AI 上完之后，

组织怎么办

从工具上线到组织能力的系统重写

J叔

AI 上完之后，

组织怎么办

从工具上线到组织能力的系统重写

J 叔 著

AI 上完之后，组织怎么办

从工具上线到组织能力的系统重写

作者 J 叔

版本 R15 · 官网公开版

发布网站 theunclej.com

本书由作者在个人官网公开发布，提供在线阅读及 PDF、EPUB 下载。

关于案例与来源

本书结合作者实践、公开案例与研究资料展开讨论。涉及非公开经历的案例，对个人与企业的识别性信息作了必要处理；公开案例与研究资料的出处，见正文及附录。

书中用于解释方法的场景与示例，不应视为某家企业的完整事实记录。

我写这些，是为了把组织里的问题讲清楚，不是让读者猜公司、猜人名。

CONTENTS

目录

PREFACE

序言

AI 上完之后，组织怎么办

最近见过一位企业负责人。他说想看看 AI 转型，聊到 RAG，也就是让 AI 先查公司自己的资料再回答，聊到企业知识库该怎么管，他的反应很轻：“这个不复杂，找两个新人整理一下就行。”那一刻，我脑子里冒出四个字：叶公好龙。

让我警惕的，是把企业知识库治理想成资料整理。那场会谈给我的，是一个需要继续追问的信号，不足以证明这家企业的全部组织安排都出了问题。

资料可以请新人整理，来源可信度、维护责任、使用权限、错误处理和反馈回流，却不能靠“整理一下”四个字带过。两个人能不能把事情办成，要看这些安排是否已经存在，他们有没有相应的信息、权限和资源，不是看他们够不够勤奋。

如果真正的缺口没有查清，只报一个系统上线日期，缺口不会因此消失。

账号、系统、培训和试点，让 AI 有机会进入公司。它的输出一旦被真实工作依赖，就要继续查：谁有权使用，谁维护依据，质量怎样检查，出了问题怎样处理。组织能力不能只靠工具清单来证明。员工对知识贡献的顾虑，也该进入这个检查：

我贡献给组织 AI 的判断，最后是用来放大我，还是替代我？

如果这个问题没有答案，AI 用得越多，组织未必越聪明。它可能只是让几个聪明员工更快，也可能让流程更乱，让责任更模糊，让知识更分散，让员工更会保护自己。这本书处理的，就是这个问题。

本书要回答的是：可调用的能力，如何成为可持续的组织能力。

我的主张是，AI 改变了工作的能力边界以后，企业要检查原来的授权、协作、知识更新和结果承接，是否还覆盖得住新的工作。覆盖得住，就保留；有明确失配，就比较修复办法。该改的是已经查明的问题，不是为了上 AI 先重画一张组织图。

模型不适用、数据拿不到、需求本身不成立，也会让项目走不下去。先找瓶颈，别把所有问题都叫成组织问题。

这本书不打算教老板买哪一个工具。它要帮老板追问：为什么输出快了，交付仍然慢？为什么工时省了，收益没有出现？为什么知识存进去了，下一次还得从头解释？为什么每一步都有人确认，出了问题却没有人能处理？

这些问题要逐个定位，也要逐个验证。先说清要改善哪个结果、哪处接口妨碍了它，再看改动有没有产生预期变化，连同复核、维护和处理后果的成本一起算。规则写全、名字填上，只说明有了可以检查的安排，不说明事情已经做成。

我把这件事称为组织底层系统升级。它不以改了多少岗位、增加多少 Agent 为成绩。工作的授权、控制和责任要接得上，结果还得持续拿得回来。

岗位、流程、知识、责任、治理，是本书选择的五个诊断入口。它们帮助我们定位问题，不是五种必须出现的病。先看岗位，是为了把“裁不裁这个人”拆成“哪些任务变了，哪些结果还需要承接”；再沿着工作往下查，看交接、知识和控制在哪一步出了缺口。

诊断之后，需要有人把问题落实到具体改动：业务要什么，系统能做什么，哪些承诺可以交出去，异常又由谁处理。这些职责可以由现有岗位承担，也可以由临时小队或专门角色协作完成。书中讨论的现场翻译与孵化角色，是一种组织办法，不是必须新增的职位。

这更像一份给企业一号位的改造说明书：选一条真实流程，写清目标与边界，试着修一处，再按事前约定的结果决定保留、扩大、缩小还是停止。书中的 90 天、180 天安排提供推进节奏，不保证任何企业都能按同一日历完成转型。

写到最后，问题还会再往前走一步：当执行成本和协作条件继续变化，什么工作值得放在一个持续的组织安排里？最后一章提出责任单元作为一种可比较的设计，并说明它需要接受哪些检验。现有部门、项目制或流程负责人已经做得一样好，就没有理由为了这个名字另起一套。

合上这页之前，可以先拿一条正在运行的 AI 流程问自己：原来想改善什么，现在实际改善了什么，付出了什么代价？如果变化没有出现，先查瓶颈，再决定改工具、改接口，还是停下这个项目。

组织图可以不换。改善有没有发生，得拿结果说话。

先承认真问题

很多企业的 AI 转型，第一步就问错了。

它们问的是：我们要不要上 AI？买哪个工具？让哪个部门负责？培训先覆盖多少人？这些问题都不是没用，但它们都太靠后。真正该先问的是：

> AI 进来以后，我们这家公司到底有没有一套组织方式能接住它？

如果组织接不住，工具越多，噪音越多；Demo 越多，幻觉越重；员工越会用 AI，组织越可能变成一堆个人外挂的集合。每个人都在变快，但公司没有变聪明。

第一部先不急着给方案，先把真问题摆上桌。AI 转型不是上一轮信息化。信息化很多时候是把已有流程搬进系统里，AI 更麻烦，它会直接插进工作本身：帮人写，帮人查，帮人判断，帮人复核，甚至开始触发下一步动作。

