

FDE 个人升级计划

AI Native × AI 产品经理 × Forward Deployed Engineer

训练总纲 V1.0 | 长期训练主约束文档 / 课程路线 / Gate 标准 / 复测规则

AI Native

把 AI 变成核心执行能力

AI Product Manager

技术、机会、方案、评测判断

FDE

从陌生业务到真实交付闭环

先训练底层能力，再进入业务；先教会，再带练，再迁移，再测评。

不以陌生业务经验缺口冒充能力缺口。

这不是课程目录， 而是后续训练的“主约束”

后续聊天、题目、案例和训练任务必须服从本文件。新能力缺口只能登记到能力地图或阶段待补项；除非存在结构性错误，不随意改动主路线。

8 条不可违反的训练原则

01 不凭经历直接定性

既往做事风格与项目经历只用于出题和观察，不直接得出“强/弱/AI Native/FDE”结论。

02 知识 / 能力 / 经验分开

不会某行业知识，不等于逻辑能力差；没做过真实项目，也不等于没有潜质。

03 先教会，再测评

统一遵循：讲解 → 示范 → 带练 → 半独立 → 迁移 → 正式测评。

04 真实 Case 不做早期底测

必须先具备必要知识与方法，真实业务 Case 才能作为综合验收。

05 多样本 + 跨场景

任何能力不能由一道题定性；至少多个独立样本，并包含迁移测试。

06 允许快进，不允许跳 Gate

通过 Gate 就加速；没通过就针对缺口补练，不平均用力。

07 目标是闭环，不是背框架

能判断、能取舍、能落地、能评测，并在不确定性下完成闭环。

08 Bad Case 必须回流

错误、误判和训练失败沉淀成可复用资产，而不是只纠正当次答案。

路线治理规则：新增训练建议必须回答：属于哪个 Stage？训练什么能力？前置 Gate 是否通过？如何验收？

最终目标不是“会答面试题”， 而是形成三层可独立验证的能力

三层能力互相叠加，但不能互相冒充。底层问题解决能力是地基，AI Native 是工作方式，AI 产品 + FDE 是最终业务闭环。

C

AI 产品 + FDE 闭环能力

从业务问题发现、技术 / 机会 / 方案判断，到原型、评测、部署、采用、ROI 和复用完成闭环。

验收

半真实业务 → 真实项目 → 结果与复盘

B

AI Native 工作方式

把 AI 当核心执行能力而非一次性辅助；会拆人机分工、定位失败原因、调 workflow 并稳定复用。

验收

AI 协作实操、迭代记录、稳定性测试

A

底层问题解决能力

面对复杂、模糊、信息不完整的问题，能分清事实 / 假设 / 现象 / 根因，建立优先级并收敛。

验收

非业务逻辑题、信息不完整题、迁移题

THE OUTCOME

不是“懂 AI”， 而是能把模糊问题推进到真实结果。

发现问题 → 建立判断 → 设计人机协作 → 快速验证 → 上线使用 → 评测结果 → 决定扩大 / 调整 / 停止
→ 沉淀 Playbook

能力地图：四个专业层， 一个共同底盘

后续训练不按“热门名词”组织，而按真正能观察和验收的能力行为组织。

S2

AI Native

能力边界判断、人机任务拆解、AI 失败诊断、工作流优化、会学也会忘、边做边学、稳定性意识、复用沉淀。

S3

AI 产品经理

技术判断、机会判断、方案判断、评测 / 结果判断。核心不是背观点，而是在信息不完整时完成取舍。

S4

技术与工程底盘

模型、API、数据、Agent、Workflow、RAG、工具、成本、部署、安全；能判断 AI 方案和 AI 生成代码是否靠谱。

S5-S7

FDE 闭环能力

Discovery、Solution Framing、Technical Framing、Prototype & Build、Eval & Safety、Deployment、Adoption、Business Outcome、Playbook。

共同底盘：底层认知与问题解决

问题定义 · 逻辑拆解 · 因果推理 · 信息分层 · 优先级与收敛 · 不确定性 · 抽象迁移 · 反证纠错 · 学习速度 · 表达压缩

最先训练

不能用什么冒充能力：会用产品术语 ≠ 会定义问题；列很多点 ≠ 逻辑拆解；自信表达 ≠ 信息分层；做得越全 ≠ 有优先级；立即给答案 ≠ 会处理不确定性。

底层认知能力： 先训练“怎么想”，再训练“懂什么”

01 问题定义

区分现象、他人给出的解释、真正问题与解决方案。

不能用：会用产品术语冒充

02 逻辑拆解

把复杂问题拆成清晰组成，避免混层、漏项和重复。

不能用：列很多点冒充

03 因果推理

区分相关、原因、结果、共同原因，并提出验证方式。

不能用：凭经验说“肯定是”

