

2026 面试核对版

AI Engineer 面试宝典

RAG 系统设计、Agent 边界、Eval 排障、Production、Behavioral。别背题，把项目讲成能被追问的工程故事。

RAG

Agents

Evals

扫码找 Rain



目录

- 00 这本书怎么用
- 01 2026 的 AI Engineer 面试考什么
- 02 答题框架
- 03 RAG 与幻觉排查
- 04 Structured Output / Function Calling / Agent
- 05 Eval、成本、延迟
- 06 Security、Coding、ML 基础
- 07 简历项目怎么写
- 08 Behavioral
- 09 7 天冲刺清单
- 10 官网来源清单

先读第 1、2 章，再按自己的薄弱项刷题。没有项目数字的人，优先看第 7 章。

这本书怎么用

先说清楚：这不是“包过题库”。AI Engineer 面试变化太快，背题只会让你看起来像没做过项目。

这本真正帮你准备的是面试官在追什么：你会不会写代码、能不能设计 RAG、懂不懂 tool calling 和 agent 边界、有没有 eval、上线以后怎么控成本和查问题。每道题后面都给答题骨架，方便你拿自己的项目去套。

8

类高频面试场景

14

道核心题 + 答题骨架

5

层排障方法



扫码了解 AI Engineer 面试准备资料

把你的项目、简历和目标岗位发来，先看差的是 RAG、Agent、eval，还是项目表达。





2026 的 AI Engineer 面试，到底在考什么

AI Engineer 不是“会调 API 的后端”，也不是“会说 prompt 的产品”。面试会把你往工程细节里拽：RAG 准不准、工具能不能安全执行、评估怎么做、线上挂了怎么排。

