把 AI 当作  
一套需要长期投资的基础设施来建设 ——  
而不是一堆各自为战的试点项目。

第五代企业系统 · 十层参考架构 · FDE 交付模式  
Via <https://fde.name>

## THE STACK

### 十层参考架构

|    |                   |
|----|-------------------|
| 01 | Model 模型          |
| 02 | MCP 协议总线          |
| 03 | Connector 连接器     |
| 04 | Skills 技能         |
| 05 | Workflow 工作流      |
| 06 | Memory 记忆         |
| 07 | Knowledge 知识      |
| 08 | Governance 治理     |
| 09 | Security 安全       |
| 10 | Observability 可观测 |

# 本次汇报的四个问题

我们要回答的不是“要不要用 AI”，而是“用什么样的结构来承载它”。



01

## 为什么现在

五代企业系统的共同规律，以及  
2026 年的投入与回报缺口



02

## 十层参考架构

Model 到 Observability：每一  
层解决什么、边界在哪、失败长  
什么样



03

## 谁来建设

组织缺口与 FDE（Forward  
Deployed Engineer）交付模式



04

## 如何落地

18 个月分阶段路线图、成熟度  
自评与决策请求

Via <https://fde.name>

# 五句话讲完这份汇报

如果只记住一页，请记住这一页。

## 01 AI 已经是基础设施级支出

Gartner 预测 2026 年全球 AI 支出 2.6 万亿美元，其中 **超过 45% 流向基础设施**。这个量级不可能靠项目制消化。

## 02 但回报高度集中在少数企业

BCG：约 **5%** 的企业获得了显著价值，**60%** 投入巨大却几乎没有实质产出。

## 03 失败的原因不是模型不够强

Gartner 归纳四个根因：数据质量差、风险控制不足、成本失控、业务价值不清 —— 全部是**基础设施与治理问题**。

## 04 决定回报的是 workflow 重构

McKinsey 在 25 个变量中发现：**重构 workflow** 对 EBIT 影响最大，而只有 21% 的企业真正做了。

## 05 所以需要一套栈，和一个负责人

十层参考架构定义“建什么”，FDE 模式定义“谁交付”。缺任何一半，都会退回到试点循环。

## CHAPTER ONE

# 01

# 为什么现在 必须把 AI 当基础设施

过去四十年，企业每一次系统换代都遵循同一条规律：  
先是工具，然后是平台，最后成为不可拆除的基础设施。

01 为什么现在

02 十层参考架构

03 谁来建这十层

04 18 个月怎么走

# 第五代企业系统

每一代系统的本质，都是把一类此前无法管理的东西变成可管理的标准件。



每一代都不是替代，而是叠加在上一代之上的新一层地基。

AI Infra 是第五层地基，不是第 N 个应用。

# 换代的三个阶段：工具 → 平台 → 基础设施

规律不变，只是这一次的周期被压缩到了三年之内。



01

## 工具期

部门自行采购，各买各的。价值真实但零散，无法相加。

ERP 之前的进销存、Cloud 之前的虚拟机、今天的 AI 试点



02

## 平台期

重复的部分被抽取成共享能力，出现统一接口与统一责任人。

主数据、数据仓库、私有云平台、今天缺失的这一层



03

## 基础设施期

成为默认前提。新业务假设它存在，拆掉它的成本高于维护它。

没有企业会问“要不要有 ERP”