一旦 AI 插进工作，岗位、流程、知识、责任和治理都会被带着动。

如果一号位还只把 AI 当工具采购，这场转型很容易从第一步就跑偏。

所以这一部先不谈“AI 能不能用”，那个问题已经有答案了。它要摆上桌的是另一个：

> 为什么 AI 上了，组织还是没变？

CHAPTER 01

为什么 AI 转型不是买工具，而是组织宪法重写

最近我越来越怕一种 AI 转型会。

不是老板不重视。恰恰相反，预算批了，账号开了，供应商 Demo 看过了，内部培训也做了几轮。PPT 上还有一张漂亮的推进表：谁负责采购，谁负责培训，谁负责试用，什么时候复盘。表面看，动作都有。

我会在这种时候问一个很不讨喜的问题：三个月以后，哪条真实流程变了？

很多会议会在这里安静一下。因为组织里的真实工作方式，往往没有被明显改动。原来怎么提需求，现在还是怎么提。原来怎么审批，现在还是怎么审批。原来谁拍板，现在还是谁拍板。原来知识散在个人脑子和群聊里，现在只是多了一个 AI 输入框。AI 更像是被加在旧流程旁边的一套外挂，没有进入真实生产系统。

工具上线不等于组织上线。

真正需要改写的，不只是工具清单，是我们怎么看公司。传统管理把公司看成法人、科层、资源配置机器，或者一张组织结构图。AI 进来以后，这个视角不够了：公司更像一个由人和机器组成的控制论集合体——就是一台一直在处理信号的机器，只不过这台机器里坐着人。

说得再土一点。客户投诉进来是输入，客服接单、群里 @ 主管、主管找生产查批次是处理，回给客户一个答复是输出，这个客户下次还下不下单是反馈。你公司每天都在跑这套东西，只是从来没人把它画出来。

放大到整家公司也是一样。市场、客户、监管和技术变化，是输入。会议、数据、流程、模型和经验，是处理。团队、系统、供应链和销售动作，是输出。结果再通过客户反馈、财务结果、质量事故、员工离职和模型错误，回到下一轮判断。

公司不是一堆人在一起工作。是一套持续处理信号的系统。人负责目标、价值判断和责任兜底。机器开始承担搜索、生成、监控和局部执行。通信系统让信号流动。反馈回路决定这套系统能不能学习。

这就是我说的组织 OS：组织如何感知环境、分配注意力、形成判断、调用资源、执行动作、接收反馈并更新自身的底层规则。它没有装在哪台服务器上，它长在公司的日常运转里。

预算、Demo、培训和账号开通，只完成了前半程：AI 进入公司。后半程才真正难：AI 进入组织的感知、判断、行动、反馈和更新回路。

进入组织，意味着工具要进入流程、岗位、责任链、考核和复盘。哪些任务必须让 AI 参与，AI 给出的东西由谁复核，复核标准是什么。结果变好时收益算谁的，结果出错时责任落在哪里。这些都要进入日常工作，不是写在方案里就算数。

所以，“能不能用”只是技术状态，“有没有转化”是管理状态。中间隔着任务拆解、数据来源、权限边界、复核机制、结果验收和责任归属。少了这些，工具会越来越多，组织仍然是原来的组织。

有钱公司和没钱公司，都可能在这里掉坑。

有钱公司的 AI 问题，往往不是资源不足。它可以买更好的模型服务，可以给员工开账号，可以请外部团队培训，也可以同时跑多个试点。资源充足会加快开始，但不一定带来清晰决策。恰恰因为能买、能试、能堆资源，真正的问题会被延后：AI 释放出来的产能，到底要服务什么组织目标？

一个团队用 AI 把报告时间缩短了一半，这一半时间怎么处理？让员工做更多项目，提高分析深度，减少加班，还是重新调整岗位配置？如果客服、销售、运营、研发都开始用 AI 提效，哪些收益归团队，哪些收益归公司，哪些会变成新的绩效要求？如果某些岗位因为 AI 变得产能过剩，组织是裁撤、转岗、扩大业务，还是把人力投入到过去做不了的事情上？

这些不是技术问题，是产能分配问题。工具已经进来了，效率也看得见一点，但公司没有提前定义效率提升之后的去向。基层会担心“我越用越证明自己

可被替代”，中层会担心“我推动越多越可能损害团队稳定”，高层又会觉得“投入已经不少，为什么组织变化不明显”。

每一层都在等另一层先给答案。

这就叫决策真空。

没钱公司的问题看起来相反：预算有限，试错空间小，老板不愿意长期投入，团队也没有专门的人去研究 AI。但它们不一定缺热情。真正的问题是，它们很难把 AI 从“听起来有用”翻译成“这个月、这个季度能改善哪一笔账”。

对没钱公司来说，AI 不能先作为战略口号出现，要先落到流程损耗三账：时间账、等待账、知识账。

时间账问的是，哪些重复劳动正在吞掉人的时间。等待账问的是，哪些流程因为等信息、等审批、等材料而卡住。知识账问的是，哪些经验只在少数人脑子里，导致新人慢、交付慢、复盘慢。

这套流程损耗三账比“全面 AI 化”更接近经营。

一个小团队不一定需要先建设完整 AI 平台。更应该先找一个高频、低风险、能复盘的流程下手。销售线索整理，报价材料准备，售后问题归类，内容初稿生成，内部知识检索，都行。切口要小到能在两三周内看见变化，也要真到能影响业务动作。

有钱公司的问题，是资源太多以后不知道怎么分；没钱公司的问题，是资源太少以后不知道从哪切。但两类公司最后都会回到同一个问题：组织到底有没有准备好接住 AI?