04 信息分层

区分事实、假设、推断、未知、证据质量。

不能用：自信表达冒充

05 优先级与收敛

信息很多时抓关键变量，敢于暂时不做次要事项。

不能用：做得越全越好

06 不确定性处理

不知道时建立假设和验证路径，而不是瞎猜或停摆。

不能用：立即给答案冒充

07 抽象与迁移

从一个案例提炼结构，并迁移到表面不同的新场景。

不能用：记住原案例

08 反证与纠错

新证据出现时能修正模型，不被旧经验绑架。

不能用：坚持原观点

09 学习速度

获得最小必要知识后，能快速用于新问题。

不能用：背知识速度

10 表达压缩

把复杂判断讲清楚，并说明关键前提和取舍。

不能用：说得长

底层能力的验收不是“答对一道题”。至少 3 类非业务场景 + 1 次陌生迁移，稳定完成问题定义、收敛和验证路径。

AI Native：从“会用 AI”升级到“设计 AI 工作方式”

判断谁做什么、失败在哪里、怎样稳定复用，比“Prompt 写得漂亮”更重要。

01

AI 能力边界判断

知道什么适合模型、规则、检索、工具、人工；不把“AI 能做”当成“应该让 AI 做”。

02

人机任务拆解

接到任务先设计谁做什么、在哪些节点人工 Check，而不是人先做完整流程再叫 AI 辅助。

03

AI 失败诊断

输出差时定位 Context、任务粒度、流程、工具、模型、数据、权限或评测问题。

04

工作流优化

从一次 Prompt 进化为可重复流程；逐步减少人工重复劳动。

05

会学也会忘

不把旧最佳实践当永恒规则；模型能力变化后重新验证工作流。

06

边做边学

缺知识时围绕任务即时补齐最小必要知识，并立即验证。

07

稳定性意识

不以一次成功证明可用；关注重复运行、边界条件和失败分布。

08

复用沉淀

把成功做法沉淀成 Skill / Agent / Playbook / Eval / 文档，而不是只完成当前任务。

一次性 Prompt
能完成一次



任务拆解
人 / AI / Tool 分工



Eval
知道什么叫好



Workflow
重复稳定运行



Skill / Agent
沉淀与复用

AI 产品经理的核心： 四类判断，而不是四套模板

01 TECH

技术判断

技术变化会改变什么能力边界、产品形态、业务流程和机会？

训练结果：形成自己的判断，不人云亦云。

02 OPPORTUNITY

机会判断

谁在什么场景有何真实问题？AI 能否产生增量？值不值得做？

训练结果：区分 AI 能力与用户需求，并做价值 / 成本 / 风险判断。

03 SOLUTION

方案判断

信息不完整下，先共识、收敛、拆卡点、比较方案并说明取舍。

训练结果：能解释为什么选 A 不选 B。

04 EVAL

评测 / 结果判断

什么叫好？怎么量化？Bad Case 如何回流？

训练结果：从用户价值到过程 / 结果指标建立 Eval。

OPEN QUESTION STRUCTURE

前提 → 假设 → 判断 → 证据 → 取舍 →
结论

开放题不是“背一段答案”，而是在不确定性下展示统筹能力：做什么、不做什么、为什么、如何验证。

FDE 不是“会部署”， 而是一条从陌生业务到结果的工作链

01

Discovery

快速进入陌生业务、访谈、流程还原、约束识别、目标共识。

02

Solution Framing

把业务问题转成可解决的问题，确定范围、边界、人工 / AI 分工。

03

Technical Framing

模型 / API / Agent / Workflow / RAG / 数据 / 权限 / 成本 / 延迟 / 可靠性判断。

04

Prototype & Build

用 AI coding、现成工具和最小实现快速产出可验证 Demo。

05

Eval & Safety

定义指标、测试集、Bad Case、风险边界、人工接管、回滚。

06

Deployment

模拟 → 小流量 → 真实环境；考虑监控、日志、权限、依赖。

07

Adoption

让真实用户使用，观察工作流是否真的改变，而不只看 Demo。

08

Business Outcome

建立 baseline、ROI、成本模型和扩大 / 停止决策。

09

Playbook

把一次交付转成可复制方案、工具、Skill、文档与评测体系。

闭环判定：上线不是终点。必须继续看真实采用、行为变化、结果指标，并根据结果决定扩大、调整或停止。

S0 → S8: 按“前置依赖”排列的总训练路线

前一阶段未达 Gate，后面的能力即使接触也不作为正式训练主线。训练允许加速，但不能跳过依赖。



训练速度由 Gate 决定：如果底层能力本来就强，可以少量样本后快速通过；某项弱，就集中训练该项，而不是拖慢所有内容。

S0

测量系统与基线

先测“底层潜质与当前起点”，不是测是否已经会做 FDE

目标：建立能力定义、记录规则和轻量潜质基线；不考业务，不因为第一次不会而下结论。

要做什么

- 建立个人能力矩阵：知识 / 能力 / 经验三栏分开。
- 用非业务题观察逻辑、因果、信息分层、抽象迁移、纠错和学习速度。
- 只做轻量基线，不因第一次不会而定性。
- 记录答题过程：先关注什么、是否补假设、是否主动验证、是否收敛。

