

AI 很强，然后呢？

当智商不再稀缺，管理成了稀缺

竹小竹8167

费曼式写法 · 由多个 AI 代理撰写与互相审校

导读

当智商不再稀缺，管理成了稀缺

先别急着等更强的模型

这两年，几乎每个用 AI 的人都说过同一句话：「等下一代模型出来就好了。」仿佛眼下产出的不如意，全是引擎不够强。

这本书想请你先做一个相反的假设：**假如 AI 的智商从今天起永远冻结，你的效率还能涨多少？**

答案可能会让你意外——不是「到头了」，而是「空间还很大」。因为看看你身边的人（包括你自己）是怎么用 AI 的：打开聊天框，扔一句话进去，把答案复制出来，完事。这就像拿到了一辆 F1 的钥匙，却只用它去楼下买菜，还抱怨它后备箱太小。

你的产出由两样东西相乘决定：AI 的智商，和你管理 AI 的能力。前者这几年涨了十倍，那是实验室的功劳，与你无关；后者，大多数人还停留在出厂设置。而后者才是你能拧的那一半——而且它不随厂商发布会过期，AI 越强，它越值钱。

这本书把「管理 AI」拆成一屋子具体的旋钮：怎么给它入职培训般的上下文、怎么把任务切到能验收、怎么搭质检线、怎么校准信任。以及一个被整个行业忽略的角落：现在大家忙着管理「和 AI 的会话」，但真正值钱的是管理「AI 交出的结果」——它们怎么呈现、怎么比较、怎么归档、怎么追溯。

读法上不用客气：章节各自独立，从哪章进来都行。但如果只读一章，读第七章。

开始吧。从「智商不是瓶颈」开始。

第一章 · 智商不是瓶颈

一个思想实验

想象明天早上醒来，你看到一条新闻：所有 AI 实验室联合宣布，模型的智商**从此冻结**，永远不再提升。今天你手里的这个模型，就是你这辈子能用到的最聪明的 AI。

问一个不太舒服的问题：在这种情况下，你的工作效率还有多大的提升空间？

大多数人的直觉是「没了，AI 不进步了嘛」。但如果你认真观察过身边的人怎么用 AI，你会得出一个反直觉的结论：**提升空间不是没有了，而是大得吓人**。因为绝大多数人对 AI 的使用，离这台机器的能力上限还差着十万八千里。

产出是一个二元函数

把「你用 AI 做出的东西」写成一个大白话公式：

大白话

你的产出 $\approx$ AI 的智商 $\times$ 你管理 AI 的能力

前一项是实验室的事，你说了不算。后一项是你的事，天天都在你手里。

问题是，这两项的增速完全不成比例。过去几年，AI 的智商像坐了火箭——从「会胡说八道的鹦鹉」到「能通过律师考试、能写能跑能调的程序」，也就两三年。而「怎么管好 AI」这件事呢？大多数人还停留在「打开聊天框，把问题原样扔进去，把答案原样复制出来」。这不是管理，这是转发。

于是出现了一个奇怪的格局：**我们管理 AI 的能力，远远落后于 AI 智商的提升**。就算智商这一项从此不再涨，光是把管理这一项从 10 分提到 80 分，产出还能翻好几倍。反过来说，如果智商不涨了，管理这一侧的提升空间不是变小了，而是更显眼了——因为那是唯一还在你手里的变量。

智商从稀缺变成了大宗商品

这里有一个更深的变化值得看清。在人类历史上，「聪明」一直是稀缺资源。一个聪明人便宜地供你差遣，这种事从来没有发生过——你雇一个聪明的员工，要付出高薪、股权、还要担心他跳槽。

AI 把这个延续了几千年的格局打破了。现在，一个在很多科目上超过普通人的「脑子」，按次计费，每次几分钱，二十四小时在线，不闹情绪，不辞职。智商——解决问题的能力本身——第一次变成了像电和自来水一样的大宗商品。

当一样东西从稀缺变成大宗商品，瓶颈一定会挪到别的地方。电刚普及的时候，谁家有电不是竞争优势，**谁会用电器**才是。今天同理：你能调用的 AI 智商和你隔壁同事能调用的，是同一个模型、同一个价格。你们的差距，只可能来自别处——来自你怎么用它。

大白话

大白话：脑子现在论斤卖了，值钱的是会用脑子的人。

这本书要回答的问题

所以这本书从头到尾只追一个问题：

AI 越来越聪明，为什么你的产出没有同步变好？差的那一块，到底差在哪？

答案的一句话版本是：差在「管理」这一侧。但「管理 AI」不是一句口号，它是一屋子可以动手拧的旋钮——每一个具体的、可调的动作：给多少背景信息、任务切成多大块、怎么验收结果、什么时候信什么时候查。每一个旋钮拧对了，效率都是成倍地变。