**今天绝大多数企业的 AI，还停在第一阶段——却已经在按第三阶段的预期考核回报。**

这就是“投入在涨、回报不出现”的结构性原因：**缺的是中间那一层，而不是更多试点。**

# 钱已经在流动，而且量级变了

AI 支出的规模与结构，已经从“创新试水”进入“基础设施采购”区间。

2.6 万亿美元

2026 年全球 AI 总支出  
同比  
+47%

45% 以上

其中流向 AI 基础设施  
是占比最大的单一板块

31.9%

IDC 预测 2025-29 年  
企业 AI 支出年复合增速

1.7%

企业 AI 支出占营收比重  
2026 年将  
翻倍（原 0.8%）



## 这已经超出项目预算的量级

2.6 万亿美元的支出结构里，基础设施是最大单项。任何一家企业按“每年几个试点”的方式立项，都无法对齐这个量级的资本节奏。



## 占营收比重翻倍意味着什么

当一项支出从 0.8% 走向 1.7% 的营收占比，它就不再是创新预算，而是**成本中心**——会被要求有资产、有折旧、有可审计的产出。



## 2026 是拐点年，不是观望年

Gartner 指出企业当前仍偏好“渐进改良型”的战术性 AI 投入，尚未真正释放支出潜力。先建结构的企业会拿到复利。



# 但成果没有按同样速度兑现

失败率不是模型能力问题的证据，而是交付结构问题的证据。

50% 以上

GenAI 项目在 POC 后被放弃

Gartner 2026 年回顾：这是已经发生的实际值，高于其 2024 年给出的 30% 预测。

40% 以上

Agentic AI 项目将于 2027 年底前被取消

原因：成本攀升、业务价值不清、风险控制不足。

42%

企业在投产前放弃大部分 AI 计划

较上一年的 17% 大幅上升；平均每家放弃了 46% 的 POC。

39%

仅此比例报告了企业级 EBIT 影响

且其中多数表示 AI 贡献不足 5% 的 EBIT。

注：本页刻意不采用流传甚广的“95% 试点失败”说法——其出自未经同行评议的工作论文，原文实为“95% 的组织尚未取得回报”，与失败率并非同一指标。

# 为什么会失败：Gartner 的四个根因

把根因逐条映射到架构层，就得到了后文十层参考架构的需求清单。

## ❌ 数据质量差

模型只能放大数据现状。Gartner：到 2026 年，企业将放弃 60% 缺乏 AI-ready 数据支撑的 AI 项目。

→ 对应架构层：Knowledge • Memory

## 🔒 风险控制不足

74% 的企业计划两年内部署 agentic AI，但只有 21% 对自主智能体建立了成熟的治理模型。

→ 对应架构层：Governance • Security

## 💰 成本失控

Token、检索、工具调用的成本在试点阶段可以忽略，在生产阶段呈非线性增长——且默认不可见。

→ 对应架构层：Observability

## ❓ 业务价值不清

缺少统一的评估口径与埋点，无法回答“这个 agent 到底替公司省了多少”。

→ 对应架构层：Observability • Workflow

✅ 四个根因没有一个能靠“换一个更强的模型”解决——它们全部是基础设施与治理的缺失。

# 真正的鸿沟不在模型，在生产化

试点验证的是“能不能”，生产要求的是“能不能一直、安全、可算账地做”。

| PILOT 试点 | PRODUCTION 生产 | 25%   | 仅四分之一的企业将 40% 以上的 AI 实验推进到了生产环境 |
|----------|---------------|-------|---------------------------------|
| 小团队、几个月  | 跨部门、持续运营      | 2/3   | 约三分之二的组织尚未开始在企业范围内规模化 AI        |
| 清洗过的样本数据 | 真实、脏、在变的数据    | 18 个月 | 预估 3 个月的用例，在集成复杂度暴露后可能拖到 18 个月  |
| 隔离环境     | 与现有系统集成       |       |                                 |
| 没有安全评审   | 安全评审 + 合规检查   |       |                                 |
| 无监控      | 监控、评估、持续维护    |       |                                 |