所以企业要重新写清楚几件基本规则。什么工作由人做，什么工作可以交给 AI 做。什么判断必须保留在人手里，什么判断可以由系统辅助。谁有权调用 AI，谁有权修改流程，谁负责复核结果。

规则还要继续往下写。AI 带来的效率收益如何分配。员工的工作过程数据能不能被采集、训练和复用。组织如何在提效的同时维持基本信任。

规则停在旧版本，AI 再新，也只是外挂。

工作、权力、责任、分配、边界与信任，合在一起才是这里所说的组织宪法。这套东西一升级，我叫它 AI 组织 OS 升级。

过去很多企业的规则，是围绕人和传统系统设计的。岗位说明书假设任务相对稳定，流程制度假设审批链条相对清楚，绩效考核假设产出主要来自个人或团队。

AI 进入以后，这三条假设都会被轻微但持续地改动。一个员工交上来的东西，可能来自他的个人经验、团队知识库、AI 生成的内容和历史数据，四样搅在一起。一个决策建议，可能由模型先生成，再由人修正，再由系统记录。你要问“这是谁做的”，答案已经不像三年前那么好答。

如果新工作已经超出旧规则的适用范围，却没有补上安排，就可能出现责任模糊、收益不清、信任透支。更关键的是，如果没有系统负责人，AI 转型会被拆散在业务、IT、HR、法务和供应商之间。每一方都参与一点，却没人对流程、知识、权限、复核、反馈回路和结果负责。

这就是底层系统没有升级。

AI 组织 OS 先从五个入口检查：岗位、流程、知识、责任、治理。查到了失配，才谈怎么改。

岗位断点，是很多老板一上来就问错了问题。他问“要不要裁这个岗位”，但 AI 先替代的不是岗位，是任务、判断和流程片段。流程断点，是 Demo 很漂亮，但进不了真实工作；一个系统能演示，不代表有人每天用，有人每天用，也不代表流程被重写。

知识断点，是员工都在用 AI，组织却没有变聪明。Prompt 就是员工敲给 AI 的那段指令，写得好不好，差别很大。这段指令、他的判断、他的复盘、他摸出来的客户洞察、他踩过的失败经验，如果都留在个人账号里，员工走的时候，这些东西也一起走。等于每个员工都多了一台小发动机，公司却没装传动轴。人人都更快了，公司还是原来那个速度。

责任断点，是 AI 可以参与工作，但不能承担组织责任。AI 能生成建议、整理资料、识别异常。但最终谁判断、谁批准、谁兜底、谁把错误写回流程，必须有人负责。报价单上的一个数字是 AI 算错的，客户按这个数字签了单，出事那天被叫进会议室的还是销售和他的主管，不是模型。

治理断点，是 AI 用得越深，越不能自由生长。一个员工为了赶报告，把客户名单和未公开的报价整段贴进公网大模型，这件事今天在你公司发生了，你多久以后会知道，还是永远不会知道？治理不是限制 AI。是让 AI 能安全、稳定、规模化地用起来。

这五个入口，是这本书的主诊断框架。作用只有一个：让老板下一次开会知道该问什么。不是五种必患的病，也不是必须从底到顶修一遍的塔。

看到的症状	从哪里查	下一步核什么
人变快了，岗位仍照旧	岗位	任务与判断变了什么，权限和产出是否相配
演示很顺，真实工作接不上	流程	卡在哪个交接，输入、验收和异常由谁接
好方法只在个人账号里	知识	哪些判断可以复用，谁核验、更新和授权调用
有输出，后果没人接	责任	谁有信息、有权限、有资源处理结果与例外
用途扩了，边界说不清	治理	数据、输出、控制和后果的边界是否仍适用

会上有人说：“开了三十个账号。”下一句问：“哪条流程变了？”有人说：“效率高多了。”再问：“同一段时间，投入和结果怎么变？”有人说：“报价第二步改了，张伟复核。”这才有了核查入口——继续看张伟有没有权限，复核记录在哪里，结果是否真的改善。报出一个名字，还不是验收。

这不是我一个人的观察。

斯坦福数字经济实验室 2026 年 4 月出了一份企业 AI 落地报告。它看的是 41 个组织、7 个国家的 51 个项目，覆盖超过一百万名员工[①]。这些项目全都已经过了试点，能交出可量化的业务价值。

①Elisa Pereira、Alvin Wang Graylin、Erik Brynjolfsson，《The Enterprise AI Playbook: Lessons from 51 Successful Deployments》，斯坦福大学数字经济实验室（Stanford Digital Economy Lab），2026 年 4 月，p.7。

报告里最扎心的一条是：同样一个 AI 场景，在一家公司几周就跑通，在另一家公司几年跑不动。差别主要不来自模型，来自组织能不能承接。

报告把最难的挑战指向三件事：变革管理、数据质量、流程重写。这 51 个成功项目里，77% 把这三件事列为难点[①]。注意，这三件事一件都不是技术。你以为 AI 贵在模型费，真正贵的是流程要重画、数据要接上、人要愿意用、责任要有人背。这笔钱不会出现在供应商的报价单上，只会出现在你组织的时间账里。

这份报告有它的边界：它挑的全是成功案例，存在选择偏差，数字也依赖企业自报，不代表所有企业 AI 项目的平均结果。但它至少说明一件事——分水岭不在模型那一侧。

部署现场要接住什么，从 AI 公司的岗位说明里也能看见。第 22 章会展开 OpenAI 和 Anthropic 的两份公开招聘说明：部署、工作流、生产系统和企业现场，都在职责里。岗位说明不能替它们写一部战略转型史，但能提醒企业，买到模型之后还有一份活要干。模型仍然重要。

即使模型已经能完成某项任务，企业侧的问题也不会因此消失。这个能力如何进入真实流程，如何和已有系统连接。谁来用？谁来验收？出错时怎么追溯？

承担这份工作的一类角色，叫前线部署工程师，英文是 FDE，Forward Deployed Engineer。就是把系统真正装进客户业务里的那个人。

这类角色之所以变得重要，不是因为企业突然喜欢新头衔，是因为“把 AI 带进现场”已经变成一个独立问题。

接口只是最小的部分。他真正处理的，是把客户现场那些没写清楚的流程、隐性规则、数据边界、审批习惯和责任结构，翻译成一套能跑起来的系统。这个翻译动作，已经超出纯技术交付，进入组织设计。

普通企业要补问的是下一句：能生成之后，能不能部署进组织？前一问过关，不等于后一问已经有了答案。

①同上，p.13。

这里要补一个容易被忽略的边界：外部团队可以帮你把第一个系统搭起来，但外部团队不能替你长出内部组织能力。

供应商可以接系统，顾问可以做方案，AI 公司可以派人帮你做部署。但你的业务现场里，谁知道这条流程真正慢在哪里，哪些客户承诺不能交给 AI，哪个老员工脑子里的判断一旦丢了，组织就会失去一块能力？谁又知道一个流程改完以后，绩效口径、岗位职责、复核责任该怎么同步？这些东西，外部人不可能替你长期负责。

