

3

第 3 章

AI 产品设计方法论

在 AI 技术迅速渗透各行各业的今天，一个尴尬的现象正在普遍发生：大量产品经理仍在使用传统互联网的设计逻辑打造 AI 产品。产品经理将大模型视为一种“高级插件”，不假思索地嵌入现有产品流程——比如在电商 App 里横加“AI 导购”入口，在办公软件里硬塞“智能摘要”按钮，却忽视了大模型对产品底层逻辑的重构能力。这种“旧瓶装新酒”的做法，往往导致 AI 功能沦为摆设，使得用户使用率低、体验割裂，甚至因“智能不足”引发负面反馈。

笔者曾任互联网大厂研发管理和产品设计的产品研发总监，深耕互联网产品设计和研发领域十余年，目睹了许多从移动互联网到 AI 时代转型的从业者，我深刻意识到：AI 产品的设计不是功能叠加，而是一场关于“人机协作范式”的系统性革命。要打破当前的设计困局，产品经理必须先破除对大模型的认识误区，建立适配 AI 特性的方法论体系。

1. 当前 AI 产品设计的三大认知陷阱

多数产品经理对大模型的理解仍停留在“更强大的算法工具”层面，这种认知陷阱直接导致了三类典型的设计失误。

(1) 陷阱一：将“AI 能力”等同于“用户价值”

笔者曾见到某教育产品的课后练习环节被强行加入“AI 解题”功能——用户上传题目后，大模型直接输出答案。产品经理团队天真地认为“这比传统搜题更智能”，却忽视了学习场景的核心诉求：学生需要的是解题思路引导而非直接给出答案。结果可想而知：功能上线后使用率不足 5%，还被教师群体投诉“助长投机心理”。

这暴露了一个核心问题：产品经理混淆了“AI 能力”与“用户价值”。大模型

的“生成能力”只是工具，真正的价值在于能否解决场景中的核心矛盾。就像上述案例，若将功能设计为“AI 分步解析 + 错题归因”，就能贴合学习场景的本质需求。

（2）陷阱二：用“确定性思维”设计“概率性系统”

传统软件的功能逻辑是“输入确定输出”，比如支付功能只要参数正确就必然完成交易。但大模型的输出具有天然的“概率性”——同样的问题可能得到不同答案，甚至出现事实错误（即“幻觉”）。

不少产品经理对此缺乏认知，在设计时沿用“零容错”标准。例如笔者曾见到某医疗咨询产品的“AI 问诊”功能，因担心误诊风险，将问答范围严格限定在预设知识库内，导致大模型无法处理超出范围的用户提问，最终沦为“高级关键词检索库”。这种设计看似规避了风险，却抑制了大模型的泛化能力优势。实际上，更合理的做法是：在产品层设计“容错机制”——比如明确告知用户“AI 建议仅供参考”，同时提供“人工复核入口”，将概率性输出转化为可控的用户体验。

（3）陷阱三：忽视大模型对“用户认知模型”的重塑

传统产品的用户体验设计，建立在“用户适应系统”的逻辑上——比如用户必须学习复杂的操作流程才能完成任务。但大模型的核心价值之一，是让系统主动适应用户的自然交互习惯（如语音、自然语言等）。

如在职场人常用的办公软件的文档编辑功能中加入“AI 写作”模块时，要求用户必须通过三级菜单才能触发该功能，且需要严格按照“主题 + 风格 + 字数”的格式输入指令。这种设计完全违背了大模型“自然交互”的特性，最终导致用户依旧选择传统编辑器。

2. AI 产品设计的底层逻辑：从“功能驱动”到“能力进化”

大模型与传统技术的本质区别，在于其具备“类人脑”的泛化能力、上下文理解能力和持续进化能力。这要求产品经理从根本上转变设计思维：不再局限于“定义具体功能”，而是构建“人机协作的动态系统”。

（1）以“能力边界”为设计起点，而非“功能清单”

设计 AI 产品的第一步，是明确大模型在当前场景下的能力范围——哪些问题能解决（且效率高于传统方案），哪些问题暂时无法解决。这需要产品经理深入理解大模型的技术特性，比如 GPT-4 在逻辑推理、多模态理解上的优势，以及在精确计算、实时数据处理上的短板。