 “试点疲劳”：立一个新试点比把已有试点推上生产更便宜，于是组织不断重启，永远不收敛。

# 拿到回报的企业做了什么不同的事

差别不在模型，在于是否重构了 workflow、并把能力沉淀成平台。

## THE AI VALUE GAP · BCG 2025

5%

Future-built

营收增长 5 倍、成本下降 3 倍

35%

正在规模化

开始产生可见价值

60%

投入大、几无实质价值

钱花了，结构没建

## MCKINSEY · 25 个变量的对照

在测试的 25 项属性中，**重构 workflow** 对企业能否从 GenAI 获得 EBIT 影响的作用最大。

而当时只有 **21%** 的受访企业重构了至少一部分 workflow。



## 规模化成功的相关因素

- 把 AI 嵌入业务流程，而非旁挂
- 建设技术与数据基础设施
- 为 AI 方案设置并跟踪 KPI
- 具备企业级的交付组织与流程

结论：回报差异来自**结构差异**。这正是把 AI 当基础设施建设、而不是当项目验证的理由。

来源：BCG 《The Widening AI Value Gap》(2025-09-30)；McKinsey State of AI (2025-03；2025-11-05)。

## CHAPTER TWO

# 02

# 十层参考架构

从 Model 到 Observability。

每一层回答一个独立问题，也各自有独立的失败模式。

01 为什么现在

02 十层参考架构

03 谁来建这十层

04 18 个月怎么走

# 全栈总览

十层，三个功能带。上五层决定能力，中两层决定质量，下三层决定能否上生产。

|                                                                                       |               |      |         |                        |
|---------------------------------------------------------------------------------------|---------------|------|---------|------------------------|
|  01  | Model         | 模型   | 用什么脑子   | 多模型路由 · 成本 / 能力分级      |
|  02  | MCP           | 协议总线 | 用什么标准接  | 统一工具协议 · 一次接入多处复用      |
|  03  | Connector     | 连接器  | 接到哪些系统  | ERP/CRM/ 数据仓 · 身份与权限透传 |
|  04  | Skills        | 技能   | 会做哪些事   | 可版本化、可测试的原子能力          |
|  05  | Workflow      | 工作流  | 按什么顺序做  | 编排 · 审批断点 · 人机交接       |
|  06  | Memory        | 记忆   | 记得什么    | 会话短期记忆 vs 组织长期记忆       |
|  07  | Knowledge     | 知识   | 依据什么回答  | 混合检索 · 重排 · 图谱增强       |
|  08  | Governance    | 治理   | 被允许做什么  | 策略 · 审计 · 合规映射         |
|  09  | Security      | 安全   | 如何不被利用  | 注入防护 · 最小权限 · 租户隔离     |
|  10 | Observability | 可观测  | 如何知道好不好 | 成本 · 时延 · 工具链路 · 评估分   |

## 能力层

L01-L05

定义 AI 能做什么、怎么被调用

## 上下文层

L06-L07

决定输出质量的上限

## 控制层

L08-L10

决定能不能上生产、能不能规模化

# 使用这张架构图的四条原则

在逐层展开之前，先约定它的读法 —— 否则很容易被当成又一张产品选型图。

## 不是技术栈，是责任划分

这张图的价值不在于“用哪个产品”，而在于每一层都有**明确的归属人、验收标准和失败信号**。分层的目的是让缺失可被发现。

## 每一层都要“建一次、用多次”

判断某个能力是否该进基础设施，只有一个标准：**第二个用例是否能直接复用它**。不能复用的，就还是项目。

## 自下而上决定上限，自上而下决定体验

控制层（治理 / 安全 / 可观测）缺失，上面做得再好也上不了生产；能力层缺失，下面再规范也没有产出。

## 不必十层同时建，但必须十层同时想清楚

路线图可以分阶段，架构决策不能分阶段 —— 后补的治理与安全，代价通常是重写。

 一句话：这是一张**验收清单**，不是一张采购清单。



## L01 Model

模型

这一层回答的问题

用什么脑子？

CAPABILITY

能力层

### ✂ 这一层要建什么

- 多模型并存，按任务难度与成本分级路由
- 统一调用入口、密钥与配额，模型可替换
- 模型版本与提示词纳入变更管理
- 推理成本与时延的分级 SLA
- 敏感场景的私有化 / 本地模型兜底方案