所以，企业 AI 转型最后一定会回到内部组织能力建设。

不是所有公司都要马上设一个新部门，也不是所有企业都要照搬 AI 公司那套 FDE。但每家公司都要回答一个更朴素的问题：谁在内部负责把 AI 带进真实业务现场？谁负责把一次试点经验变成下一次可以复用的方法？谁负责让 AI 从“个人会用”变成“组织会用”？

这个人可以在业务部门，可以在 AI 转型办公室，也可以是 CHO 和 CIO 共管下的项目负责人。

名字不重要，责任重要。

如果没有这个责任，AI 项目就会不断回到同一个循环：工具上线，员工试用，少数人变快，流程没变，责任没变，知识没沉。三个月后，大家又开始讨论下一个工具。

这不是转型。这是工具轮换。

这套底层系统还要说得更具体一点。它不是一个比喻，是一家公司每天运行工作的底层配置。

岗位，是组织把人放到哪里。流程，是组织让工作怎么流动。知识，是组织记住什么、忘掉什么、复用什么。责任，是组织把结果归到谁身上。治理，是组织允许什么、限制什么、怎么纠错。

AI 进入以后，这五件事都会被重新触碰。

例如，招聘里 AI 可以先筛简历，但岗位不只是“HR 会用 AI 筛简历”。真正的问题是：筛掉谁、保留谁、风险提示由谁复核、候选人的文化匹配和信任潜力谁判断、错筛以后流程怎么复盘。

例如，客服里 AI 可以先给话术，但流程不只是“客服多一个助手”。真正的问题是：AI 给出的价格、承诺、退换建议，会不会对客户形成真实承诺；哪些话必须人工确认；客户情绪价值要不要保留在人那里。

例如，知识库里 AI 可以回答问题，但知识不只是“文档能被搜到”。真正的问题是：谁维护知识，谁淘汰过期答案，谁把前线反馈写回系统，谁确保新人调用到的是最新判断。

这些问题一旦展开，就能看见 AI 转型的真实难度。难的不是让 AI 输出。是让组织接住这个输出。

所以第一章不是要给老板一个复杂理论，只要求老板换一个问法。

不要问“我们用了多少 AI”。要问“AI 改了哪条流程”。

不要问“员工会不会用工具”。要问“员工用出来的好方法，有没有沉淀成组织资产”。

不要问“AI 能不能替代人”。要问哪些任务可以交出去，哪些判断必须留在人手里，哪些结果必须有人负责。

不要问“今年能不能靠 AI 降本”。要问组织结果在 AI 介入以后，是不是更稳、更可控、更能复制。

这些问法变了，会议就变了。会议一变，组织才可能变。

AI 转型最怕的不是老板不懂模型。不懂模型可以找人补。最怕的是老板不懂组织。不懂组织的人，会把 AI 想成一条绕过人的捷径。最后绕过的不是低效。是组织能力。

这场组织诊断会，最后不要形成一份“AI 工具使用情况汇总”。那种汇总没用，它只会告诉你谁比较积极、谁比较会截图、哪个部门用了几个工具。真正有用的输出，是一张断点地图：岗位、流程、知识、责任和治理的断点在哪里。每个断点后面，都要接一个具体问题。

这张表先把疑点摆出来，再用实际记录核查。查到组织失配，就补那一处；没查到，也要回头看模型、数据和需求。工具使用次数不能代替效果，诊断表也不能代替证据。

第一章之所以要讲这么多，是为了把整本书的地基钉住。后面的章节会讲岗位怎么拆，流程怎么重写，知识怎么沉淀，责任怎么划，治理怎么进日常，但所有这些动作，都回到同一个起点：AI 不是外挂。AI 一旦进入真实业务，就会变成组织的一部分。

组织如果不承认这一点，就会用旧规则管理新能力。旧规则管理新能力，最常见的结果不是失控，是浪费：能力在那里，人也在用，但组织没有变强。

这才是最隐蔽的失败。因为它不像项目失败那么明显，它看起来还挺忙：账号在增长，培训在继续，试点在推进，汇报在更新，只是业务没有真正变，组织没有真正变。这本书要反复拆的，就是这种“看起来在转型”的失败。

这一章最后落到一个动作：不要先问“我们买什么 AI 工具”，先开一场组织诊断会。

会上只问五个问题。第一，岗位：AI 进入以后，哪些任务、判断、流程片段已经被改写？第二，流程：哪一条真实流程，因为 AI 介入而改变了节点、复核和异常升级？第三，知识：员工用 AI 产生的判断、模板、失败经验，有没有沉淀成组织记忆？

第四，责任：AI 输出错了以后，谁复核、谁验收、谁承担结果？第五，治理：哪些场景可以放开，哪些场景必须审批，哪些场景绝对不能交给 AI？

这五个问题，你不需要预算，不需要审批，下周的例会上就能问一遍。

如果答不上来，公司不是没上 AI，是还没让组织接住 AI。这一步不做，后面全是热闹，也全是风险。

这不是工具采购，是企业底层运行方式升级。

CHAPTER 02

Demo 活了，组织为什么还是死

最危险的一刻，不是 Demo 翻车。

是 Demo 很顺。屏幕上输出正常，老板点头，团队松一口气。演示的人讲完最后一句，会议室里会出现一种很微妙的放松：你看，AI 能做这件事。

我现在反而会在这一秒警惕。因为很多公司的误判，就是从这一秒开始的。Demo 活了。组织还没活。

Demo 能跑，只说明一段技术链路在一个被控制过的场景里暂时连起来了。输入是人准备好的。问题是人挑小的。边界是人提前收好的。

演示路径也被设计过。这当然有价值。但它还没有回答另一些更硬的问题：明天谁每天用？

谁复核输出？什么叫成功？出错谁升级？规则谁维护？

经验写回哪里？三个月后它还在不在真实流程里？老板如果只看见 Demo 成功，很容易以为公司已经进入 AI 改造。其实只是会议室里出现了一个好看的瞬间。

说得再具体一点。演示那天，输入的那份合同是最标准的一份，甲方条款干净，扫描件清晰，页码不缺。真实的合同不是这样：客户临时换了一版，附件是手机拍的，中间还夹着两页手写批注。演示时台上那个人是最懂这套系统的工程师；上线以后，坐在这个位置上的是刚入职三个月的合同专员。这中间的差距，不是模型能不能处理，是组织有没有为这个差距准备过任何东西。