例如在设计 AI 客服产品时,产品经理需要清晰界定:简单咨询(如“退货政策”)可由大模型独立处理;复杂纠纷(如“物流丢件理赔”)需人机协同;而涉及用户隐私的操作(如“账户密码修改”)则必须由人工完成。这种“分层处理”的设计,既发挥了大模型的效率优势,又通过边界控制避免了体验风险。

(2) 构建“数据闭环”,让产品具备“自进化”能力

传统产品的迭代依赖用户反馈和 A/B 测试,而 AI 产品的核心竞争力在于数据驱动的自我优化能力。产品经理需要在设计初期就规划完整的数据闭环,例如用户交互数据如何收集(如对话评分、纠错反馈等)、如何标注(如意图分类、答案质量标签等)、如何反哺模型训练(如微调数据集构建等)。

某智能问答产品的成功案例值得借鉴:其在产品内设计了“用户纠错”入口,当用户标记答案错误时,系统会自动记录问题类型、错误原因,并将高价值样本纳入微调库。三个月内,该产品的答案准确率从 65% 提升至 89%,用户留存率提升 40%。这印证了一个道理:AI 产品的“智能”不是上线时就定型的,而是通过数据闭环持续成长的。

(3) 重塑“人机协作”范式,而非简单替代人力

优秀的 AI 产品从不追求“完全替代人类”,而是通过人机协作来放大整体效能。产品经理需要思考:在具体场景中,人类的优势(如情感理解、创造性决策)和 AI 的优势(如信息处理速度、重复性工作效率)如何互补?

以医疗类 AI 产品为例,成熟的设计方案往往是“AI 初筛 + 医生复核”:AI 快速分析医学影像并标记可疑区域,医生聚焦关键决策和患者沟通。这种模式既提升了诊断效率(AI 处理速度是人类的 10 倍以上),又通过医生的专业判断规避了误诊风险。反观那些试图“让 AI 独立看病”的产品,它们大多因用户信任不足而失败。

3. AI 产品设计的实践框架:从 0 到 1 的落地路径

基于上述逻辑,笔者总结出一套“三阶设计法”,帮助产品经理系统化构建 AI 产品。

(1) 第一阶段:场景解构,找到“AI 适配点”

在启动设计前,需用“场景五要素”模型拆解目标场景:用户、核心任务、痛点、瓶颈、大模型的独特价值,如图 3-1 所示。

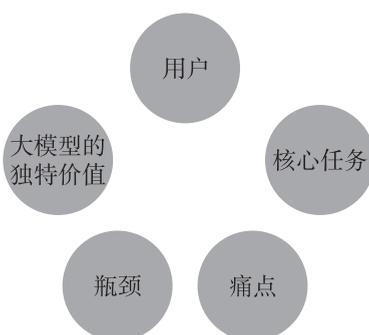


图 3-1 场景五要素

以“企业会议记录”场景为例：传统方案的痛点是“记录不完整、提炼效率低”，而大模型在“语音转写、要点采集、行动项提取”上具备明显优势。但进一步分析会发现，单纯使用 AI 生成的纪要可能遗漏“潜台词”（如领导的语气暗示），因此设计时需保留“人工编辑”入口，形成“AI 初稿 + 人类优化”的协作模式。

（2）第二阶段：能力封装，设计“可控的智能”

在明确场景价值后，需要将大模型能力转化为具体产品形态，核心任务是平衡“智能度”与“可控性”。产品经理需要重点设计的三个层面如图 3-2 所示。

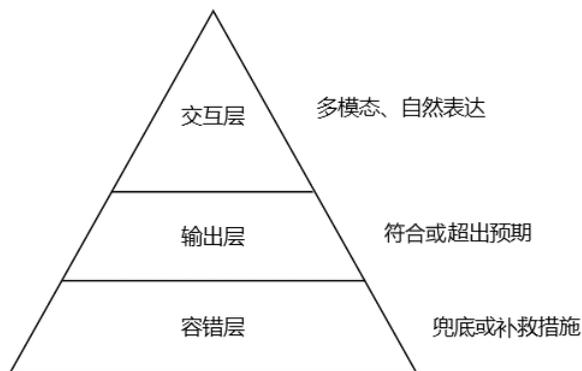


图 3-2 智能可控的平衡三层模型

- ①交互层：如何让用户自然地表达需求（如支持语音、文字混合输入）。
- ②输出层：如何呈现 AI 结果才符合用户预期（如在生成摘要时标注信息来源）。
- ③容错层：当 AI 出错时如何补救（如一键切换人工、历史记录回溯等）。