接下来的章节，我们就一个旋钮一个旋钮地拧过去。在拧之前，下一章先看一个每天都在发生的现象：同一台「车」，不同的人开，为什么速度差出好几倍。

第二章 · 给新手一辆 F1

同一个模型，两种人生

先讲两个真实感很强的场景。

小张让 AI 帮他写一份行业分析。他打开聊天框，输入「帮我写一份新能源汽车行业分析」，拿到一份四平八稳的八股文，改了两小时，最后还不如自己写。他的结论：AI 也就那样。

小李做同样的事。她先花十分钟把自己知道的东西倒给 AI：报告给谁看、他们关心什么、上次的报告哪里被老板打回来了、她手上有哪几份数据、她猜结论可能是什么但不确定。然后她不要「一份报告」，先要五个不同的分析框架，挑了一个，让它先出提纲，再分段写，每段写完她挑刺，挑完再改。最后她花四十分钟拿到了一份能直接交的东西。

同一个模型，同一道题。差的那几倍时间，差在哪？

车是一样的，挡位不一样

把 AI 想成一辆 F1 赛车。它的引擎能跑到三百公里时速，这是厂商的事。但赛车交到一个从没摸过方向盘的人手里，他大概率会：挂一档，踩住油门不敢松，转弯踩刹车踩到死，然后抱怨「这车也没比我的家用车快多少」。

他没说错——**在他的开法下**，F1 确实不快，甚至更慢，因为他还得为它的敏感和脾气付出代价。

小张和小李的差距就是这样。车是一样的车。区别是小李知道这台车有哪些旋钮——那些可以动手调节、每个都会显著改变结果的开关——而小张只知道油门和刹车。

大白话

大白话：AI 不好用，先别怪车，先看看自己挂了几挡。

差距不是线性的，是相乘的

一个容易低估的事实是：这些旋钮的效果是**相乘**的，不是相加的。

给足背景信息，结果好一倍；任务切对大小，再好一倍；要求它先给方案再动手，避免一次返工，又省一半时间；会验收，少错一次漏一次。每一项单独看都不算神奇，但四五项叠在一起，就是数量级的差距。这就是为什么「AI 到底有没有用」这个问题永远没有共识——提问者们开的根本不是同一辆车。

这也是为什么争论「现在的 AI 够不够聪明」经常是错的焦点。对大多数人来说，**手边这台车的马力远远没有用完**，等更强的引擎，不如先学换挡。

好，那旋钮都在哪？

看到这儿，合理的反应是：行，那你说说，这屋子的旋钮到底有哪些？

下一章我们干一件有点笨但很有用的事：把「管理 AI」这件听起来很虚的事，拆成一张具体的旋钮清单。你会发现它们没有一个高深莫测——难的不是知道它们存在，而是意识到它们**每一个都值得你花心思去调**。

第三章 · 一屋子旋钮

先把地图画出来

这一章做一件看似笨拙的事：把「管理 AI」拆成一张清单。不是操作规程——后面每章会挑重要的细讲——而是一张地图。有了地图你才知道自己在哪、还有多少地方没去过。

清单有个排序原则：**按大多数人拧得最差、而拧好收益最大的顺序排。**

旋钮一：上下文——你给它看了多少

AI 对你的世界一无所知，除非你告诉它。你给的背景、约束、例子、忌讳，统称为上下文——就是 AI 开工前能看到的那叠材料。

这是杠杆最大的旋钮，也是被拧得最差的。大多数人给 AI 的信息量，相当于对新同事只说了一句「把那事儿办一下」。第四章专门讲它。

旋钮二：任务切分——一块切多大

「帮我写本书」和「帮我写第三章第二节的初稿，五百字，讲清楚一个类比」是同一家店的两种点法。前者大概率端出一盘糊的，后者才可能精确命中。

任务切分的核心不是「切小」，而是切到**你能判断它对不对**为止。第五章专门讲。

旋钮三：验收——你怎么知道它做对了

AI 交付的东西不能默认是对的。你需要一条质检线：什么地方抽查、什么地方全查、什么错误绝对不能放过。没有这个旋钮，前面拧得再好，最后一步全漏光。第六章讲。

旋钮四：结果管理——产出怎么放、怎么比、怎么找回来

这是整本书最被低估的一个旋钮，值得多说一句。现在大家用 AI，产出都散落在一条条聊天记录里：生成的文案、代码、方案，聊完就沉底，下次想用找不到，想比较两个版本没法

比，想知道「上个月那版为什么更好」无从查起。

绝大多数人还在琢磨会话管理——怎么整理聊天记录、给对话分类。但真正值钱的是结果管理：AI 给你的东西，怎么高效地呈现、比较、存档、复用。第七、八两章专门讲这个，这也是做 AI 产品的人最该想的事。