组织改造从来不是一个瞬间。它是一套新工作方式闯进旧流程，跟旧责任、旧考核、旧知识库、旧审批习惯打架。

打赢了，才叫组织能力。没打过，只是一个被围观的功能。

Demo 的边界要说清楚。

Demo 证明的是：输入可以进去，模型可以处理，输出可以展示。这叫技术链路短暂成立。它不证明业务流程已经改变。它不证明员工愿意使用。

它不证明一线主管知道怎么验收。它不证明法务、财务、运营、销售这些角色知道自己在新流程里的位置。更不证明这个能力可以被复制、被审计、被维护、被新人接手。很多 AI 项目最容易混淆的，就是“功能能跑”和“组织能跑”。

功能能跑，是工程问题。组织能跑，是管理问题。工程问题问：接口通不通，延迟高不高，成本能不能接受，效果能不能达到测试集。管理问题问：它放进哪一步，改变谁的动作，谁来复核，错了谁改机制，哪些知识要沉淀，哪些权限要收住。

这两组问题都重要。但不是一组问题。如果公司只用工程问题验收 AI 项目，最后得到的常常是一个能演示、能截图、能写周报、但进不了日常经营的系统。它不是没用。

它是还没有被组织接住。

很多 POC 不是死在模型上。

POC 就是那个花三个月做的小规模试点——先在一条流程上试一试，看看行不行，再决定要不要全公司铺开。它死在组织链路上。反复出现的从来不是模型参数。是五个组织问题。第一，场景不真。会议室里选的是最顺的样例，真实现场里遇到的是脏数据、例外需求、客户临时变更、上游资料缺失。

第二，数据不稳。试点时有人手工整理输入，上线后没人持续维护数据，AI 输出自然开始漂。第三，责任不明。业务说这是技术项目，技术说这是业务需求，最后一线员工变成事实上的兜底人。

第四，验收不清。大家都说“效果还不错”，但没人能说清楚到底是速度变快了，错误变少了，返工变少了，收入变多了，还是风险降低了。第五，上线后无人维护。POC 时有人盯，汇报时有人讲，上线后规则变了、流程变了、员工绕开了系统，却没人负责更新。

所以一句话要钉住：技术链路能跑，不代表组织链路能跑。POC 真正要验证的东西只有一个：这个动作能不能进入组织的日常运行。如果不能，它就不是组织能力。

它只是一次技术展示。

第一道门，是有没有真实流程 owner。没有 owner，AI 项目就没有家。这里的 owner，不是提需求的人。提需求的人可能只是觉得某个环节慢、某类材料多、某个岗位忙。

他能描述痛点，但不一定为流程结果负责。真正的流程 owner，是上线以后每天要对这个流程结果负责的人。比如一个合同审核 AI，如果上线后合同风险漏了、业务审批慢了、法务复核堆了，谁要面对后果？是法务负责人、销售负责人、风控负责人，还是那个牵头做项目的人？

说不清，就不要急着扩大试点。因为 AI 一旦进入流程，就会改变权力和责任。它会改变谁先看到信息。谁先做判断。

谁有资格修改输出。谁能让流程继续往下走。这些东西如果不在上线前讲清楚，后面一定会用人情、习惯和甩锅来补。很多 Demo 最后变成展示品。不怪技术。是组织里没有一个人，愿意把它接回自己的流程。

这里有一个反过来的判断，值得管理者记住：AI 落地的关键，不是懂 AI 的人找到业务。是懂业务的人学会用 AI 改业务。让最懂 AI 的人去业务部门问“你们有什么场景能用 AI”，这是一场拿着锤子找钉子的运动，做出来的多半是 Demo。反过来，让最懂这条流程的业务骨干下场，他知道哪一步真的慢，哪一步真的贵，哪一步真的错不起——他才是那个天然的 owner。

一个没有 owner 的 AI 项目，最常见的命运是：大家都觉得有用。没人真正负责让它有用。这就是死局。所以你回去先做一件事：把最近这个 AI 项目的流程 owner 名字写出来，一个人名，不是一个部门名。写不出来，先别扩大试点。

第二道门，是验收指标。

AI 项目最怕一句话：效果还不错。听着友好，实际危险。不错在哪里？

速度快了多少？错误少了多少？返工少了多少？客户等待少了多少？

一线少查了多少资料？主管少做了多少重复复核？风险暴露提前了多少？如果答不上来，POC 就永远可以被解释成“还行”。

“还行”不是经营语言。老板需要的是经营语言：时间。质量。

错误率。返工率。收入。成本。

风险。满意度。技术团队可以用准确率、召回率、延迟、成本评估系统。但业务验收必须回到业务指标。

一个 AI 客服项目不能只说回答像不像人，要看升级工单有没有减少、首响有没有变快、错误承诺有没有下降。一个 AI 投放辅助项目不能只说建议看起来专业，要看异常发现是否提前、复盘是否更完整、下一轮规则是否真的被更新。验收指标不是为了给项目找麻烦。

它是为了防止一个项目永远停在“大家感觉不错”的状态里。感觉不错，不能进预算。下次听到“效果还不错”，你就追一句：“不错在哪个数上，改前是多少，改后是多少。”答不上来，这个项目就还停在感觉里。

第三道门，是复核和责任。

AI 输出错了以后，责任最容易溜走。供应商说模型只是建议。IT 说系统按设计运行。业务说自己只是试用。

一线员工说主管没讲清楚。最后组织会找一个最容易被拿出来的人背锅。这不是负责。这是背锅。

负责和背锅不是一件事。背锅是事情发生以后，被动承担后果。负责是事情发生之前，就有权提出风险、调整流程、要求复核、停止上线。如果一个人没有权力改变流程，却要对 AI 输出结果负责，那不是人在回路中。

那是人在锅里。所以 AI 项目上线前，必须把复核机制写清楚。哪些输出必须人工复核？哪些错误必须升级？

哪些场景必须保留人工兜底？每一次 AI 做了什么、谁下达指令、输出是什么、谁改过、谁批准，都要能追溯。留痕不是为了抓人。留痕是为了让组织能复盘、能修复、能继续用 AI。