### ⚠ 缺失时的表现

- 各团队自行申请 Key，成本无法归集
- 业务代码硬编码模型，换模型等于重写
- 供应商涨价、降智、下线时没有退路
- 把“效果不好”归因于模型，瓶颈在下层
- 合规问询时，说不清数据流向哪个模型

### 📶 验收信号

- 换主力模型的改动量：应接近配置级
- 单位业务动作的推理成本可按部门归集
- 各模型调用占比与失败率可查
- 模型升级有回归评估集，不靠体感
- 每个场景的模型选择有成本 / 质量依据



### 2026 年的现实

模型层是**最不必担心、却最容易被过度关注**的一层。前沿能力每几个月刷新，把架构绑死在单一模型上是当前最常见的高成本错误——应设计为可替换部件。





## L02 MCP

协议总线

这一层回答的问题

用什么标准接？

CAPABILITY

能力层

### ✂ 这一层要建什么

- 工具调用标准化：一次接入、多处复用
- 工具与数据源以统一协议暴露，不写胶水层
- MCP Server 注册表：归属、权限、版本
- 对接身份系统，令牌不透传、作用域最小
- 自建与第三方 Server 分级准入

### ⚠ 缺失时的表现

- $N$  个 agent  $\times$   $M$  个系统 =  $N \times M$  套私有集成
- 同一能力被五个团队各实现一遍且行为不一
- 无人知道某个 agent 实际能访问哪些系统
- OAuth 作用域过宽，形成混淆代理风险
- 第三方 Server 未经审计即接入生产

### 📶 验收信号

- 新增工具接入耗时：天级而非月级
- 工具复用率：同一 Server 服务多少 agent
- 每个 Server 权限清单可审计、可撤销
- 协议升级不导致存量 agent 失效
- 第三方 Server 有准入评审记录



### 2026 年的现实

MCP 由 Anthropic 于 2024 年提出，已于 2025 年 12 月捐赠给 Linux Foundation 旗下 Agentic AI Foundation —— 从单厂商协议变为中立治理的标准。



## L03 Connector

连接器

这一层回答的问题

接到哪些系统？

CAPABILITY

能力层



### 这一层要建什么

- 存量系统适配：ERP、CRM、数仓、工单
- 权限透传：以发起人身份访问，非超级账号
- 读写分离与写白名单，高风险动作走审批
- 数据出域策略：哪些字段必须脱敏
- 统一的重试、限流与超时约定



### 缺失时的表现

- 为跑通试点给了管理员账号，然后忘了收回
- 权限以 agent 而非用户为单位，越权难追溯
- 各团队自建连接器，重试限流审计各不相同
- 敏感字段随上下文进了模型日志
- 集成工作量被低估，项目一再延期



### 验收信号

- 任一 agent 的有效权限可一处查询与撤销
- 写操作具备幂等键与完整审计轨迹
- 连接器故障时降级为拒绝，而非编造
- 脱敏由平台统一执行，非各自实现
- 接入第二个同类系统的成本显著低于第一个



### 2026 年的现实

Connector 是十层中**最容易被低估工作量**的一层。Deloitte 观察到：预估 3 个月的用例，在集成复杂度暴露后可能延至 18 个月。



## L04 Skills

技能

这一层回答的问题

会做什么事？

CAPABILITY

能力层

### ✂ 这一层要建什么

- 把流程封装为**可复用技能**，而非提示词
- 每个技能有输入、输出与失败契约
- 技能可单测、可版本化、可回滚
- 技能目录：谁在用、用了多少次
- 高风险技能标注并强制审批

### ⚠ 缺失时的表现

- 能力藏在个人的提示词里，人走即失
- 同一动作在不同 agent 里行为不一
- 改一个提示词，不知影响哪些流程
- 无法回答某个 agent 究竟会做什么
- 技能与工具混谈，权限无法收敛

### 🔄 验收信号

- 新用例中**复用已有技能**的比例
- 每个技能有测试用例与通过率
- 技能变更走代码评审与版本发布
- 可导出某 agent 的完整技能清单
- 下线一个技能不引发未知故障