旋钮五：信任校准——什么时候信，什么时候查

对 AI 全信，会翻车；全不信，等于没用。正确姿势是心里有本账：哪类任务它几乎不翻车，哪类任务它翻车率三成，按账本来分配你的注意力。第九章讲。

旋钮六：并行和流程——从带一个到带一群

当你单线程地用 AI，你的上限是你自己盯屏幕的时间。AI 不累，你可以让它三路并进；你还可以把反复好用的做法沉淀成模板和流程，下次直接套。第十章讲。

剩下的：模型选择、提示技巧、工具接入

还有一些旋钮是真实存在但容易被高估的：选哪个模型（差距通常没你想的大，而且厂商替你拧）、提示词技巧（有用，但本质是上下文旋钮的子集）、接什么工具（重要，但属于工程问题）。这本书会把注意力集中在你能拧、且拧了立竿见影的那几个上。

地图的用法

这张清单的正确用法，是拿来给自己做个体检：每个旋钮问自己一句——「这个我认真调过吗，还是一直停在出厂位置？」

大白话

大白话：别问 AI 行不行，先问自己这排旋钮有几个动过。

接下来我们一个一个来，从杠杆最大的那个开始：上下文。

第四章 · 上下文是新的入职培训

一个每天都失忆的新员工

想象你招来一个非常聪明的新员工：名校全优，博览群书，反应极快。但有两个怪癖：第一，他对你们公司一无所知；第二，**他每天早上都会彻底失忆**，昨天你教他的东西，今天全忘了。

这就是 AI。它训练时读完了半个互联网，但你桌上的事——你的客户叫什么、你们老板讨厌什么风格、上次这个方案为什么被毙——它一概不知。每次新对话，它都是第一天上班。

带这样的员工，什么功夫最值钱？**入职培训的功夫**。管理学里叫 briefing——开工前把背景、目标、边界一次性说清楚的那个动作。对失忆员工来说，这不是入职第一天做的事，而是**每次开工都要做的事**。

大白话

大白话：AI 不是不懂你，是你根本没跟它说过。

为什么这个旋钮杠杆最大

道理很朴素：AI 的输出质量上限，由它能看到的信息决定。你再聪明，脑子里有二十年行业经验，只要你没打出来，对 AI 来说就不存在。

人和 AI 之间有一条严重的信息不对称：你知道得太多，它知道得太少，而唯一的传输通道是你敲键盘的速度。所以管理 AI 的第一个觉悟是：**你在上下文上省的时间，最后都会在改稿上还回去，还加倍。**

更划算的想法是把 briefing 当成一次投资：十分钟的背景交代，常常省下一小时的返工。这个回报率，很少有别的动作比得上。

一份好 briefing 里有什么

不用背模板，记住「新员工第一天需要知道什么」，照着给就行：

- **这事是给谁看的。**给老板、给客户、给自己存档，三种写法完全不同。
- **什么叫好。**最好给一两个你认可的样本——「照这个感觉来」比十句形容词都管用。反例同样值钱：「千万别写成上次那种八股文」。
- **边界和忌讳。**篇幅、口径、不能碰的红线、必须包含的要点。
- **你已经知道什么。**你的数据、你的判断、你的猜测。AI 最值钱的用法不是替你从零想，而是在你的判断基础上往前推。
- **你不确定什么。**明确说「这一点我也没把握，你帮我看看」，比假装胸有成竹得到的回答好得多。

注意这张单子没有一条是「提示词技巧」。它全是**管理工作**——恰恰是带人多年的好经理天天在做、而没做过管理的人最容易忽略的事。

一个反着用的检验

下次 AI 给了你一个不满意的答案，别急着骂它笨。先做个检验：把你自已发的那些话原样拿给一个聪明的陌生人看，问他「凭这些话，你能交出我要的东西吗？」

如果答案是「不能」，那问题不在 AI 的智商，而在你的 briefing。这就是第一章那个公式的具体含义：智商那一项很高了，**卡住你的是你给的信息**。

上下文解决了「它知道什么」。下一个问题是：让它干什么——任务应该切多大块？

第五章 · 把任务切成能验收的块

委派的第一定律

带过团队的人迟早会悟到一件事：委派任务的失败，十有八九不是下属不行，而是**任务本身没法验收**。你说「把活动办漂亮点」，对方办完，你说不够漂亮，他问你什么叫漂亮，你答不上来——这个局从一开始就是死的。

对 AI，这一定律不但成立，而且更硬。人类下属还会察言观色、会主动来问；AI 不会，它会信心满满地把你那句模糊的需求，平推成一份看起来很像样的东西。所以带 AI 的第一定律是：

大白话

大白话：好任务只有一个标准——交付的时候，你能判断它对还是不对。

为什么块不能太大：错误是相乘的

假设 AI 每一步做对的概率是九成——这已经是很给面子的估计了。那么：

- 一个一步能完成的任务，成功率 90%。
- 一个要连对五步的任务，成功率是 0.9 的 5 次方，约 59%。
- 一个要连对十步的任务，成功率掉到约 35%。

这就是「帮我写本书」必死的原因：它不是一个任务，是几百个串起来的步骤，而你只打算在最后验收一次。中间任何一步走偏，后面的全是精致的浪费。

切小块的真正作用，不是「让 AI 更轻松」，而是**让错误尽早暴露**。每一步都验收，错误就停在它发生的那一步，而不是复利式地滚到结尾。软件开发里管这个叫「快速失败」，瀑布开发几个月才发现方向错了的惨痛，和一次性让 AI 写五十页的惨痛，是同一种惨痛。