01

Coding 基本盘

不是让你刷到天荒地老。AI Engineer 仍然要能写稳代码：字符串、数组、hash map、API wrapper、并发、错误处理。

可能这么问：写一个 rerank 前后的 top-k 合并函数；给一个 tool call 日志，统计失败率和 P95 latency。

02

LLM 应用设计

考你能不能把 prompt、schema、tool、memory、context 管起来。

可能这么问：设计一个客服 Copilot：输入是什么，输出结构是什么，失败怎么兜底。

03

RAG 系统设计

高频。考切片、embedding、hybrid search、rerank、权限隔离、引用、eval。

可能这么问：给法律 SaaS 做 RAG：50GB 文档、5000 QPD、多租户隔离。

04

Tool / Agent

考你是否知道模型不直接执行工具，工具执行权在应用代码。

可能这么问：设计一个能查库存、下单、退款的 agent，哪些工具必须加权限和人工确认。

05

Eval 与排障

AI 面试最能拉开差距的地方。考你有没有数字和回归测试。

可能这么问：用户说 AI 胡说，怎么定位是 retrieval、prompt、model、schema 还是数据问题。

06

Production

延迟、成本、监控、安全、灰度。

可能这么问：老板说 AI 月账单翻倍，怎么拆账、降本、保质量。

07

项目深挖

面试官最爱从你简历里追问。你写了 RAG，就会问数据哪来、怎么评、错例是什么。

可能这么问：讲一个项目：从问题、架构、metric、失败案例到上线结果。

08

Behavioral

外企很看重。不是背故事，是证明你能处理 trade-off、冲突、模糊需求。

可能这么问：说一次你怎么说服 PM 不上 autonomous agent，而先做 workflow。

判断自己准备好了没：随便拿你简历上的一个 AI 项目，能不能讲出数据来源、架构图、三个失败案例、一个评估集、一次降本或提速。如果讲不出来，先别急着刷更多题。



答题框架：面试里别一上来堆名词

很多候选人一听 RAG，就开始报 LangChain、pgvector、rerank。面试官真正想看的是：你能不能先澄清需求，再把取舍说清楚。

FRAME 1

系统设计四步

澄清目标 → 画数据流 → 讲 trade-off → 讲 eval 和上线。不要一上来堆技术名词。

FRAME 2

排障五层

输入/权限 → 检索 → Prompt/Schema → Model/Tool → 监控/回归。按层查，不要凭感觉改 prompt。

FRAME 3

项目讲法

问题有多具体 → 你做了什么 → 用什么 metric 证明 → 失败在哪里 → 下一步怎么改。

FRAME 4

Behavioral

STAR 够用，但每个故事都要带数字、取舍、个人动作。没有数字，就像没发生。

系统设计回答顺序：

1. 目标：用户是谁，成功标准是什么
2. 数据：从哪里来，怎么更新，权限怎么做
3. 架构：indexing / retrieval / generation / eval
4. 取舍：准确率、延迟、成本、安全
5. 上线：trace、canary、feedback、回归测试



RAG 与幻觉排查：最容易拉开差距

LangChain 官方 RAG 教程把流程拆成 indexing 和 retrieval/generation；LlamaIndex 也把 RAG 定义为把私有数据在 query time 提供给 LLM。面试里别只说“接了向量库”，要说清这两段怎么分工。

RAG 设计

01

给法律 SaaS 设计一个 RAG，日均 5000 查询、50GB 文档、多个客户互相隔离。你怎么做？

1. 先问约束：文档类型、更新频率、权限模型、准确率目标。
2. 方案：ingestion pipeline、chunk by legal structure、metadata filter、hybrid search + rerank、source citation。
3. 评估：50–200 个真实 query，分 factuality、citation、permission leak 三类打分。
4. 上线：canary、trace、feedback、错例回灌。

幻觉排查

02

用户说 AI 答错事实，你怎么查？

1. 先复现 query 和 trace。
2. 看 retrieved context 是否含正确答案；没有就是 retrieval 问题。
3. context 有答案但模型没用，查 prompt、context ordering、answer instruction。
4. 最后加 eval，防止同类错下次回归。

STRUCTURED OUTPUT

03

为什么不只让模型“输出 JSON”？

1. JSON mode 只保证格式更像 JSON，不等于满足 schema。
2. Structured Outputs / schema 可以约束 required keys、enum、类型。
3. 面试答案要提 validation、retry、refusal path、版本兼容。



Structured Output、Function Calling、Agent 边界

这块最怕背概念。OpenAI、Anthropic、Gemini 文档都讲得很明确：模型可以请求工具调用，但真正执行工具的是你的应用或平台。安全边界就在这里。

FUNCTION CALLING

04

function calling 里模型会不会真的执行函数？

1. 不会。模型产生 tool/function call 和参数。
2. 你的应用代码执行工具，再把 tool result 交回模型。
3. 所以权限、幂等、审计、错误处理都在应用侧。

AGENT 边界

05

什么时候用 autonomous agent，什么时候用 workflow？

1. 路径能穷举，优先 workflow。
2. 需要动态选择工具但任务单域，单 agent。
3. 跨域且可拆子任务，再考虑 multi-agent。
4. 面试要主动讲成本、可控性、观测性。

EVAL 设计

06

怎么给客服 bot 建 eval？

1. 先定义 success criteria：准确、引用、语气、隐私、延迟、成本。
2. 离线数据集：真实问题、边界问题、对抗问题。
3. grader：能用 code 就不用 LLM；复杂主观用 rubric + LLM-as-judge + 抽样人工。
4. 线上：trace、user feedback、负反馈进回归集。

一句话记住：tool call 是模型的“请求”，不是业务系统的“授权”。退款、删数据、发邮件这类动作，必须由你的代码做权限、幂等、审计和人工确认。



Eval、成本、延迟：面试官最爱追问的生产问题

Anthropic 文档说得很直：做 LLM 应用要先定义 success criteria，再做 evaluation。LangSmith 也把 offline evaluation、online evaluation、dataset、run trace 分开讲。你能把这些说顺，面试官会觉得你真上过线。