### 2026 年的现实

Skills 层是把试点变成资产的关键 —— 试点交付的是一个能跑的 demo，基础设施交付的是**可被下一个用例直接调用的能力单元**。



## L05 Workflow

工作流

这一层回答的问题  
按什么顺序做？

CAPABILITY

能力层

### 🔗 这一层要建什么

- 多步编排：确定性自主性分工
- 关键节点设人工审批断点
- 长流程状态持久化，支持断点续跑
- 失败补偿：回滚、重试、人工接管
- 人机交接时上下文完整传递

### ⚠️ 缺失时的表现

- 编排逻辑塞进提示词，行为不可预测
- 流程中断后只能从头重来
- 出错时无人知道停在第几步
- 审批只是提示，agent 仍可绕过
- 自主规划用在本该固定的流程上

### ✅ 验收信号

- 端到端完成率与人工介入率分开看
- 任一实例可回放具体步骤
- 高风险动作 100% 有审批记录
- 流程变更不需要改模型提示词
- 重跑失败实例不产生重复副作用



### 2026 年的现实

经验法则：能写成规则的部分不要交给模型推理。自主性用于判断与例外，而非流程主干——主干越确定，越容易审计与规模化。



## L06 Memory

### 记忆

这一层回答的问题

记得什么？

CONTEXT

上下文层

### ✂ 这一层要建什么

- 区分会话短期记忆与组织长期记忆
- 短期：多轮上下文与当前任务状态
- 长期：用户偏好、历史决策、纠错
- 写入策略：什么值得被长期记住
- 记忆的可见性、有效期与删除权

### ⚠ 缺失时的表现

- 每次对话从零开始，纠正过的错再犯
- 把所有历史塞进上下文，成本失控
- A 用户的记忆泄漏到 B 用户
- 错误信息被写入长期记忆并扩散
- 记忆无法删除，触碰数据主体权利

### 🔄 验收信号

- 同类问题第二次无需重述背景
- 单次会话的平均上下文成本可控
- 记忆条目可溯源、可人工纠正
- 按用户与租户维度有隔离测试
- 支持按主体删除其全部记忆



### 2026 年的现实

主流 Agent 平台已把两类记忆做成**独立原语**（如 AWS Bedrock AgentCore 的 short-term / long-term memory）—— 它属于基础设施。



## L07 Knowledge

知识

这一层回答的问题

依据什么回答？

CONTEXT

上下文层

### ✂ 这一层要建什么

- **混合检索**：向量 + 关键词 + 结构化
- 重排与引用回链，答案可核对
- 文档权限与检索权限保持一致
- 知识时效：失效文档必须下架
- 按领域建评估集，度量检索质量

### ⚠ 缺失时的表现

- 只做向量检索，专有名词与编号搜不到
- 答案没有引用，业务方无法验证
- 检索绕过权限，越权看到他人文档
- 过期制度与现行制度同时被召回
- 一切归因于模型幻觉，实为检索失败

### 📶 验收信号

- 答案**可点开引用**并定位到原文
- 检索命中率与重排后精度分开度量
- 权限用例：无权文档不出现在结果
- 知识更新到检索生效的时延可测
- 拒答优于编造，有明确兜底话术



### 2026 年的现实

多数“幻觉”在工程上其实是检索问题。当成模型问题就会不断换模型；当成检索问题，才会去修**分块、重排与权限**。



## L08 Governance

治理

这一层回答的问题

被允许做什么？

CONTROL

控制层

### ✂ 这一层要建什么

- 用例登记与**风险分级**，先分级后放行
- 策略即代码：额度、动作、数据范围
- 全量决策留痕，可回溯到人与版本
- 把外部框架映射为内部控制项
- 模型与技能上线前的评审关卡

### ⚠ 缺失时的表现

- 不知道公司内到底有多少 agent 在跑
- 治理写在文档里，运行时并不生效
- 审计时无法证明某个决策的依据
- 合规部门只能靠一刀切禁止来控风险
- 每个用例重复走一遍自定义审批