怎么切：先切到「能判断对错」为止

切分没有固定尺寸，只有一个停止条件：**这一块交回来，我能不能在几分钟内判断它行不行？**

判断不了，就继续切。一份市场报告切不动，就切成：先给三个可选的切入角度（这个好判断）；选定后给提纲（也好判断）；然后一节一节写（每节都能看）；数据单独核（能查）。每一步的验收成本都很低，合起来的总质量却远超一口气写完。

这里有个常见的误会要拆掉：切小块**不是**效率的敌人。很多人觉得「我一次让它干完多省事」，算的是提示词的账，没算返工的账。一口气生成五十页然后全部推翻重来，和分五步生成、每步花两分钟验收，后者几乎总是更快——而且结果好得多。

说不清对错怎么办

总有些任务，你自己也说不清什么叫对——比如「帮我想个 slogan」「这个方案好不好」。这类任务的切法是换一个方向：**不让它交答案，让它交选项。**

「给我十个方向，每个一句话，附各自的代价」——判断「十个里哪个顺眼」远比判断「这一个好不好」容易，因为人天生擅长比较、不擅长凭空打分。拿不准对错的任务，就把它改造成选择题。这本身就是一种验收设计。

任务切好了，验收的标准也有了。下一个旋钮顺理成章：验收这条质检线，具体该怎么搭？

第六章 · 评审是质检线

没验收的产出不是资产，是负债

先记住一个有点刺耳的判断：**一段你没检查过的 AI 产出，在你的账上应该记成负债，不是资产。**

为什么？因为它用起来像资产——能交差、能填坑、看起来像那么回事——但里面埋着未知数量的错误。每个错误都是一颗定时炸弹：报告里错的一个数字，会在客户会议上爆；代码里错的一个边界条件，会在上线当晚爆。没验收就入库的东西，你是在赊账，利息照算。

这就是为什么第五章说「切到能验收」：切分只是让验收*可能发生*，这一章讲的是验收怎么真的发生。

流利不等于正确

评审 AI 产出最大的心理障碍，是它写得**太顺了**。排版整齐，语气自信，滴水不漏——人脑对流畅的东西有天然的信任倾向，这在心理学上叫「流畅性启发」：读起来越顺，越觉得对。

对 AI，这个直觉必须手动关掉。它把错话写得和对的话一样顺，这是它的工作原理决定的：它生来就是在生产「看起来像样的话」，正确性是额外的、不保证的属性。你越觉得「这段写得真好，应该没问题」，越要小心。

大白话

大白话：AI 的自信是出厂设置，不是质量保证书。

质检线分三层

评审不是「通读一遍看看顺不顺」。工厂质检也不是老师傅从头摸到尾，而是分层：

第一层：能机器验的，绝不人眼看

数字、链接、引用、代码——这些有客观对错的东西，最不该用眼睛抽查。数字加总用程序核一遍，链接点开看死没死，代码跑测试。AI 写的代码之所以是 AI 产出里最好管的一类，恰恰因为测试这层自动质检是现成的：跑过了就是跑过了。

第二层：判断性的内容，抽样深查

观点、论证、措辞——没有客观对错，但有好坏。不必每段都细读，抽几段往深里问：这个论据成立吗？这个结论跟前面的数据对得上吗？抽到的合格率高，整批放行；抽到一个严重问题，加严再抽。这是质检行业用了一百年的抽样逻辑，直接搬过来就行。

第三层：错了会死人的，全查

对外发布的数字、法律口径、不可逆的操作——代价不对称的地方，不适用抽样的经济学，必须全查，最好换人、换方法交叉查。

用 AI 审 AI

一个越来越实用的手法：让另一个 AI（或同一个模型换个身份）当评审。做法是明确的对抗姿态——「假设这份报告里有三处硬伤，找出来」「以挑刺为乐，逐段攻击这个方案」。

这为什么有效？因为生成和评判是两种不同的能力，评判往往更容易——就像改别人的文章比自己写容易。让 AI 挑 AI 的刺，等于花几分钱雇了一个不知疲倦的质检员。当然它不是万能的（它可能漏，也可能误报），所以它是第二层抽样的好帮手，而不是第三层全查的替代品。

评审要算进总账

最后一个提醒：评审花的时间，是 AI 工作流的**正常成本，不是意外损耗**。算「AI 帮我省了多少时间」的时候，必须把验收时间记进成本栏。记完之后账还是漂亮的——验收一份东西远比从头做一份快——但只有记了这笔账，你才知道哪些任务用 AI 真划算，哪些其实不划算。

