

AI产品经理 vibe coding 能力要达到什么程度？

 AI产品经理不一定要变成专业程序员，但必须具备“用AI把想法快速做成可验证产品”的能力。

这就是 AI 产品经理的 vibe coding 能力。

它不是让你靠感觉乱写代码，也不是让你替代研发，而是让你能够在产品早期阶段，借助 AI 完成：

需求澄清 → 原型实现 → 前后端联调 → 简单测试 → Demo验证 → 和研发高效沟通。

真正有竞争力的 AI 产品经理，不是只会写 PRD，也不是只会说大模型概念，而是能把一个产品想法快速推进到“可体验、可验证、可讨论”的状态。

一、先明确：vibe coding 不是“瞎写代码”

很多人对 vibe coding 有误解，以为就是：

“我不会代码，但我让 AI 帮我写一堆代码，然后能跑就行。”

但这其实是低阶的 vibe coding。

真正有价值的 vibe coding，不是凭感觉堆功能，而是要有一套产品开发意识。

也就是说，AI 产品经理要知道：

一个功能为什么要做？

用户路径怎么设计？

页面状态有哪些？

数据从哪里来？

接口怎么传？

异常情况怎么处理？

上线后怎么验证效果？

如果你只是让 AI 生成一个页面，这叫“会用工具”。

如果你能让 AI 帮你把一个功能从需求、交互、接口、数据、测试全部跑通，这才叫“具备 AI coding 能力”。

二、AI产品经理为什么需要 vibe coding 能力？

因为 AI 产品的研发节奏变了。

以前产品经理的核心交付物是 PRD、原型图、流程图，然后交给设计和研发实现。

但现在不一样了。

很多 AI 产品在早期阶段，最大的问题不是“怎么写得更完整”，而是：

这个想法到底能不能跑？用户到底能不能用？模型效果到底靠不靠谱？

所以 AI 产品经理不能只停留在文档层面。

你需要能快速做出一个 Demo，用它去验证：

用户是否理解这个功能；

AI 输出是否符合预期；

流程是否真的顺；

异常场景是否会崩；

数据结构是否支持后续扩展；

这个方案有没有必要投入研发资源。

这也是为什么现在越来越多团队会看重产品经理的 vibe coding 能力。

因为它代表的不是“你会不会写代码”，而是：

你能不能更快地把想法变成可验证的产品。

三、AI产品经理的 vibe coding 要达到什么程度？

我建议至少达到四个层级。

第一层：能用 AI 做出可交互原型

这是最基础的能力。

AI 产品经理至少要能用 AI coding 工具，把自己的产品想法做成一个可以点击、可以演示、可以体验的前端原型。

比如你要设计一个 AI 酒店搜索产品，不能只写：

“用户输入需求后，系统推荐酒店。”

你最好能直接做出一个简单 Demo：

用户输入：

“我想找一个适合亲子、离迪士尼近、有早餐的酒店。”

系统页面展示：

用户意图识别结果；

推荐酒店列表；

推荐理由；

筛选条件；

用户反馈按钮；

重新生成入口。

这个 Demo 不一定要有真实后端，但至少要让面试官或业务方看到：

你对产品流程是有完整理解的。

这一层的能力重点不是代码写得多好，而是你能不能把抽象需求变成一个具体体验。

第二层：能理解前后端基本逻辑

只会做静态页面还不够。

AI 产品经理还要理解基本的前后端关系。

至少你要知道：

前端负责什么；

后端负责什么；

接口是什么；

数据结构怎么传；

状态怎么变化；

接口失败怎么处理；

模型返回异常怎么兜底。

比如还是酒店搜索场景。

用户输入一句自然语言后，系统可能经历：

第一步，前端收集用户 Query；

第二步，后端调用大模型做意图识别；

第三步，系统提取目的地、预算、入住时间、偏好标签；

第四步，调用酒店检索接口；

第五步，对候选酒店进行排序；

第六步，前端展示推荐结果和推荐理由。

如果你不懂这些基础逻辑，你写出来的 PRD 很容易停留在表面。

但如果你懂，你就能在需求里讲清楚：

输入是什么；

输出是什么；

接口字段是什么；

异常场景怎么处理；

用户看到什么反馈；

系统什么时候走兜底逻辑。

这就是 AI 产品经理和传统产品经理很大的区别。

AI 产品不是只设计页面，它本质上是在设计一套“模型 + 数据 + 工程 + 交互”的完整链路。

第三层：能用 AI 快速搭建小型 Demo

更进一步，AI 产品经理最好能独立搭建一个小型 Demo。

这个 Demo 不一定要达到生产级别，但要能跑通核心链路。

比如你面试 AI 产品经理，可以准备一个小项目：

一个 RAG 知识库问答 Demo；

一个 AI 简历优化工具；

一个 AI 酒店推荐 Demo；

一个 AI 客服助手 Demo；

一个 AI 内容审核 Demo；

一个 AI Agent 任务执行 Demo。

你不需要把系统做得很复杂，但至少要能体现：

用户输入；

模型处理；

结构化输出；

结果展示；

异常兜底；

反馈收集。

比如做一个 AI 客服助手 Demo。

用户问：

“我预订的酒店能不能取消？”