某款 AI 邮件助手的设计很具启发性：其生成邮件草稿后，会自动标注“建议修改点”（如语气过于正式、信息可能遗漏等），并提供“更简洁”“更礼貌”等一键优化选项。这种设计既发挥了 AI 的生成能力，又通过可控性设计降低了用户的使用门槛。

(3) 第三阶段：灰度验证，快速迭代认知

AI 产品的市场反馈存在太多不确定性，因此需要采用“小步快跑、灰度验证”的策略。产品经理可按“功能复杂度”和“用户影响范围”划分3个验证阶段，如图3-3所示。

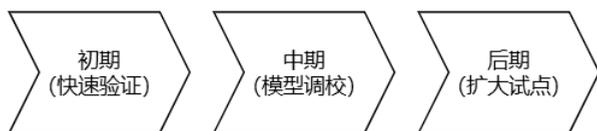


图 3-3 3 个验证阶段

①初期：用提示词工程（Prompt Engineering）快速验证核心假设，比如在现有产品中嵌入轻量化 AI 功能（如用 GPT-4 API 实现的临时导购机器人）收集用户反馈；还可以先在团队内部进行小规模白名单用户的灰度测试，即时收集用户反馈以进入下一轮迭代。

②中期：基于验证结果优化产品形态，搭建基础数据闭环，开始小规模模型微调。

③后期：扩大使用范围，通过用户行为数据持续优化模型和产品体验。

这种渐进式验证，能帮助团队在可控成本范围内快速修正认知偏差，避免在大规模投入后发现方向错误。

在以上阐述的“三阶设计法”体系里，认知 AI 技术常识、运用第一性原理解构场景以及秉承“智快精简”四字诀开展交互设计，这三项能力构成了核心基础。它们不仅是“三阶设计法”得以有效落地实施的关键支撑，更是本章后续内容将深入剖析与展开论述的重点所在。

4. 结语：从“工具思维”到“生态思维”的跃迁

AI 时代的产品竞争，本质上是“认知深度”的竞争。当多数产品经理还在纠结“如何添加一个 AI 按钮”时，少数先行者已经在重构产品的底层逻辑。真正的 AI 产品不是“用大模型做加法”，而是以大模型为支点，重新定义用户价值、重构业务流程、重塑人机关系。对产品经理而言，这既是挑战也是机遇。产品经理们需要跳出传统的功能设计舒适区，主动学习大模型技术原理、理解数据闭环逻辑、探索人机协作新范式。唯有如此，才能设计出真正“智能且好用”的 AI 产品，在这场技术革命中占据先机。

3.1 AI 技术常识

作为产品经理，面对 AI 技术浪潮时，最需要的是“用得上、用得对、用得好”的 AI 能力认知框架。对于产品经理而言，AI 领域的名词不用死记硬背，应知应会的 AI 技术常识的学习关键是结合实际场景理解它们“能做什么”“怎么用”“要注意什么”。

1. 应知应会之模型大小之分

接下来，笔者将对 AI 大模型和 AI 小模型进行详细介绍，深入剖析并明确二者之间的核心区别，见表 3-1。

表 3-1 AI 大模型和 AI 小模型的核心区别

类型	定位	使用场景	效果
AI 大模型	全能型选手	通用场景	能处理复杂语义，但在专业领域或垂直领域直接使用可能出错
AI 小模型	专精型选手	具体的单一功能场景	特定任务效率远超大模型，比如 OCR 识别发票准确率能达到 99%

大模型是一位“全能型选手”，例如 Chat GPT、百度文心一言、DeepSeek、豆包等。它通过海量数据训练，能看懂文字、听懂语音、写代码、做分析，似乎无所不知但又不够专精。