测量纪律

- 未训练前的错误，只记录“当前起点”。
- 知识缺失导致失败，先补最小知识，再复测能力。
- 诊断必须有证据来源、训练状态和样本数。
- 不同题表现差异大时，记录“稳定性不足”，不取平均掩盖问题。

GATE

能力记录规则稳定；任何诊断都能说明“证据来自哪里、是否训练过、样本数多少”。

BASELINE $\neq$ VERDICT

基线是起点，不是判决。

如果一个人只是“没学过”，不能把知识空白误判成能力缺陷；这条规则贯穿整个计划。

S1

逻辑与问题解决底盘

让后续所有专业训练有稳定的思考骨架

问题定义

现象 / 根因 / 约束 / 方案分离。

信息分层

事实 / 假设 / 推断 / 未知 / 证据强度。

收敛

限制时间、预算、信息，练“最小必要动作”。

自我纠错

故意加入新证据，观察是否能更新判断。

因果与反事实

如果 X 不存在，结果还会发生吗？

优先级

在 10 个正确事项里选最先做的 1~2 个。

抽象与迁移

同一底层结构换生活、游戏、产品、工程等不同表面场景。

核心结果

不是“更会分析”，而是更快抓关键、建立验证路径并停止无效发散。

GATE

至少 3 类非业务场景 + 1 次陌生迁移中，能稳定完成问题定义、收敛和验证路径。

S2

AI Native 工作方式

从“会用 AI” 升级到“设计 AI 工作方式”



GATE
独立把一个重复任务从“AI 辅助一次” 改造成可重复流程，并能用 Eval 说明稳定性。

S3

AI 产品四类判断

形成 AI 产品经理的统筹能力

技术判断	每次新技术都回答：能力变化 → 产品变化 → 业务变化 → 机会 / 风险。
机会判断	用户 / 场景 → 现有链路 → 卡点 → AI 增量 → 价值 / 频率 / 成本 / 风险。
方案判断	先共识目标 → 收敛主问题 → 拆卡点 → 多方案 → 取舍。
Eval	定义“什么叫好”，建立过程指标、结果指标、Bad Case 回流。
开放题表达	前提、假设、判断、证据、取舍、结论；避免背模板。

GATE
未知但可理解的 Case 中，不依赖老师提示完成四类判断，并能解释取舍。

PRODUCT JUDGEMENT

AI 能力 ≠ 用户需求。
“能做”也不等于“值得做”。

真正的产品判断必须同时把价值、频率、成本、风险和现有工作流放到一张桌子上。

S4

技术与工程最小底盘

不是培养传统程序员，而是能判断 AI 方案与 AI 生成代码是否靠谱

模型能力

Token、上下文、推理、结构化输出、多模态、工具调用的基本边界。

数据

结构化 / 非结构化、清洗、标签、向量化、检索、数据质量。

RAG 与工具

什么时候检索，什么时候直接调用业务系统。

部署

本地 / 云端、环境变量、权限、监控、回滚、数据安全。

API 与系统

请求 / 响应、认证、超时、错误、并发、限流、日志。

Agent / Workflow / Harness

何时需要、如何控制、怎样避免过度流程化。

成本

调用量、模型、存储、第三方、开发、运维、错误成本与风险缓冲。

AI Coding

能读懂关键结构、测试、报错和修改方向。

GATE

给定业务量和方案，能做粗成本 / 架构取舍；能读懂并审查 AI 生成的小型系统。

S5

FDE 方法训练

把前面能力串成一条从问题到结果的工作链



Discovery	知道第一次会议问什么、为什么问。
业务流程还原	As-Is、卡点、责任、数据、风险。
Scope	定义第一阶段做什么 / 不做什么。
Solution Architecture	规则、AI、工具、人如何协作。
Prototype	优先验证最大不确定性，而不是先做完整产品。
Eval	离线历史数据 → 模拟 → Shadow → 小流量试点。
Deployment & Adoption	真实使用、监控、人工接管、回滚。
ROI / Playbook	baseline、时间、人力、转化、错误、AI 成本、扩大价值；沉淀 Skill / Agent / 模板 / 文档。