系统不是简单回答一句话，而是会展示：

问题分类：订单售后；

用户意图：取消政策咨询；

需要字段：订单号、入住日期、酒店规则；

回答结果：根据规则说明是否可取消；

兜底提示：如果无法判断，引导用户联系人工客服。

这个 Demo 就能体现你不是只会讲概念，而是能把 AI 产品链路产品化。

第四层：能用工程化思维管理 AI coding

这是最重要的一层。

很多人用 AI 写代码，最大的问题是：

一开始很爽，后面全乱。

页面越改越崩；

代码结构越来越乱；

功能之间互相影响；

AI 改了旧功能又破坏新功能；

出了 bug 不知道从哪里查；

最后只能重写。

所以真正有竞争力的 vibe coding，不是“让 AI 随便写”，而是要有工程化意识。

这里就要提到一个很重要的能力：

SDD：Specification Driven Development，规范驱动开发。

也就是你不能上来就让 AI 写代码。

你要先让 AI 帮你澄清需求，再输出功能说明、页面结构、数据结构、接口设计、异常场景、测试用例，然后再进入开发。

换句话说：

不是先写代码，再修需求；而是先定义清楚，再让 AI 执行。

这才是 AI 产品经理真正应该掌握的 coding 工作范式。

四、具体来说，面试里可以怎么讲？

如果面试官问：

“你作为 AI 产品经理，需要具备什么样的 vibe coding 能力？”

你可以这样回答：

我理解的 vibe coding 不是让产品经理替代研发，而是让产品经理具备快速验证产品想法的能力。

AI 产品经理至少要做到三点。

第一，能用 AI coding 工具快速做出可交互原型，把抽象需求变成可以体验的 Demo，而不是只停留在 PRD 和原型图。

第二，能理解基本的前后端、接口、数据结构和模型调用逻辑，这样在设计 AI 产品时，能把用户输入、模型处理、系统返回、异常兜底和效果评估讲清楚。

第三，不能纯靠感觉写代码，而是要用规范驱动开发的方式，先澄清需求、定义数据结构、拆解功能模块、补充异常场景和测试用例，再让 AI 辅助开发。

所以对 AI 产品经理来说，vibe coding 的核心不是代码能力本身，而是产品验证能力、技术沟通能力和工程化协作能力。

五、AI 产品经理不需要达到什么程度？

这里也要讲清楚。

AI 产品经理不需要达到专业研发水平。

你不一定要精通算法；

不一定要能写复杂后端架构；

不一定要独立完成高并发系统；

不一定要负责线上代码质量；

不一定要替代前端、后端、算法工程师。

但你不能完全不懂技术。

你至少要知道一个 AI 功能从用户输入到结果输出，中间经历了哪些环节。

比如：

用户输入自然语言；

系统进行意图识别；

调用知识库或业务数据库；

进行向量检索或关键词检索；

对结果进行重排；

把结果交给大模型生成回答；

前端展示结果；

用户进行反馈；

系统根据反馈继续优化。

这些你不一定都要亲自开发，但你必须能讲清楚。

因为 AI 产品经理的核心价值，就是把业务问题拆成可被模型、数据和工程解决的问题。

六、真正有竞争力的 vibe coding，是这五种能力

1. 需求拆解能力

你不能只跟 AI 说：

“帮我做一个 AI 酒店推荐工具。”