判断一个人是真负责还是在背锅，看一件事就够：他有没有权力喊停。喊得停，是负责；喊不停还要担后果，那就是背锅。

第四道门，是知识沉淀。

很多 AI 项目做完以后，经验没有留在组织里。它留在供应商的交付文档里。留在项目经理的脑子里。留在某几个积极员工的个人账号里。

下一次再做一个类似项目，组织还是从头开始摸索。这个代价可以数出来：又是三个月的试点周期，又是两轮供应商比价，又是四五场跨部门对齐会，又是一茬新人从看不懂到会用。这些都发生过一次了，只是没人把它记下来。这说明什么？说明 Demo 没有变成组织记忆。AI 项目真正要沉淀的，不只是代码和提示词。

还包括：这个场景为什么值得做。哪些输入最容易出问题。哪些异常必须人工兜底。

哪些判断标准已经被验证。哪些失败模式下次不要再犯。如果这些东西没有被写进知识库、流程规范和复盘机制，这个项目就算上线了，也没有让组织变聪明。员工变快了。

组织没长记性。判断标准很简单：这个项目的经验，能不能在明年新人入职的第一周就交到他手上。交不出来，它就还在某个人的脑子里，不在组织里。

看一个脱敏的业务现场。

一个投放管理场景里，AI 看起来能做很多事。它可以帮团队找达人。可以批量整理内容数据。可以生成互动话术。

可以把复盘材料先做出初稿。如果只做 Demo，这个场景很好看。因为每一个动作都能展示“AI 很快”。但真正落地的时候，问题很快变成另一组：

达人筛选标准是谁定？AI 找出来的候选名单，谁判断合作风险？内容建议如果引发客户不满，谁负责？投放结果变好以后，经验是沉淀成下一轮规则，还是留在某个运营同学的个人提示词里？

如果一批自动化采集节点批量拿回内容，采集回来的东西怎么去重、怎么判质量、怎么转成下一步决策？这不是 AI 能不能跑的问题。这是组织能不能把 AI 跑出来的东西接成一条业务链路的问题。真正有价值的不是“AI 帮我们找了一堆达人”。

真正有价值的是：组织开始知道什么样的达人值得找。知道什么样的数据要排除。知道什么样的内容可以进入下一轮投放。

知道哪类异常要人工判断。知道下一次怎么更快复用这套规则。Demo 展示速度。落地考验记忆。

如果 AI 加速了前面几个动作，但后面的审批、复核、客户确认、预算分配仍然堵死，结果只是把等待提前了。前面跑得越快，后面堵得越明显。这就是很多 Demo 死掉的原因。不是它没让某个点变快。

是它没有让整条流程变顺。

Demo 把速度给了单点，组织要的是把顺序还给全链。

最有欺骗性的一种死法，是这样开始的。

大部分公司选试点场景的时候，天然会选流程最前端的那一段——内容生产、初稿撰写、信息整理、资料归集。原因很简单：这些环节最容易被 AI 接住，效果也最容易被看见。Demo 做出来一定好看。

但流程不是一个点，是一条链。前端提速以后，中间的审批、后面的发布、末端的交付如果还是旧速度，整条链的通过时间不会变。变的只是堵点的位置——它从前面挪到了后面。

对身处其中的人来说，这个变化甚至是负面的。前端的同事产出翻倍，中间的审批人收到的量也翻倍，但他的人手、工时和判断速度都没变。他会更累，会积压，会开始压批量、拖时间，或者干脆降低复核颗粒度。前端的人则会发现，自己快了没用，东西还是卡在同一个地方，只是这次卡得更久。

于是很容易得出一个错误结论：AI 没什么用。其实 AI 起作用了，只是组织把它放在了一个不改变全局的位置上。

所以 AI 进流程，从来不是“选一个节点接进去”就完事。它会逼着组织继续往后看：这一段快了以后，下一段要不要跟着改？审批的颗粒度要不要重定？复核的方式能不能也用 AI 辅助？发布和交付的节奏要不要重排？

这些问题在 Demo 阶段全都不会暴露，因为 Demo 只跑那一段。它们要等到上线以后、量真的起来了，才会一次性砸下来。

有一个判断可以帮管理者提前看到这件事：AI 不会让流程自动变快，它只会把这条流程的下一处堵点照出来。

照出来是好事。前提是有人愿意接着往下改，而不是在第一处堵点面前停下来，说这个 AI 好像也没什么用。

老板判断一个 Demo，要学会问更难听的问题。第一，这个 Demo 如果明天关掉，谁的日常工作会受影响？如果答案是“暂时没有”，它还没有进入流程。第二，这个 Demo 如果输出错了，谁会第一时间发现？

如果答案是“大家都会看一下”，它没有责任人。第三，这个 Demo 如果跑通三个月，组织会多出什么资产？如果答案是“一个系统”，不够。系统不是资产。

能复用的流程规则、判断标准、错误模式、责任表，才是资产。第四，这个 Demo 如果要复制到第二个团队，最难复制的是什么？如果最难复制的是某个关键人脑子里的经验，就说明知识还没有沉淀。第五，这个 Demo 的价值能不能进经营账？

进不了账，就很难进预算。进不了预算，就很难走出试点。这五个问题问完，很多 Demo 会立刻露出原形。它们不是没价值。

它们只是还没准备好进入组织。

所以 Demo 评审会不能再只让技术团队汇报。至少要让三类人同时在场。第一，流程 owner。他要回答：这个 Demo 如果进入流程，哪一步会被改，谁的日常动作会变化。

第二，复核责任人。他要回答：AI 输出错了以后，谁发现、谁修正、谁决定能不能继续往下走。第三，知识维护人。他要回答：这次试点产生的规则、错误、提示、边界，最后沉淀在哪里，谁维护。

三类人不在场，Demo 评审就会变成技术演示。技术演示可以继续。但不要把它叫落地评审。落地评审必须让组织上桌。

组织不上桌，Demo 越顺，误判越大。

第五道门，是上线后维护。

AI 项目不是上线那天结束。它是在上线那天才真正开始。业务规则会变。客户问题会变。

产品政策会变。知识库会过期。模型输出会漂。员工也会找到绕开系统的新路径。

如果没有人定期看这些变化，系统就会慢慢变成一个“曾经好用”的东西。很多公司不是没有 AI 系统。是有一堆没人维护、没人相信、没人负责的 AI 系统。最后大家会得出一个错误结论：AI 不适合我们。其实不是。是组织从来没给它配过维护机制。