产出验收合格了，然后呢？大多数人让它们在聊天记录里沉底。下一章要讲的是这本书最核心的一次观念转换：从管理「会话」，到管理「结果」。

第七章 · 从会话到结果

看看你上一个 AI 产出在哪

做个小实验。想想你上周用 AI 做出来的最重要的那个东西——一份报告、一段代码、一套方案。现在，不要打开任何软件，回答三个问题：

- 它现在存在哪？你能十秒钟内找到吗？
- 你当时试过几个版本？为什么留下的是这版？
- 三个月后你想复用它，你打算怎么找？

如果对大多数人诚实，答案大约是：它躺在某个聊天软件的第三十七条对话里，和我问「红烧肉怎么做」的那条挨在一起。

整个行业在卷错的东西

这里有一个很少有人点破的现象：这几年 AI 产品在「管理」上做的几乎所有努力，都是会话管理——对话列表、对话分组、对话搜索、对话置顶、对话导出。手机厂商卷拍照的时候，AI 厂商在卷怎么帮你整理聊天记录。

但聊天记录不是资产。想想邮件：没有人会去「管理邮箱线程」当成目的，线程只是过程，你真正关心的是线程里那几封带合同、带结论的邮件。会话之于 AI，正如线程之于邮件——**它是生产的流水线，不是生产出来的东西。**

真正值钱的是结果：AI 交付给你的那些成品——文档、代码、表格、方案、图片。而今天，这些结果几乎处在无人管理的状态：散落在几十条对话里，没有名字，没有版本，没法比较，没法追溯，没法复用。我们造出了一台产能惊人的机器，然后让它的产出直接掉在地上。

大白话：大家忙着整理「和 AI 聊了什么」，没人管「AI 交出了什么」。前者是过程，后者才是工资该买的东西。

为什么这个转变现在才发生

会话为中心不是谁的愚蠢，它是历史的自然顺序。聊天是人和 AI 最早的界面，就像命令行是人和电脑最早的界面——先有交互方式，然后人们才意识到交互的产出需要管理。

但有两个变化让「结果管理」从可有可无变成了刚需：

第一，产出量爆炸了。一天生成五段文字，记在脑子里就行；一天生成五十份草稿、二十版代码，脑子根本管不过来。量过了临界点，管理方式必须换代——Excel 管得了十家客户，管不了一万家。

第二，产出开始真正被使用。早期的 AI 产出多是「看看玩」，丢了不心疼；现在它们进报告、进产品、进客户的眼睛。一旦产出进入了正式流程，「这版是哪里来的、谁验过、和上一版差在哪」就从洁癖变成了责任。

结果管理管什么

把「结果」当成正经资产来管，具体就是四件事：

- **呈现**：结果要以结果的样子出现——一份文档就是一份文档，而不是埋在三千字闲聊中间的几个段落。
- **比较**：同一任务的几个版本、几个模型的答案，能并排放着比。
- **归档**：每个结果有名字、有归属的任务、有状态（草稿/已验收/已交付），找得到。
- **追溯**：从任何一份结果能回到它的来历——当时的任务是什么、给了什么上下文、谁验收的。

程序员对这套东西熟得不能再熟——这就是版本管理、代码评审、变更记录在做的事。写代码的人早就明白：**产出的管理，和产出的生产，是同一件事的两半。**只不过这一次，需要这套东西的人从程序员变成了所有人。

一次观念的搬家

从会话到结果，看起来是软件功能的事，实际是观念的事：你雇 AI 是为了它交出来的东西，不是为了和它聊天。聊天框里的热闹是过程，能留在桌上、能被复用、能追责的成品，才是你花时间花电费真正买到的东西。

下一章把这四件事拆开揉碎：呈现、比较、归档、追溯，每一件具体怎么做，以及为什么现在的工具大多不及格。

第八章 · 结果的呈现、对比与归档

把结果当代码管

上一章说了结果管理的四件事：呈现、比较、归档、追溯。这一章把它们落地。最省事的做法，是直接抄程序员的作业——他们管「产出」管了五十年，工具链的名字叫版本管理。AI 的结果，值得享受同等待遇。

呈现：结果是一等公民

打开任何一个 AI 聊天产品，你会看到同样的景象：你问的话、它的客套、中间跑题的三轮、最后的成品——全部平铺在一条流里，成品被泡在水里。

好的呈现只有一条原则：**让结果脱离对话，成为独立的东西**。一份文档，就是一个干净的文档页面，有标题、有正文，没有「好的，我来帮您」；一段代码，就是一个代码文件，能从对话里一键拿出来。对话是车间，成品要摆上货架，没有人把车间当仓库用。

大白话

大白话：聊天记录是快递箱，里面的东西才是你买的。别连箱子一起收藏。

比较：并排是人类的超能力

第五章说过，人擅长比较、不擅长凭空打分。结果管理要把这个超能力喂饱：同一个任务的三版草稿，能不能**并排**摆开？改过的版本和上一版，**差异**在哪，能不能高亮出来？

程序员管这个叫 diff——两个版本之间的差异清单。它是所有协作工具的基石，因为「这次改了什么」是评审者最关心的问题。而今天的 AI 产品里，你让 AI 改一版稿子，它输出全新的一份，两处改动的位置要你自己用眼睛捞针。这相当于程序员退回没有 diff 的年代——明明一个早就解决的问题，在 AI 界面上又复活了。