AI 大模型的主要使用场景是较为通用的场景，如智能客服闲聊、文案生成、多轮对话等。使用大模型时，要根据场景校验其应用效果，需要注意的是大模型虽善于处理复杂语义，但在专业领域（如医疗诊断）可能出错。

需要指出的是，产品经理不能指望它“一步到位”，复杂场景的 AI 大模型应用需要搭配垂直领域数据微调（比如利用其诊断疾病前，需要给它喂大量病例数据，才能让它更懂得如何看病）。

而 AI 小模型则是一名“专精型选手”，例如专门的语音识别、光学字符识 (Optical Character Recognition, OCR) 的模型。小模型只擅长某一类任务，但准确率高、速度快、成本低。

AI 小模型的主要使用场景为具体的单一功能场景，如 App 语音输入、发票扫描识别等。使用 AI 小模型时会发现在某些特定任务中效率远超 AI 大模型，比如 OCR 识别发票准确率能达到 99%。

需要指出的是，小模型功能单一，跨任务能力差，例如语音识别模型不能进行

文本翻译，因此切勿寄予 AI 小模型超出其能力范围的厚望。

2. 应知应会之算力和训练

AI 场景离不开算力、数据、算法。其中算力即指图形处理单元（Graphics Processing Unit, GPU）。GPU 相当于 AI 应用的“超级计算器”——大模型训练和运行都需要大量计算，普通电脑的中央处理器（Central Processing Unit, CPU）根本不胜其任，必须用 GPU（比如英伟达 A100）才能快速完成计算任务。当然，从成本角度考虑，使用大模型和 GPU 的产品，GPU 服务器成本会更高；即使用户端需要本地运行 AI 功能（如手机离线语音识别），产品经理也得考虑用户设备的 GPU 性能对产品功能的影响。

和数据、算法相关的则是模型“训练”过程。所谓训练，也就是教 AI 做事的过程。比如想让模型认识“电商商品名称”，就向其喂送 100 万条商品标题数据，让它从中找寻规律。产品经理需要注意的是，垂直场景中通用大模型的适配度往往不高，因为其对行业术语不敏感（比如医学领域的“房颤”“早搏”等），必须用行业数据进行微调训练才能解决其在垂直场景的适配问题。

3. 应知应会之与产品交互相关的 AI 术语

在产品设计中，常用的 AI 术语见表 3-2。

表 3-2 产品设计的常用 AI 术语

AI 术语	释义	使用场景	效果
ASR（语音转文字或语音识别）	把语音变成文字	会议纪要 App 语音输入法 智能音箱指令识别	安静环境下准确率达 95% 以上，但方言、噪声会让准确率下降
TTS（文字转语音）	把文字变成人声	听书 App 视力障碍辅助工具 自动电话通知	能模仿真人语气，但长文本朗读可能显得生硬
OCR（图文识别）	把图片里的文字“抠”出来	发票报销（扫描发票自动填写金额） 快递单信息录入	清晰的印刷体识别率近 100%，手写体、模糊图片容易出错
意图识别	判断用户说的话是想表达什么需求	智能客服（自动分流用户需求） 语音助手（理解指令）	常见意图（如“查余额”“改地址”）识别准确率达 90% 以上，模糊表达（如“这东西不太对”）容易判断失误
抽槽（实体抽取）	从用户输入中提取关键信息	外卖下单（自动填地址和时间）、客服系统（自动记录用户提到的订单号）	结构化信息（如日期、电话号码）抽取准确率高，模糊信息（如“下周三左右”）可能抽取错误

如表 3-2 所示，在 AI 产品设计和开发中，常用 AI 专用术语主要涉及 ASR、TTS、OCR、意图识别、抽槽等。其中：

（1）ASR

ASR（Automatic Speech Recognition，自动语音识别）是语音识别的基础。ASR 将用户的语音变成文字，该功能常见于微信的语音转文字功能。

ASR 的主要场景包括会议纪要、语音输入法、智能音箱指令识别等。当用户周边环境安静时，ASR 识别准确率可达 95% 以上，但方言、噪声降低识别准确率下降，如在地铁里使用该功能可能错漏信息。