### 📶 验收信号

- 可在**一处**列出全部 agent 及其风险等级
- 策略变更即时生效，无需改业务代码
- 任一决策可导出完整证据链
- 外部审计问题能在一天内取证
- 新用例合规审批周期显著缩短



### 2026 年的现实

不必自创框架：**NIST AI RMF**（含生成式 AI 配套 600-1）提供治理语言，**ISO/IEC 42001** 提供可认证的管理体系。



## L09 Security

安全

这一层回答的问题

如何不被利用？

CONTROL

控制层

### 🔗 这一层要建什么

- 把 agent 当作**可被操控的内部人**防护
- 提示注入防护：不信任检索到的内容
- 工具调用最小权限，高危动作双人确认
- 租户与数据隔离，密钥集中托管
- 红队演练与越权用例纳入回归

### ⚠️ 缺失时的表现

- 文档里的一句话就能让 agent 转账或外发
- 同时具备读敏感数据与外发能力
- agent 用超级账号运行，越权无法追溯
- 把安全等同于模型内容过滤
- 第三方工具引入未评估的供应链风险

### 🔄 验收信号

- 注入用例集**常态化回归**并有拦截率
- 任一 agent 的有效权限可一处查询
- 敏感动作全部有二次确认与留痕
- 密钥可轮换，无硬编码凭据
- 安全事件有可复现的回放数据



### 2026 年的现实

风险来自组合而非单点：当同一 agent 同时具备**访问私有数据、接触不可信内容、对外通信**三项能力时，注入即等于数据外泄。





## L10 Observability

可观测

这一层回答的问题

如何知道好不好？

CONTROL

控制层

### 🔗 这一层要建什么

- 每次调用留痕：**成本、时延、工具链路**
- 把评估分数作为一等信号持续采集
- 线上问题可回放到完整上下文
- 成本按部门、用例、模型三维归集
- 质量回归门禁：不达标不发版

### ⚠️ 缺失时的表现

- 只有满意度问卷，没有客观质量指标
- 账单只有总额，无法定位是谁在烧钱
- 用户报错时无法复现当时的上下文
- 模型或提示词升级后靠体感判断好坏
- 故障定位靠翻聊天记录

### 📶 验收信号

- 任一次会话可**完整回放**并定位失败步
- 单位业务动作成本可按月对比
- 每个用例有基线评估集与趋势曲线
- 发版前后质量差异可量化
- 异常成本与失败率有自动告警



### 2026 年的现实

OpenTelemetry 已发布**生成式 AI 语义约定**，把模型、token 用量与工具调用纳入标准链路追踪 —— 无需自建私有埋点格式。

# 十层里，通常缺哪几层

按我们观察到的普遍情况标注。这一页可以直接作为内部自检的起点。

|    |           |          |      |             |    |               |     |      |              |
|----|-----------|----------|------|-------------|----|---------------|-----|------|--------------|
| 01 | Model     | 模型       | 已具备  | 云端 API 早已可用 | 06 | Memory        | 记忆  | 普遍缺失 | 几乎全部从零开始     |
| 02 | MCP       | 协议总线     | 部分具备 | 刚成为中立标准     | 07 | Knowledge     | 知识  | 部分具备 | 有 RAG，缺重排与引用 |
| 03 | Connector | 连接器      | 部分具备 | 有集成，但权限未透传  | 08 | Governance    | 治理  | 普遍缺失 | 写在文档，未进运行时   |
| 04 | Skills    | 技能       | 普遍缺失 | 能力散在提示词里    | 09 | Security      | 安全  | 普遍缺失 | 仍按内容过滤思路做    |
| 05 | Workflow  | workflow | 部分具备 | 有编排，缺审批与续跑  | 10 | Observability | 可观测 | 普遍缺失 | 只有账单，没有链路    |