你要能拆成：

用户是谁；

核心场景是什么；

输入内容是什么；

输出结果是什么；

页面有哪些模块；

推荐逻辑是什么；

异常情况有哪些；

成功标准是什么。

AI coding 的效果，很大程度取决于你能不能把需求说清楚。

产品经理不是输一句话让 AI 自由发挥，而是要像导演一样控制 AI 的执行方向。

2. 原型实现能力

你要能快速做出一个前端页面，哪怕只是 Demo。

比如：

搜索页；

结果页；

详情页；

弹窗；

加载状态；

空状态；

错误状态；

用户反馈入口。

这些东西不需要你手写每一行代码，但你要能用 AI 快速生成、调整和迭代。

面试里如果你能直接展示一个自己做过的 Demo，竞争力会比只讲概念强很多。

3. 技术理解能力

你要知道基本的技术概念。

比如：

什么是 API；

什么是 JSON；

什么是前后端联调；

什么是数据库；

什么是向量检索；

什么是 prompt；

什么是模型调用；

什么是流式输出；

什么是接口超时；

什么是兜底策略。

这些不需要你达到研发深度，但你要能和技术同学对齐语言。

否则你做 AI 产品，很容易只会说“希望模型更智能一点”，但说不清楚到底该怎么优化。

4. 测试验证能力

AI 产品尤其需要测试。

因为 AI 的输出不是完全稳定的。

所以你做 vibe coding 的时候，不能只看功能能不能跑，还要看：

不同输入下结果是否稳定；

模型有没有胡说；

边界场景是否能处理；
接口失败有没有提示；
用户误输入有没有兜底；
敏感内容有没有拦截；
生成结果是否可解释；
用户反馈是否被记录。

一个成熟的 AI 产品经理，不能只追求“Demo 看起来很酷”，还要追求“链路可验证、效果可评估、问题可定位”。

5. 和研发协作的能力

vibe coding 最终不是为了自己单干，而是为了提升协作效率。

当你能自己做出 Demo，你和研发沟通时就不再只是说：

“我想要一个差不多这样的功能。”

而是可以说：

“这是我用 AI 做的原型，核心链路已经跑通了。这里是页面结构，这里是字段定义，这里是用户输入和模型输出，这里是异常状态。你们正式开发时可以基于这个方案做工程化实现。”

这会极大降低沟通成本。

研发也更容易判断：

这个需求有没有价值；

实现难度大不大；

接口怎么设计；

哪些逻辑要后端做；

哪些展示由前端处理；

哪些部分需要算法或模型支持。

所以 vibe coding 的本质，不是替代研发，而是让产品经理更懂研发、更能推进研发。

七、一个具体例子：AI 酒店搜索产品

假设你要做一个 AI 酒店搜索功能。

低阶产品经理可能会写：

用户输入自然语言需求后，系统推荐合适酒店。

这个描述太粗了。

具备 vibe coding 能力的 AI 产品经理，会拆成：

用户输入：

“帮我找一个五一去杭州住的酒店，预算600以内，适合情侣，最好离西湖近一点。”

系统处理：

识别目的地：杭州；

识别时间：五一；

识别预算：600以内；

识别人群：情侣；

识别偏好：靠近西湖；

识别需求类型：酒店推荐。

系统输出：

推荐酒店列表；

每家酒店的推荐理由；

距离西湖的距离；

价格区间；

适合情侣的原因；

可订状态；

用户可继续追问。

异常场景：

用户没说时间，系统追问入住日期；

用户预算过低，系统提示调整条件；

酒店无房，系统推荐相似替代；

模型无法判断时，展示常规筛选结果；

接口失败时，提示稍后重试。

效果指标：

搜索转化率；

推荐点击率；

用户追问率；

筛选使用率；

无结果率；

人工客服转接率；

用户满意度反馈。

如果你还能把这个逻辑用 AI coding 做成一个可演示 Demo，那你的面试表达就会非常有说服力。

因为你展示的不是“我懂 AI”，而是：

我能把 AI 能力真正转化成产品功能。

八、AI 产品经理 vibe coding 的能力标准

可以用一个简单标准来判断。

初级水平

能用 AI 生成页面；

能改简单样式；

能做静态原型；

能展示基本功能流程。

这个水平适合入门，但竞争力有限。

中级水平

能拆解完整需求；

能生成可交互 Demo；

能理解数据结构；

能模拟接口返回；

能设计异常状态；

能做简单测试。

这个水平已经可以支撑 AI 产品经理的日常工作。

高级水平

能独立搭建小型 AI 应用；

能接入模型 API；

能设计 prompt 和结构化输出；

能跑通前后端链路；

能设计评估指标；

能把 Demo 转化成正式研发需求。

这个水平在面试里会非常加分。

更高级水平

能用工程化方式管理 AI coding；

能做模块化拆分；

能设计可扩展架构；

能管理版本迭代；

能建立测试用例；

能判断哪些能力适合自研，哪些适合外采，哪些适合平台化沉淀。

这个水平就不只是“会用 AI 写代码”，而是已经具备 AI 产品负责人视角。

九、面试中最容易踩的坑

第一个坑：把 vibe coding 说成“我会写代码”

这会让面试官觉得你没有抓住重点。

AI 产品经理不是去和研发比代码能力，而是要强调：

我能用 coding 能力提升产品验证效率和研发沟通效率。

第二个坑：只展示页面，不讲产品逻辑

很多人做了一个好看的 Demo，但讲不清楚背后的业务流程、数据结构和评估指标。

这也不够。

AI 产品经理一定要把 Demo 和业务目标连接起来。

第三个坑：只讲工具，不讲方法论

比如只说自己会 Cursor、Lovable、Bolt、Claude Code，这些都只是工具。

面试官真正关心的是：

你用这些工具解决了什么问题？

你怎么拆需求？

你怎么验证功能？

你怎么和研发协作？

你怎么保证结果可靠？

工具会变，但方法论更重要。

第四个坑：把 AI coding 当成捷径

AI coding 不是让你省掉思考，而是让你把思考更快落地。

你越不懂产品逻辑，AI 越容易带偏你。

你越懂需求、交互、数据和工程边界，AI 越能帮你放大效率。