

AI产品经理如何写好一个优秀的 Skills?

 优秀的 Skills，不是把提示词写得更长，而是把一个业务任务，拆成 AI 能稳定理解、稳定执行、稳定交付的能力模块。

很多人一写 Skills，就会陷入一个误区：

把它当成“高级提示词模板”。

于是里面全是：

你是一个专业的产品经理；

请你认真分析；

请你结构化输出；

请你不要胡编乱造……

这些话不是不能写，但如果一个 Skills 只有这些，那它本质上还是一个 Prompt，不是一个真正可复用的 AI 能力。

真正优秀的 Skills，核心不是“让 AI 看起来很聪明”，而是让 AI 在特定场景下，知道要做什么、怎么做、做到什么程度、什么情况不能乱做。

一、先明确：Skills 解决的不是“聊天问题”，而是“任务问题”

AI 产品经理写 Skills 的第一步，不是上来写指令，而是先判断这个 Skills 到底要解决什么任务。

很多人会写：

写一个产品分析 Skills

写一个竞品分析 Skills

写一个用户调研 Skills

这类描述太大了，AI 很难稳定执行。

更好的写法应该是：

帮助 AI 产品经理快速拆解一个 AI 产品的竞品分析框架，并输出适合面试表达的结构化答案。

帮助产品经理基于用户反馈，归纳核心问题、需求类型、优先级和后续产品动作。

帮助 AI 产品经理将一个业务场景拆解成 RAG / Agent / 工作流 / 模型能力的产品方案。

你会发现，优秀的 Skills 一定不是泛泛而谈，而是有明确任务边界。

它要回答三个问题：

第一，这个 Skills 服务谁？

第二，它解决什么具体任务？

第三，它最终要交付什么结果？

如果这三个问题没想清楚，后面写再多提示词都没用。

二、优秀的 Skills，要有清晰的输入和输出

很多 Skills 不好用，核心原因是输入输出不清楚。

用户随便丢一段信息进来，AI 不知道该提炼什么；

AI 输出一大堆内容，用户也不知道能不能直接用。

所以写 Skills 时，一定要先定义清楚输入和输出。

比如你写一个“AI 产品方案分析 Skills”，输入可以包括：

用户场景、业务目标、现有问题、目标用户、已有功能、数据来源、技术约束。

输出可以包括：

需求背景、用户痛点、产品目标、核心功能、AI 能力拆解、流程设计、异常场景、指标体系、迭代规划。

这就叫输入输出契约。

不是让 AI 自由发挥，而是告诉它：

你拿到什么信息；

你要补充判断什么信息；

你最终必须按照什么结构交付。

一个优秀的 Skills，必须让结果稳定。

不是这次输出像 PRD，下次输出像公众号文章，再下次输出像聊天建议。

而是只要输入类似，输出结构就应该保持相对一致。

三、不要只写“要求”，要写“执行步骤”

很多人写 Skills 的方式是：

请你专业。

请你深入。

请你结构化。

请你不要遗漏。

请你结合业务场景。

这类写法最大的问题是：它只告诉 AI “结果要好”，但没告诉 AI “过程怎么做”。

优秀的 Skills，一定要把任务拆成执行步骤。

比如写一个“竞品分析 Skills”，不要只写：

请你分析竞品的核心优势和不足。

而要写成：

第一步，先判断竞品所属赛道和目标用户。

第二步，拆解竞品的核心使用路径。

第三步，对比功能、体验、数据、AI 能力和商业模式。

第四步，总结对我方产品的启发。

第五步，输出可以用于面试表达的结构化结论。

这时候 AI 才不会乱飘。

因为你不是在让它“自由分析”，而是在给它一套可执行的方法论。

对 AI 产品经理来说，写 Skills 的本质其实就是写一套轻量级 SOP。

四、Skills 里最重要的不是“聪明”，而是“边界”