比较还有一个更大的用法：同一道题，同时扔给两个模型，答案并排。这既是挑结果，也是第九章要讲的信任校准——它们一致的地方可以快读，分歧的地方正是你该花精力的地方。

归档：给结果一个身份

能找回来的结果才是资产。归档的要义不是「存下来」（聊天记录本来就存着），而是给每份结果一个**身份**：

- **名字**：不是「新对话 37」，而是「三季度渠道分析-终稿」。
- **归属**：它属于哪个任务、哪个项目。同一任务的历次版本挂在一起，最新的在上面，旧的没丢。
- **状态**：草稿、已验收、已交付——第六章的质检线在这里留下痕迹，没验收的永远不许混进「已交付」。

追溯：从成品能回到车间

最后一件，也是最被低估的一件：任何一份结果，都应该能一键回到它的来历——当时的任务描述、给过的上下文、用过的版本历史、谁验收的。

为什么重要？因为结果总会出问题。三个月后客户问「这个数字哪来的」，老板问「这版方案为什么这么写」，你要么能顺藤摸瓜回到源头，要么只能脸红。软件行业管这叫可追溯性，出了事故靠它定位；AI 产出越多地进入正式流程，这条藤就越不能断。

而且追溯还有别的用处：当你发现某个结果特别好，顺藤回去看「那次我做对了什么」，你就提炼出了一条自己的最佳实践——这是第十章「管理的复利」的原料。

一个诚实的现状

说到这里你可能发现了：这四件事，没有一件高深，全是成熟行业的老做法。但今天的 AI 产品大多一件都没做好——它们把所有心思花在了车间（对话体验），货架（结果管理）还是毛坯房。这个落差不是疏忽，是一个还没被占领的产品空间。最后一章我们会专门聊它，先别急。

下一章回到一个更私人的旋钮：面对 AI 的产出，你到底该信几分？

第九章 · 信任的校准

两种翻车方式

用 AI 的人会在两个相反的方向上翻车。

第一种是**信过头**。AI 给的数字直接抄进报告，AI 写的代码不看就上线——直到某天当众爆炸。这种人犯的错叫「过度信任」。

第二种是**不信**。每个字都自己核对，每段代码都自己重写一遍——最后算下来，用 AI 比自己干还慢。这种人犯的错叫「信任不足」，它不爆炸，所以更难被发现，但它让你把这本书前面所有的旋钮都白拧了。

正确的位置既不是全信也不是全不信，而是一个准确的中间态：**校准**——你对 AI 的信任程度，要和它真实的可靠程度匹配。它九成可靠的领域你信九成，七成可靠的领域你信七成。

学开辅助驾驶

最好的类比是辅助驾驶。好的辅助驾驶用户脑子里有一张清晰的图：高速跟车几乎不用管，进出匝道手要回到方向盘，雨雪天干脆自己开。他不是「相信这个系统」或「不相信这个系统」，他相信的是**分场景的可靠性**。

糟糕的司机只有两种：要么开辅助驾驶睡大觉，要么因为「机器不可信」而一路自己开、累个半死。车是同一辆车，差别全在这张脑内的图上。

用 AI 也是一样。你需要的是自己的那张图——**按任务类型分开记账的翻车率**。

建你自己的校准表

通用说法（「AI 写代码很行」「AI 写数字容易错」）只能当起点，真正的校准表是你自己攒出来的，因为可靠性取决于**你的任务、你的领域、你的验收标准**。方法朴素得可笑：用，出错，记下来。

用久了，你会有这样一本账：

- 「改写、润色、翻译——几乎不翻车，快读放行。」
- 「总结我给的资料——基本可靠，但会丢我不希望丢的细节，抽查。」
- 「凭它自己的知识答具体事实（日期、人名、法规条文）——翻车率不低，凡是要引用必查原文。」
- 「算数——不许让它心算，让它列过程或用工具。」

这本账没有一条是定律，换个模型版本就要重记——这正是关键：**校准是一次性的学习，更是一项持续的维护**。每次模型升级，聪明的做法不是庆祝，而是拿几个你出过错的旧任务重新试一遍，更新你的账。

两个加速校准的技巧

第一，用分歧当警报器。同一道题问两个模型（或问两次），答案一致的地方通常可以快速，分歧的地方几乎一定是该你细看的地方。你不必知道谁对——分歧本身就是「此处可靠性存疑」的路标。这比通篇精读省力得多。

第二，按代价分配信任，而不只按概率。翻车率两成的任务，如果翻车了改一行字就好，可以闭眼用；翻车率只有百分之二的任务，如果翻车是发给一万个客户，也必须查。信任的正确单位不是「这题多容易错」，而是「错误概率 × 错误代价」——这正是第六章三层质检线背后的同一条算术。