成本优化

07

AI 月账单从 \$2k 涨到 \$8k，怎么查？

1. 按 endpoint、model、prompt version、用户、tool 调用拆账。
2. 看 token 输入、输出、重复 query、失败重试、agent 步数。
3. 措施：prompt caching、batch、small model routing、max tokens、retrieval 精简。
4. 降本必须重新跑 eval，不能只看账单。

LATENCY

08

RAG 问答 P95 从 3 秒涨到 12 秒，怎么排？

1. 分段计时：retrieval、rerank、LLM、tool、network。
2. 查 top-k、rerank 模型、context size、并发、重试。
3. 临时止血：减少 k、并行检索、缓存、streaming。
4. 长期：trace dashboard 和 SLO。

PROMPT 注入

09

网页内容里写了“忽略前面指令”，RAG 系统怎么防？

1. retrieved context 是 data，不是 instruction。
2. system prompt 明确忽略文档里的指令。
3. 工具权限最小化，敏感操作二次确认。
4. 加入 adversarial eval。

如果你的项目没有 eval 和 trace，面试容易被追到卡住。扫 Rain 的码，把项目 README 发来，先看哪里能补数字。

Rain



Senior IT Career Consultant

在澳洲近20年，Master of HRM，15年企业HR及猎头招聘工作经验，熟悉中国及澳洲就业市场情况。尤其在IT领域中拥有超过6年的咨询及招聘经验。

—— 是你拿下offer的最佳辅助！

毕业生求职做规划

辅助在职人士跳槽晋升

已帮助数千余人成功就业

免费简历诊断

3000+

3000+咨询案例

1200+

1200+辅助成功
斩获offer

1000+

1000+简历诊断

 <http://linkedin.com/in/rainqiu>





Security、Coding、ML 基础：别让基本盘拖后腿

AI 面试不是只聊 AI。你要能写函数、看日志、懂权限。尤其是 agent 类项目，安全问题比 demo 更重要。

项目深挖

10

你简历写“RAG 命中率提升 20%”，怎么算的？

1. 定义 denominator 和 metric。
2. 说明 baseline、数据集、改动点、统计口径。
3. 展示错例：哪些问题仍然错。
4. 不要只说“感觉更准了”。

CODING

11

写一个函数，合并 vector search 和 keyword search 的结果，按综合分排序并去重。

1. 考 hash map、排序、权重、边界条件。
2. 加分：保留 source、score breakdown、tie-breaker。
3. 别只写伪代码，能说复杂度。

ML 基础

12

为什么 embedding 相似不等于答案正确？

1. 相似只说明向量空间接近。
2. 可能 chunk 不完整、上下文冲突、权限错、语义漂移。
3. RAG 需要 retrieval metric + answer quality metric 分开评。

SECURITY

13

tool 能退款、删数据、发邮件，怎么做安全边界？

1. 分 read/write/dangerous tools。
2. dangerous tool 要权限、幂等 key、审批、人类确认、审计日志。
3. 模型只建议，不直接绕过 policy。



简历项目怎么写，才像真的干过

面试能不能聊起来，一半取决于简历怎么写。别写“用了 LangChain”，写你解决了什么问题，用什么指标证明，失败过什么。

别这么写

- Built an AI chatbot using LangChain and OpenAI.
- Used RAG to answer questions from documents.
- Improved prompts and reduced hallucinations.

改成这样

- Designed an internal knowledge-base RAG over 1,200 policy docs; added metadata filters and rerank; improved citation accuracy from 71% to 89% on 80 hand-labeled queries.
- Built tool-calling support for refund / order lookup flows with idempotency keys, audit logs and human confirmation for write actions; reduced manual lookup time by 38% in pilot.
- Set up 120-case eval suite for FAQ assistant; gated prompt changes in CI and tracked P95 latency / cost per successful answer in LangSmith traces.