产品经理需要注意的是，使用 ASR 能力时最好辅以设计“纠错入口”，允许用户手动修改识别错误的文字。

（2）TTS

TTS（Text - To - Speech，语音合成）能够将文字变成人声，该功能常见于各大导航 App 的语音播报。

TTS 的主要场景包括各类小说听书 App、视力障碍辅助工具、自动电话通知功能等。目前 TTS 已经具备模仿真人语气、语调进行文字朗读的能力，但长文本朗读可能显得生硬。

产品经理需要注意的是，使用 TTS 时最好提供“语速调节”“音色选择”功能，以适应不同用户习惯。

（3）OCR

OCR 能把图片里的文字“抠”出来，该功能常见于扫描身份证提取信息的场景。

OCR 的主要场景包括发票报销应用中扫描发票自动填写金额、快递单信息录入等。OCR 对清晰、干净的印刷体识别率近 100%，但处理手写体、模糊图片容易出错。

产品经理需要注意的是，在复杂场景（如多栏排版的合同扫描）使用 OCR 时需搭配人工审核。

（4）意图识别

意图识别主要用于判断用户说的话是想表达什么需求，比如用户说“我想退这个订单”，意图就是“申请退货”。

意图识别的主要场景包括智能客服（如自动分流用户需求）、语音助手（如语音导航功能需要理解用户指令）等。较为常见的意图识别（如“查余额”“改地址”）

准确率可达 90% 以上,但对于模糊表达(如“这东西不太对”)容易判断失误。此外,在诸如商旅等复杂场景,往往需要实时识别用户意图的切换,如用户在查询机票后不立即购票而是先查询酒店,就是典型的意图实时切换。

产品经理需要注意的是,为了提高意图识别准确率和速度,往往需要在产品里预设“常见意图库”,并通过补充用户反馈不断改善(比如关于退货的不同说法,要让模型知道这是用户的“退货”意图)。

(5) 抽槽

抽槽主要用于从用户输入的语音或文字中提取关键信息,比如用户说“明天下午 3 点送 2 箱牛奶到朝阳区光华路”,抽槽就是把“时间 = 明天下午 3 点”“物品 = 2 箱牛奶”“地址 = 朝阳区光华路”这些信息摘出来。

抽槽的主要场景包括外卖下单(如自动填地址和时间)、客服系统(如自动记录用户提到的订单号)等。抽槽时,程序对结构化信息(如日期、电话)抽取准确率高,但处理模糊信息(如“下周三左右”)时可能抽取错误。

产品经理需要注意的是,需要设置兜底策略以便在抽槽失败时,用引导语让用户补充信息(比如“请问具体送达时间是几点呢?”)。

基于上述 AI 技术常识,产品经理在 AI 产品设计时还需要遵守 3 条铁律:

铁律 1: 勿让技术绑架体验。比如在语音输入功能中,必须保留手动输入选项,因为不是所有用户都习惯语音交互。

铁律 2: 给 AI 留“退路”。所有 AI 功能都应配备人工支持选项,比如智能客服无法解决的问题可以一键转人工;OCR 识别错误的内容应允许手动修改。

铁律 3: 数据闭环优先于完美上线。初期 AI 功能即使准确率 80% 也能上线,但必须设计用户反馈入口(如“这个回答准吗?”“识别错了? 点击这里修正”),以便使用户数据持续优化模型——这就像产品迭代需要用户反馈,AI 迭代更需要!

总而言之,这些 AI 技术本质上都是产品经理的“工具”,产品经理无须掌握 AI 技术的底层原理,相反,产品经理的工作重点应是厘清这些问题:这个功能能够解决用户什么痛点?现有技术能不能支撑功能实现(比如方言多的地区慎用 ASR)?如何通过产品设计弥补技术缺陷?