试点只需要前三层；**规模化卡在 L04、L06 和整个控制层。**

这也解释了为什么“再招几个人做几个试点”无法解决问题——缺的不是用例数量，是这几层公共设施。

## CHAPTER THREE

# 03

## 谁来建这十层

架构图不会自己变成系统。

决定成败的不是选型，而是有没有一个角色对整条链路负责。

01 为什么现在

02 十层参考架构

**03 谁来建这十层**

04 18 个月怎么走

# 现有编制里，没有人拥有这条链路

把十层交给现有任何一个部门，都会缺掉它职责之外的那几层。

## 业务部门

提出用例，但不掌握数据权限与系统集成能力

→ 能说清要什么，做不出来

## 平台 / 数据团队

能建平台，但不深入业务流程细节与例外处理

→ 做得出来，但没人用

## 合规 / 安全

能否决，但缺少提出可行替代方案的手段

→ 只能说不，不能说怎么做



三方都在场，却没有人对“**从用例到生产**”这条完整链路负责。

这就是 AI 项目最典型的失败形态：不是技术不行，而是每一次跨部门交接都损失一部分上下文。ERP 时代我们用“实施顾问 + 业务专家”这套编制解决了同样的问题 —— AI 基础设施需要它的对应角色。

# FDE：一个已被验证的编制

不是新造的头衔——头部 AI 公司正是用这个角色把模型能力变成客户产出。



## Forward Deployed Engineer

前置部署工程师 · FDE

一名工程师，直接坐在业务现场：既写生产代码，也参与定义问题；对用例上线并产生业务结果负责，而不只对交付代码负责。



### 对结果负责

成功标准是业务指标改善，不是需求交付完毕



### 端到端权限

可跨十层动手：改集成、改流程、改策略



### 短周期迭代

与业务方同处一室，按天而非按季度反馈



### 沉淀为公共层

把这次的解法抽成下一次可直接复用的能力



### Palantir

该模式的发源地：工程师驻场，与客户共同定义问题



### OpenAI

公开招聘 FDE：与客户共建部署，反哺产品



### AWS

2025 年宣布投入 10 亿美元扩充此类交付工程编制

# FDE 与三种相近角色的区别

这三种角色都有价值，但没有一种以“沉淀可复用基础设施”为成功标准。

|            | 交付什么       | 成功标准       | 离开后留下          |
|------------|------------|------------|----------------|
| 外部咨询顾问     | 方案与建议书     | 方案被接受      | 一份文档           |
| 厂商售前 / SA  | 可行性验证      | POC 跑通     | 一个 demo 环境     |
| 客户成功 / CSM | 使用率提升      | 续约与活跃      | 培训与陪跑记录        |
| 内部 FDE     | 生产系统与公共能力层 | 业务指标改善且可复用 | 可被下一个用例复用的基础设施 |