一个 Skills 写得好不好，关键不只是看它能不能输出内容，还要看它会不会乱输出。

AI 最大的问题不是不会说，而是太会说。

信息不够的时候，它也能讲得头头是道；

边界不清的时候，它也能自动脑补；

需求模糊的时候，它也能假装自己理解了。

所以优秀的 Skills 一定要写清楚边界。

比如：

当用户没有提供业务背景时，要先基于常见场景做合理假设，并明确标注假设。

当涉及具体数据时，不得凭空编造，只能使用用户提供的数据或说明需要补充。

当用户的问题过于宽泛时，要先收敛成具体任务。

当无法判断时，要输出“不确定点”和“需要补充的信息”。

当涉及方案设计时，要区分“当前可落地方案”和“未来迭代方向”。

这部分非常关键。

因为 Skills 的价值，不是让 AI 永远给答案，而是让 AI 知道什么时候该给答案，什么时候该降级，什么时候该提醒风险。

这才是产品经理真正应该关注的地方。

五、优秀的 Skills，要内置判断标准

很多 Skills 最大的问题是，只定义了“怎么输出”，没有定义“什么叫输出得好”。

比如你让 AI 写产品方案，它确实能写。

但好不好？完整不完整？能不能用于面试？能不能用于真实项目？没有标准。

所以一个好的 Skills，最好要内置评估标准。

比如一个 AI 产品方案类 Skills，可以要求输出必须满足：

是否有明确业务目标；

是否讲清用户痛点；

是否拆解 AI 能力边界；

是否包含流程设计；

是否覆盖异常场景；

是否包含指标体系；

是否能落到产品动作；

是否有短中长期迭代路径。

这样 AI 在生成内容时，就不是随便写，而是会围绕这些标准自我校验。

这就是很多普通 Prompt 和优秀 Skills 的差距。

普通 Prompt 是“帮我写一个东西”。

优秀 Skills 是“按照一套标准，帮我完成一个高质量任务”。

六、写 Skills 时，要把经验沉淀成可复用模块

AI 产品经理写 Skills，最有价值的地方，不是一次性生成一个结果，而是把自己的方法论沉淀下来。

比如你经常做 AI 产品分析，那就可以沉淀成：

用户场景拆解模块；

AI 能力判断模块；

RAG / Agent / Workflow 方案选择模块；

指标体系设计模块；

异常场景分析模块；

面试表达优化模块。

以后你遇到不同问题，只要复用这些模块，就能快速生成高质量内容。

这其实就是 AI 产品经理和普通使用者的区别。

普通人使用 AI，是临时提问。

AI 产品经理使用 AI，是沉淀能力。

你不是每次从零开始问 AI：

“帮我分析一下这个产品。”

而是逐渐形成一套自己的 Skills 库：

竞品分析 Skills；

PRD 生成 Skills；

AI Agent 评估 Skills；

RAG 方案拆解 Skills；

用户反馈归因 Skills；

面试回答优化 Skills。

这套东西积累多了，才是真正属于你的 AI 工作流资产。

七、一个优秀 Skills 的基本结构

如果要把它整理成一个标准模板，可以这样写：

1. Skills 定位

这个 Skills 是用来解决什么任务的？

服务什么用户？

适合什么场景？

2. 输入信息

用户需要提供哪些信息？

如果信息不完整，AI 应该如何处理？

哪些信息必须明确，哪些信息可以合理假设？

3. 执行流程

AI 应该先分析什么，再判断什么，最后输出什么？

中间是否需要分类、归因、对比、校验？

4. 输出结构

最终结果用什么格式输出？

是文章、表格、PRD、面试答案、汇报材料，还是方案文档？

5. 规则与边界

哪些内容不能编？

哪些地方必须提示风险？

哪些情况需要降级或补充说明？

6. 质量标准

什么样的结果算合格？

什么样的结果算优秀？

是否满足业务目标、逻辑完整性、可执行性和表达清晰度？

7. 迭代方式

用户反馈后如何优化？

是否保留原结构？

是否只改表达、不改逻辑？

是否支持不同风格版本？

这套结构看起来很简单，但已经能覆盖大部分 Skills 的核心设计。