三种项目别放主位：克隆 ChatGPT、只有 prompt 没 eval、拿模板跑通但自己说不清数据和错误案例。项目可以小，但必须能被追问。



Behavioral: 用 STAR 讲 trade-off, 不讲鸡汤

AI Engineer 的 behavioral 不只是“团队合作”。更常见的是：你怎么处理模糊需求、怎么拒绝不安全方案、怎么说服别人做 eval、怎么在成本和质量之间取舍。

BEHAVIORAL

14

讲一次你为了质量牺牲速度的经历。

1. STAR: 场景、任务、动作、结果。
2. 必须有数字和 trade-off。
3. 把“我们”改成“我做了什么”。

STAR 准备 4 个故事就够用

- ☐ 一次质量 vs 速度的取舍 **必备**
比如先上 workflow, 不上 autonomous agent。
- ☐ 一次生产事故或错例复盘 **必备**
要讲你怎么定位, 不是只讲“我们修好了”。
- ☐ 一次和 PM / 法务 / 安全的沟通 **加分**
AI 项目很多边界不是代码能单独决定。
- ☐ 一次主动学习新工具并落地 **加分**
别说“我学得快”, 说你用了几天、做出什么、结果如何。



7 天冲刺清单

临近面试别再开新课。把你现有项目整理到能被追问，收益更高。

7 DAYS 每天只做一件硬事

- ☐ **Day 1: 画一张项目架构图** 必做
输入、检索、模型、工具、输出、监控都画出来。
- ☐ **Day 2: 写 20 个 eval query** 必做
正常、边界、攻击、权限、空答案都要有。
- ☐ **Day 3: 准备 3 个失败案例** 必做
答错、慢、贵、工具误触发都可以。
- ☐ **Day 4: 把简历项目改成数字版** 必做
准确率、延迟、成本、用户数、文档量，至少补两个。
- ☐ **Day 5: 模拟一次 RAG 系统设计** 练习
录音，听自己有没有跳步骤。
- ☐ **Day 6: 准备 4 个 STAR 故事** 练习
每个故事写出数字和个人动作。
- ☐ **Day 7: 准备反问问题** 收尾
问 eval、trace、cost ownership、tool safety，不要只问技术栈。



想做一次 AI Engineer mock interview?

扫码找 Rain。可以先从你的简历项目开始，按 RAG、Agent、Eval、Production 四块追问。



官网来源清单

核对日期：2026-06-30。这本只把官方文档、官方教程、公开研究和自家已核内容作为依据。题目是按这些真实能力面向整理，不声称来自某家公司内部题库。

- [OpenAI Prompt engineering](#)
- [OpenAI Function calling](#)
- [OpenAI Structured Outputs](#)
- [OpenAI Evals](#)
- [OpenAI Deployment checklist](#)
- [Anthropic Prompt engineering overview](#)
- [Anthropic Define success criteria and build evaluations](#)
- [Anthropic Tool use with Claude](#)
- [Google Gemini Function calling](#)
- [LangChain RAG tutorial](#)
- [LangSmith Evaluation concepts](#)
- [LlamaIndex high-level concepts](#)
- [Business Insider on LinkedIn Jobs on the Rise 2026](#)
- [AI Skills Improve Job Prospects](#)



别背题。把项目讲扎实。

AI Engineer 面试会追 RAG、Agent、Eval、成本、延迟和安全。卡住了，扫 Rain 的码，把你的简历项目发来，先看该补哪一块。

Rain

Senior IT Career Consultant

在澳洲近20年，Master of HRM，15年企业HR及猎头招聘工作经验，熟悉中国及澳洲就业市场情况。尤其在IT领域中拥有超过6年的咨询及招聘经验。

是你拿下offer的最佳辅助！

毕业生求职规划

辅助在职人士跳槽晋升

已帮助数千余人成功就业

免费简历诊断

3000+

3000+咨询案例

1200+

1200+辅助成功
斩获offer

1000+

1000+简历诊断

<http://linkedin.com/in/rainqiu>



扫码了解 AI Engineer 面试准备

简历项目诊断、mock interview、RAG / Agent / Eval 补强路线。