大白话

大白话：别问「AI 可信吗」，问「这件事上它值几分信任」——你的账本会告诉你。

到目前为止，我们说的都是怎么管好「一个 AI」。可它不累、不贵、可以同时干十件事——为什么你只让它干一件？下一章，从带一个到带一群。

第十章 · 从带一个到带一群

你的瓶颈又搬家了

前面九章解决的是「管好一个 AI」。但请抬头看一眼这个荒谬的画面：AI 不知疲倦、随叫随到、雇佣一个和雇佣一百个的成本几乎一样——而你让它一次干一件事，干完了等你回来，看完才派下一件。

这好比你管着一百个工人，却要求他们排队，每次只有一个人进你的办公室领任务。瓶颈早就不在工人身上了，**瓶颈是你的注意力**——你派发任务和验收结果的速度。

认识到这一点，玩法就变了。问题从「怎么让 AI 干好一件事」变成「怎么让我一个人同时管住十件事」。

并行：让等待重叠

最简单的一步，是消灭串行的等待。AI 干活你要等，等它写完报告初稿的三分钟里你盯着屏幕——这三分钟，另外三个任务本可以同时跑：让它写报告的同时，另一个窗口在整理数据，第三个在起草邮件。

这听起来像废话，但看看自己的实际行为：大多数人用 AI 的姿势是**一次一个、全程陪同**。把它改成「派完一个就派下一个，回头统一验收」，同一下午的产出直接翻倍。你唯一需要升级的，正是第七、八章说的结果管理——三条线并行的产出，没有归档和状态标记，立刻乱成一锅粥。这就是为什么结果管理是并行的前提，而不是并行的点缀。

编排：从派活到设计流水线

再进一步，你不再手动派活，而是**设计流程**：一个复杂任务拆成几步，每步谁来干、谁审谁、什么条件进下一步，事先定好，然后让它自己跑。你只在验收节点出现。

这一步的实质，是你从「操作工」变成「车间主任」。操作工的时间花在做，车间主任的时间花在**设计谁做什么**。有意思的是，第六章讲的「用 AI 审 AI」在这里变成了标配：写作的

产物交给评审，评审不过的打回重写，你在终点盖章——一条最小流水线就长这样，而你已经在这本书的写作现场见过它了（这本书就是这么生产的）。

管理的复利：把做法变成模板

带一群人最贵的地方不是派活，是**同样的活反复派**。每次重新想 briefing、重新定验收标准，你的注意力就被消耗一遍。解决办法是所有好经理都会的那招：沉淀。

- briefing 做得好的，存成模板：给这类任务的固定上下文，下次填空。
- 验收清单攒下来：这类产出必查哪五项，做成固定条目，不靠记性。
- 翻车的案例记下来：第九章的校准表，本质上就是翻车牌的沉淀。

每一次沉淀，下一次同类任务的成本就降一截。这就是管理的**复利**：你的经验不再只存在你脑子里随取随忘，而是固化成流程，在你睡觉的时候也替你工作。普通用户和高手的差距，拉到一年尺度看，主要不是谁聪明，而是**谁把自己的做法存下来了**。

一个清醒的边界

并行和编排放大的是你已经会的东西。如果你单线程时就 briefing 不清、验收不严，开十条线只是把混乱放大十倍——垃圾的流水线产量更高。所以顺序不能反：先把前九章的旋钮在一个任务上拧顺，再谈放大。

到这里，「怎么用 AI」的主体讲完了。还剩两个收尾问题：你怎么证明自己真的变快了（第十一章），以及——如果你是做 AI 产品的，这本书说的这一切对你意味着什么（第十二章）。

第十一章·你怎么知道变快了

「感觉快了」不算数

问十个重度 AI 用户「AI 让你快了多少」，九个会说「快多了」，再问「多多少」，全都卡壳。这不是谦虚，是真不知道。而不知道是有代价的：度量不了，你就分不清哪些用法真划算、哪些只是**感觉**划算——前九章的旋钮往哪边拧，全凭手感。

这一章解决一个问题：怎么知道自己是不是真的变快了。

先避开几个假指标

最容易拿到的数字，几乎都是错的：

- **AI 生成的量**。生成了五千行代码、三十页文档——这是机器的工作量，不是你的产出。生成五版废稿比生成一版能用的，这个数字还更高。
- **对话轮数、调用次数**。这只证明你很忙。拿它当成绩，就像用「打了多少电话」考核销售。
- **单次任务的提速**。「这份报告以前三小时，现在四十分钟」——听着不错，但它是挑出来的成功案例，不是你全天工作的样子。

这些数字共同的病：它们度量的是**过程的热闹**，不是**结果的交付**。管理学里有个老话，一项指标一旦成了目标，就会被人不自觉地优化到失真——为了「生成更多代码」而生成更多代码，为了「多用 AI」而把简单的事也扔给 AI。假指标不只没用，还会把你带偏。