所以上线评审那天就要定下来：这套东西多久看一次，谁看，看完谁改。定不下来，就当它三个月后会自己坏掉，因为它真的会。

所以下次评审之前，先画一张七列的表，横着摊在桌上。第一列：它替代或改变了哪一个真实流程节点？第二列：谁是这个流程 owner？第三列：验收指标是什么？

第四列：哪些动作可以在授权范围内自动执行，哪些例外需要人工处理？第五列：错误如何升级，后果如何恢复或承接？第六列：这次试点会沉淀哪些组织知识？第七列：上线后谁维护规则、数据和知识？

填写时将下面七项横排成七列，一条流程填一行。这里竖排列出填写口径，方便核查。

字段	应留下的内容
真实流程节点	AI 改变哪一步，输入和输出是什么
流程 owner	谁负责结果，拥有哪些实际权限
验收指标	对照什么基线，怎样判断运行结果
授权与控制	自动执行的条件、人工处理的例外及各自权限
异常与后果	如何限制、升级，怎样恢复或承接已发生的影响
知识沉淀	哪些规则、例外与证据留下，怎样核验更新
上线后维护	谁维护规则、数据和知识，什么变化触发复评

一张 A4 纸就能开始。让做项目的人当场填，填不出来的格子留空，继续补证据。填得满，也不等于技术适用、控制有效或真实运行已经过关。

这七列如果填不出来，Demo 再顺，也不能叫落地。它只能叫演示成功。真正的 AI 落地，不是让老板看见 AI 能做什么。是让组织回答：

这个东西明天由谁用，怎么用，错了谁管，价值怎么算，经验怎么留下。演示跑通了，不稀奇。组织活了，才稀奇。

CHAPTER 03

有钱没钱为什么都会 AI 转型失焦

我见过两种老板，坐下来说的第一句话完全相反。一个说：钱不是问题，方案给我，明年预算我批。另一个说：我们没那个预算，员工自己在用免费的，先跑着看。一年以后再问，两个人的答案却是同一句：好像也没什么变化。

有钱公司和没钱公司，AI 转型死法不一样。但病根经常是同一个。不是工具不够。是组织没接住。

有钱公司容易死在决策真空里，钱花得出去，但不知道 AI 上完以后组织怎么承接。没钱公司容易死在认知真空里，工具用起来了，但不知道 AI 有没有真正进入经营。

两边看起来差很多。一个有预算，一个没预算。一个能买系统，一个只能用低成本工具。一个 PPT 很厚，一个截图很多。

但如果最后都说不清：哪条流程变了、谁负责结果、产能去了哪里、知识有没有沉淀，那它们其实都停在同一个地方。停在旧组织方式里。

有钱公司的决策真空

有钱公司的问题，往往不是没动作。

恰恰相反，动作很多。规划做了，POC 跑了，系统选了，预算批了，外部团队也进来了。会议上看起来什么都准备好了。但三个月过去，什么都没发生。

不是模型不行。不是供应商不行。不是预算不够。是组织在真正要做选择的地方，选择了不选择。AI 一旦跑通，后面马上会出现一组问题：

原来做这件事的人去哪？这条流程谁来重写？释放出来的产能是降本、增产、提质，还是重组？被 AI 接走的判断，谁复核、谁负责？

这些问题没有一个是模型能回答的。它们都是组织决策。有钱公司最容易把技术验证当成转型进展。POC 成功了，以为事情已经过半。其实最难的部分刚开始。技术验证回答的是“能不能做”。

组织决策回答的是“做成以后，组织怎么变”。很多有钱公司卡住，是因为前一个问题有答案，后一个问题没人拍板。沉默不是中立。沉默是在消耗技术验证的有效期。

沉默不只发生在一号位。审批链上每一个不表态的人，都有他不表态的理由——而这些理由，通常不在人品上。

我跟进过一个项目。规划做完了，验证跑通了，报价也给了，推动这件事的管理者已经做了决策。但最终它不了了之。事后回看，不是任何一个具体的人站出来反对，是组织在旧路径和新路径之间做了一个保守选择——沉默了，然后时间把这件事消耗掉了。这不是那家公司特有的毛病，是一个结构性现象。

原因不在人品，在位置。每一次 AI 重写流程，都在同时重写依附于旧流程的人的位置。旧流程给了这些人三件东西：合法性、资源和身份。

合法性是指，他们在旧流程里做的判断和决定，被组织认可为有价值的贡献。资源是指，他们因为熟悉旧流程而拥有的信息优势、预算审批权和人员调配权。身份是指，旧流程给了他们一种“我是做这件事的人，这件事我懂”的自我认知。

一个干了十二年的主管，全公司没几个人说得清那张表为什么这么填，哪个客户的单子必须先过他手。开会时大家等他点头，新人有问题第一个找他。AI 把那张表接过去以后，表还在，等他点头的人没有了。他没被降职，也没被减薪，但他在这间会议室里的分量变了。AI 重写流程，等于把合法性、资源、身份这三件东西同时拿走，换成一套他不熟悉的新逻辑。在新逻辑里，他从专家变回了初学者。

所以阻力不是来自懒惰，也不是来自恶意。它来自旧位置给的这三件东西正在失去时的自然反应。这个判断很重要，因为它决定了管理者该怎么应对。如果阻力来自人的问题，那要做的是教育和说服；如果阻力来自结构问题，那要做的是在新流程里给这些人重新找到合法性、资源和身份的来源。前一种解法用在后一种问题上，怎么讲都讲不通。

这也解释了一件事：为什么高层拍了板、技术也交付了，事情还是走不动。因为“拥抱 AI”这条路动的不只是一个流程，是一批人的资源和身份。这个

问题没有被正面处理之前，再好的技术方案、再有力的高层背书，都会在这一层被消耗掉。技术验证的有效期有限，沉默却可以无限延长。

对 CEO 和 CHO 来说，这意味着推动 AI 转型不只是一个项目管理问题，还是一个组织设计问题。什么人会因为这次转型失去什么，怎么设计他们在新结构里的位置。这些问题如果不提前回答，审批链上的沉默，就是答案。