关键区别只有一条：FDE 的 KPI 不是交付完成，而是这一层以后还能不能被复用。

# 怎么把这个角色建起来

不需要先建大团队 —— 需要的是一个有明确权限与正确考核口径的小编制。

起步编制：3-5 人，直接向 CTO 汇报

## FDE Lead

1 人

对十层的完整性负责，有权跨部门推进

## 平台工程师

2-3 人

建 MCP、连接器、可观测这类公共层

## 业务派驻

轮换

业务方出人常驻，不是访谈式参与

三个前置条件 —— 缺一条就会退化成又一个项目组

### 01 预算来自基础设施，不来自单个用例

按用例立项，就只会产出用例；公共层永远排不上优先级。

### 02 有生产环境的实际权限

需要能改集成、改策略、改发布，而不是提工单等排期。

### 03 考核复用率，而非上线数量

成功标准是“第二个用例更快”，不是“今年做了几个 agent”。



ERP 时代最稀缺的不是软件许可，是懂业务的实施顾问。这一代最稀缺的也不是模型，是 FDE。

## CHAPTER FOUR

# 04

## 18 个月怎么走

不需要一次建完十层。

但每一个阶段都必须留下下一阶段可以直接站上去的东西。

01 为什么现在

02 十层参考架构

03 谁来建这十层

**04 18 个月怎么走**



# 18 个月：三个阶段，每阶段留下一层

每个阶段的验收标准都不是“上了几个用例”，而是“下一阶段是否更省力”。

## 0-6 个月 打地基

**建** L01 模型网关 · L02 MCP · L10 可观测

**做** 选 1-2 个高价值、低风险用例上生产

**留下** 统一入口 + 成本可见 + 一个可复用工具集

☑ 判据：第二个用例接入工具的时间，比第一个短一半以上。

## 6-12 个月 接系统

**建** L03 连接器 · L04 技能 · L05 工作流

**做** 扩展到 5-8 个用例，跨 2-3 个部门

**留下** 权限透传的集成层 + 可测试的技能目录

☑ 判据：新用例可由业务团队自助组装，不必每次找平台团队。

## 12-18 个月 上治理

**建** L06 记忆 · L07 知识 · L08 治理 · L09 安全

**做** 把治理与安全前置到发布流程里

**留下** 可通过外部审计的运行时控制与证据链

☑ 判据：合规不再是上线阻塞项，而是流水线中的一个自动关卡。



顺序可以调，但**治理与安全不能整体后置**——它们的接口必须在第一阶段就预留出来。

# 八个问题，判断你在哪一段

不需要评估模型能力 —— 这八个问题问的全是基础设施是否存在。

01 能在一处列出所有在跑的 agent 及其权限吗？

L08 治理

02 换掉主力模型，需要改动多少业务代码？

L01 模型

03 新增一个工具接入，是天级还是月级？

L02 MCP

04 任一 agent 的有效权限，能一处查询并撤销吗？

L03 / L09

05 上个月 AI 花了多少钱，能拆到部门和用例吗？

L10 可观测

06 用户报错时，我们能回放当时的完整上下文吗？

L10 可观测

07 第二个用例的交付成本，比第一个低多少？

L04 技能

08 有没有一个人的 KPI 是“这十层的完整性”？

FDE

0-2 个能答  
还在试点阶段

当务之急是 L01/L02/L10 与一个 FDE 编制

3-5 个能答  
有平台，缺控制层

优先补 L08/L09，否则规模化会被合规挡住

6-8 个能答  
已具备基础设施

重点转向复用率与单位成本的持续下降

# 需要今天决定的三件事

不是批一笔预算，而是把 AI 从项目口径改为基础设施口径。



## 认可这套分层作为内部共同语言

后续所有 AI 立项，按十层说明它建了哪一层、复用了哪一层。



## 设立 FDE 编制并明确汇报关系

3-5 人起步，直接向 CTO 汇报，考核口径为复用率而非上线数。



## 把公共层预算与用例预算分开

公共层单独立项，否则永远排不上优先级 —— 这是最关键的一条。

当年没有人问“ERP 的 ROI 是多少”——因为没有它就无法经营。AI 基础设施正在成为同样的东西。

区别只在于：ERP 你有二十年时间慢慢建，这一次窗口要短得多。

FDE 实战指南

# 从个人转型到 企业 AI 落地

通俗认识前沿部署工程师，找到适合自己的成长路线，并学会把 AI 真正应用到企业业务中。

3 分钟认识 FDE →

找到我的能力路线

## 不同背景，都能参与 FDE 式交付

先识别可迁移能力；正式工程岗位还需补齐生产编码与系统交付。



### 业务专家

可迁移能力：领域发现与业务评测



### 技术人员

核心能力：生产工程与系统集成



### 产品 / 运营

可迁移能力：产品范围与组织采用



### 咨询顾问

可迁移能力：客户发现与变革协作



### 大学生

成长重点：用工程项目积累证据



### 实施顾问

可迁移能力：行业流程与现场交付



### 数据分析 / BI

可迁移能力：数据质量与效果评测



### 售前 / 架构师

可迁移能力：架构与跨团队协作

[查看适合你的能力学习路线 →](#)

Via <https://fde.name>