度量什么：端到端的结果

真指标只有一类：**一件完整的事，从领来到交付，花了你多少时间**。

注意两个词。「完整的事」：不是「生成初稿」这一步，而是包含 briefing、生成、验收、修改、交付的全过程——第六章说了，验收时间是成本的一部分，记账必须记全。「交付」：以对方接受为准，不是以你导出文档为准。

操作层面简单得惊人：挑你每周重复做的两三类事，各记几笔时间账，一个月后就有了一条 baseline；然后拧一个旋钮，再记一个月。数字会自己说话——哪些旋钮真有用、哪些是你的幻觉，一目了然。不用精确到分钟，粗粒度就够，因为我们要找的是成倍的差异，不是百分之五。

最难测的那个维度：质量

时间好记，质量难测，但不测质量的提速没有意义——交得快但被打回来的活，是负产出。可行的替代办法是用**返工率**当质量的影子指标：一件活交出去，被打回来大改几次？这个数字好记，而且诚实。提速如果伴随着返工率上升，那不是提速，是把成本转嫁到了后面。

省下的时间去哪了

最后一笔账，也是最容易赖掉的一笔：AI 每天帮你省下两小时，这两小时**实际**去了哪？

如果答案是真的干了更多活、想了更深的问题、或者光明正大地休息了——都算数，都是赚。但如果它悄悄漏进了刷手机和更频繁地「检查一下消息」，那么你的效率提升是个会计假象：分子变大了，分母也跟着变大了。

大白话

大白话：省下来的时间没花出去，就等于没省。

度量不是为了写总结汇报，它是指挥棒：没有它，你分不清哪个旋钮真灵。有了它，这本书前面所有的话才从「听起来有道理」变成「在我这儿确实有用」。最后一章，换个身份：假如你是做 AI 产品的那个人。

第十二章 · 给做 AI 产品的人

一块没人圈的地

如果你是做 AI 产品的，这本书读到这儿，应该会有点坐不住了：前面讲的几乎所有痛点——结果散落、没法并排、没有版本、验收无痕、追溯无门——**全是产品问题，不是用户问题**。用户拿聊天框当仓库，是因为你只给了他聊天框。

这就是第七章说的那个判断的产品版：行业的力气花在了会话上，结果管理还是一块没人认真圈的地。谁先圈下来，谁就先解决了一个真实且日益尖锐的痛。

界面即立场

产品有一个朴素的事实：**你给用户什么界面，用户就形成什么习惯**。界面是主语，习惯是宾语。

以对话为中心的界面，教会用户的是「和 AI 聊天」——所以产出聊完就散。而一个以结果为中心的界面会教出完全不同的习惯：任务发出去，结果一件一件回来，摆在货架上，等着验收、比较、归档。用户不是不想这么干，是现在的界面里根本没有这些动作的落点。想改变千万人的使用方式，说教没用，**把动作做成按钮**才有用。

几条设计原则

照着这本书前面的章节，一个「结果优先」的 AI 产品长什么样，其实已经能画出来了：

- **结果是一等公民**。对话是流水线，产出是货架上的货。每份结果有独立的页面、名字、状态，可以从对话里被「取出来」——而不是埋在对话里被「翻出来」。（第七、八章）
- **天生并排**。同一任务的多版本、多模型的答案，默认并排呈现、差异高亮。别让用户用眼睛做 diff。（第八章）
- **验收是流程的一环**。「待验收/已验收/已交付」是结果的基本状态，质检结论留在结果上。未验收的东西，产品应该阻止它被误当成品发出去。（第六章）

- **来历一键可查。**从任何结果能回到任务、上下文、版本历史。用户三个月后那句「这数字哪来的」，产品要替他回答。（第八章）
- **为并行而生。**用户的注意力是瓶颈，所以界面要让人一眼看到十条并行的线各跑到哪了，而不是十串聊天气泡。（第十章）

更深的一层：管理的界面

往远看一步。AI 的智商还会继续涨，这意味着「管理工作」本身也在升级：从管一个助手，到管一个团队，到管一条流水线。用户需要的不再是一个聊天框，而是一个**管理界面**——就像工厂的主任需要仪表盘而不是亲自下流水线。

仪表盘上该有什么？正在跑的任务、待验收的结果、各处的翻车率（用户的校准表应该由产品帮他记账，而不是靠脑子）、各旋钮带来的真实提速（第十一章的时间账，产品完全可以替用户记）。谁把这套东西做出来，谁就不只是「又一个 AI 应用」，而是 AI 时代的管理工具——一个现在几乎还空着的品类。

回到开头那个问题

这本书从一个思想实验开始：假如 AI 的智商从此冻结，你的效率还能涨多少？走了一圈，答案应该已经清楚了——上下文、切分、验收、结果管理、校准、并行、度量，每个旋钮都有大段没拧到头。

而且好消息是，这两个变量不相斥：模型会继续变聪明，那是实验室替你攒的；管理能力是自己的功夫，谁也拿不走，而且**它跟着模型一起升值**——引擎越强，会开车的人越值钱。

所以，别只盯着发布会。低头看看你手边那排旋钮。