GATE

在引导式 Case 中完成一套完整闭环，关键步骤能说明“为什么这样做”。

S6 → S8: 从半真实项目走向职业化证据

S6

半真实业务项目

- 先给业务资料包：行业、角色、流程、数据字段、指标、术语。
- 允许 AI 快速学习“最小必要知识”。
- 先一起做一轮，再换场景半独立。
- 加入信息不完整、跨角色冲突、预算和时间限制。
- 必须产出：问题定义、Scope、方案、Demo、Eval、风险、ROI、复盘。

Gate：新行业资料包下能快速学习并完成大部分任务；提示次数明显下降。

S7

真实业务与独立闭环

- 选择自己的项目或真实可接触场景。
- 真实数据 / 用户 / 约束下做最小试点。
- 记录每个决策、假设、失败和 Bad Case。
- 上线后看采用、行为变化、结果指标。
- 根据结果决定扩大、调整或停止。

Gate：至少完成 1 个真实闭环：真实问题、真实使用、真实结果、可复现复盘。

S8

职业化与作品集

- Case Study：背景 → 问题 → 判断 → 取舍 → 实现 → Eval → 结果 → 复盘。
- 开放题：技术趋势、Agent、AI Native、竞品动作、机会判断。
- 模拟面试：追问、反驳、信息不足、压力场景。
- 证据：Demo、截图、Eval、指标、Git / 文档、Playbook。
- 按目标岗位 JD 做能力映射。

Gate：能用证据解释做过什么、为什么做、结果如何，而不是只讲概念。

EVIDENCE > DESCRIPTION

能力最终必须能被外部看见、证明和复述。

作品集不是项目截图合集；它是“判断 + 取舍 + 实现 + Eval + 结果”的证据链。

任何新能力都不能“直接考试”

统一采用 7 步训练循环；只有第 6 步才进入正式评分，第 7 步把错误变成下一轮训练资产。



核心原则：不会 ≠ 弱。只有在“教过、练过、迁移过、复测过”之后，能力结论才有意义。

L0 → L4: 能力等级是一条“可迁移程度”阶梯

L4 | 迁移与闭环

高不确定、多约束场景能闭环，并沉淀成可复用方法。

L3 | 稳定能力

陌生场景仍能解释前提、取舍、验证并落地。

L2 | 可应用

简单 / 熟悉场景能独立使用，有基本判断。

L1 | 可理解

能在提示下分析，但结构不稳定。

L0 | 未形成

只能复述概念，遇到变化场景无法使用。

结论生成条件

- 未训练前错误只记录起点，不定义为能力缺陷。
- 知识缺失导致失败，先补最小知识，再复测能力。
- 每项能力至少需要 3 个独立样本。
- 正式“稳定”判断必须有 1 个迁移样本。

评分同时保存

- 结果分
- 过程证据
- 提示次数
- 修正速度
- 迁移表现

不同题表现差异很大时，记录“稳定性不足”，而不是取平均掩盖问题。

防止长对话后失忆： 训练必须沉淀成“可回放的证据系统”

项目不依赖聊天记录生存。主文档、能力矩阵、训练日志、Bad Case、作品集证据共同组成长期训练记忆。

01

训练总纲

固定路线、能力定义、Gate、测评规则。只有结构性调整才升版本。

02

能力矩阵

每项能力等级、证据样本、是否训练、稳定性、缺口。阶段测评后更新。

03

训练日志

每次训练题、答案、关键纠偏、下一次任务。每次训练更新。

04

Bad Case 库

错误类型、触发条件、修正方法、是否复发。发现新错误即更新。

05

作品集证据库

Demo、Eval、数据、截图、文档、复盘。从半真实项目开始持续积累。

CURRENT

当前状态：S0

训练系统重构阶段。Sales Agent Case 只保留为一次“早期观察样本”，不得作为正式能力评分依据。

NEXT

下一步

先建立《能力矩阵 V1》和《底层认知基线题库》。题库以非行业、非专业知识场景为主，测逻辑、因果、信息、收敛、抽象、迁移、纠错与学习速度。

V1.0

建立从零基础潜质到真实 FDE 闭环的主训练路线。

V1.x

措辞、题型、例子等小修订，不改变阶段顺序。

V2.0

只有发现前置依赖或能力模型存在结构性错误时升级。

Route Governance

任何新训练建议必须说明 Stage、能力、前置 Gate 与验收方式。

ONE PAGE SUMMARY

FDE 个人升级计划

真正训练的，是“把不确定性推进成结果”的能力。



通过 Gate 就加速；没通过就针对缺口补练。
不平均用力，不用“努力感”代替能力证据。