好的 AI 产品,是让用户在使用过程中感受不到 AI 的存在,但用户会感受到使用过程的便捷。

3.2 运用第一性原理解构场景

第一性原理（The First Principle Thinking）是从最基础的事实出发，层层推导结论的思维方式，不依赖经验或类比。第一性原理的核心思想是“剥离表象，直击本质”，最著名的案例是马斯克思考电动车电池成本时，不局限于“现有电池多少钱”，而是拆解原材料（锂、钴等）成本，再重构生产方式。

第一性原理的使用方式分为如下两步：

步骤一 拆解

将问题拆解至不可分割的“原子问题”。例如，将“用户为什么用外卖 App？”拆解为“省时”“方便”等本质需求。

步骤二 推导

抛开既有方案，从原子问题重新推导解决路径。例如，不照搬“满减优惠”，而是设计“准时达保险”来满足“省时”需求。

第一性原理特别适用的场景总结如下：

①产品创新：需要突破行业惯性，如从“打车按里程收费”到“按时间 + 里程综合计费”。

②复杂决策：在复杂业务场景中使用 AI 设计交互流程时可应用第一性原理。再如资源有限时，应优先解决核心痛点而非跟风模仿产品功能；

③打破思维定式：避免“别人都这么做所以我们也做”的惯性思维，回归用户价值本质。

产品经理需要注意的是，第一性原理不适合快速试错场景，更适合长期战略或底层逻辑设计。

下面笔者将结合在某教育公司应用第一性原理进行产品设计简化的案例，进一步阐述第一性原理的使用方法。

彼时，笔者所在的产品和研发团队正在进行系统重构。原有系统设计允许用户随时可进行线上上课，但用户购买课程的“课消”（教育培训行业术语，指学员实际消耗的课时费用，即家长预缴学费中已通过上课完成结算的部分）不符合预期。为此，团队将系统重构为按学期和上课的模式排课，以加快用户购买课程的“课消”消耗速度。

系统重构开始于当年 6 月初，7 月底便重构完成，8 月系统上线。期间正值暑假，

运营团队为了增收,通过市场调研,发现家长和学生对 KET、PET^① 考试有培训需求,为此运营团队有意开发相关功能以便在暑期开展报名和培训。

需求首先提交到产品经理、前后端技术经理、测试经理处,各经理初步评估后一致认为目前需求工作量较大,系统重构期间已无多余人力进行其他需求支持;但运营总监不想放弃机会,便带各经理前来与笔者协商产品管理和研发资源。

笔者根据第一性原理发现可以简化产品设计,并能实现运营增收的预期。笔者基于第一性原理进行了如下拆解和推导。

1. 拆解

KET、PET 考试培训报名和上课的过程可以拆解为:培训宣发和报名、排课、上课、取消课程、退款 5 个关键节点,如图 3-4 所示。

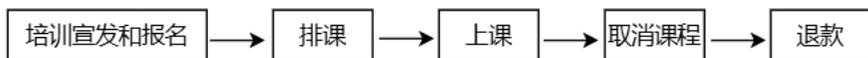


图 3-4 KET/PET 考试培训报名和上课的过程

2. 推导

在上述 5 个关键节点中,培训宣发和报名、排课为本次需求的新增功能,其他为现有系统的既有功能,如图 3-5 所示。

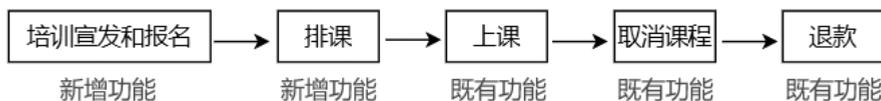


图 3-5 原子功能是否已有

培训宣发和报名、排课看似是两步,但实际可以一步到位——在宣发培训课程的 H5 页面设置几组不同时间段的课程,以供学生和家長自行选择;而非传统模式下先缴费报名,再线下沟通排课的低效模式。于是培训宣发和报名、排课两个环节合并后,仅需 1 个前端工程师、1 个测试工程师、1 个后端工程师,3 天即可完成开发和测试工作。

退款的前提是查询用户的“课消”历史记录,现有系统中已有该功能,因此可以基于已有的上课记录计算用户未完成课时的数量,进而计算得到退款金额,并基

^① KET (key English Test) 是剑桥英语考试系列中的第一级英语语言水平考试, PET (Preliminary English Test) 是其第二级英语语言水平考试。