等到所有人都不再提，这个项目就自然死了。有钱公司的另一个常见现场，是产能被释放以后悬在半空。比如一类外部信息采集、数据整理、材料初审的工作，AI 接进去以后，确实能把大量重复劳动压掉。这时候真正的问题不是“AI 能不能做”。

真正的问题是：原来做这件事的人，下一步去哪里？如果这批产能没有去向，组织会进入一种很奇怪的状态。技术团队说效率提升了。业务团队说暂时还按原流程跑。

HR 不知道岗位要不要改。财务也不知道这笔效率该算成降本、增产还是提质。最后，AI 做成了，但组织没有得到结果。效率悬在空中，变成焦虑。

这就是有钱公司的决策真空。它不缺资源。它缺的是把效率翻译成经营选择的那个人。

这件事我做过。某集团上线一套外部信息采集的 AI 应用，上线之前，CEO 下面有一个专职办公室，每天的工作是人肉收集、整理、综合外部信息，然后给 CEO 出报告。这是一件消耗大量人力的重复性工作，做起来枯燥，但对决策很重要，全时投入的编制是八个人。AI 应用上线之后，这件事被系统接管了。于是那个问题就摆到了桌面上：这八个人去哪里？

这个问题的答案，模型给不了。算法也给不了。系统集成商给不了。甚至连负责推动这个项目的 CIO 也给不了。它不是一个技术问题，是一个关于组织资源怎么重新分配的决定。

这里的逻辑顺序不能倒。不是“这批人太贵了所以要上 AI 降本”，是“AI 已经能做这件事了，这批人的产能被释放出来，那么这批产能去哪里”。前者是成本驱动的被动替换，后者是产能释放之后的主动决策。两件事看起来相似，但对组织的要求完全不同。前者只需要一个“用人减少”的决定，后者需要整套产能承接的设计。

这个案例的答案是：八个人转岗，去做其他事了，没有一个人被裁。

承接方案，是比技术交付更早开始设计的事情。

这个结果之所以能发生，不是因为系统设计得多完善，也不是因为这批人特别能适应变化。是因为在应用上线之前，CEO 那一侧就已经想好了：如果这个系统跑起来，这批人去哪里，做什么，谁来带。

很多企业走的正好相反：先把系统跑通，再回头想人的安置。结果是技术准备好了，组织没准备好；或者更常见的，技术这边宣告成功，人那边的焦虑开始累积。

正确的顺序应该反过来。在技术方案还在纸上的时候，就同步问一句：如果这件事真的被 AI 接管了，原来做这件事的人去哪里。这个问题谁负责回答。什么时候回答。以什么为依据。

在我的判断里，这个问题归根结底需要 CEO 那一侧给答案。不是 CIO，不是 HR，不是项目经理，不是外部顾问。因为承接决策本质上是资源再分配决策：这批人力被释放了，投入哪里，服务哪个新方向，谁来管理，绩效口径怎么重写。

这几件事没有一件能在一个部门里办完。八个人调去哪个业务线，那个业务线的负责人认不认这批编制。他们的考核算谁的，年底奖金从哪个池子出。谁给他们做新岗位的培训，培训期间产出算不算数。HR 可以设计转岗路径，CIO 可以给技术接口，但“这批产能往哪里投”的决定，只能由能同时调动这几个部门的人来做。

AI 接管了一件事，不等于那批人就有了去处。去哪里是组织设计问题，不是技术问题。转岗这件事不会自动发生，它需要有人做决定。

这个案例之所以值得讲，是因为它是少数真的往下走了一步的。更常见的形态，是往下那一步始终没有迈出去。

我也接触过一家制造企业，里面有几百名做图纸设计的人员，我们为它设计过解决方案。方案做了，后续怎么实施、人员怎么安排，是另外一件事。不能拿方案写完，代替组织已经接住。

还有一笔账容易算错：AI 承担了多少工作，不能直接换算成人员该怎样安排。

看 Klarna 的公开披露。2024 年 2 月 27 日，公司称 AI 客服助手上线首月处理了两百三十万次对话，占其客服聊天的三分之二，工作量折合七百名全职客

服。在 2025 年 3 月 14 日提交的上市注册文件（F-1）中，公司披露 AI 助手上线以来处理了 62% 的客服聊天，工作量折合八百多名全职客服；这个折算依据的是助手上线后，人工处理的聊天和电话对话的平均月减少量[①]。

两次披露的时点和统计口径不同，不能把七百到八百多直接读成“前后改口”，更不能读成实际少了这么多人。两份资料能确认公司披露了什么，不能单凭它们判定内部有没有把人员、质量和投入的账算清楚。

回到自己的公司，至少把四件事分开说：AI 承担了多少工作；人员实际去了哪里；服务质量怎样；维护、复核和补救花了多少。工作量折算回答不了后面三个问题。一句“做了七百人的工作”，不是“可以少七百个人”的决策依据。

再把问题落到要作的决定上。如果方案已有，技术验证也已完成，人员和能力如何承接却迟迟没有决定，卡住的就不只是技术。这种“决策真空”，不是不知道有哪些办法，而是知道了，也没动。

为什么知道了也不动？因为在这类决定上，不动的成本是隐性的，动的成本是显性的。

裁或转，明天就要付账。人的情绪要面对。部门之间的博弈要摆平。编制和预算要重排。这些账，下个月的会上就会有人找你算。

不动的账不一样。设计能力没有升级，单位产出没有变化，这批人的技能停在原地。这些代价分摊在未来几年里，没有任何一个季度会因为它而难看。

理性的个人在这种结构里，会选择不动。所以这不是某个老板的性格问题，是激励结构的问题。

真正要改的，是让不动也有成本、也有人负责、也要在某张表上被记一笔。

①2024 年首月数字见 Klarna，〈Klarna AI assistant handles two-thirds of customer service chats in its first month〉，2024 年 2 月 27 日，公司公告（https://www.klarna.com/international/press/klarna-ai-assistant-handles-two-thirds-of-customer-service-chats-in-its-first-month/）；62%、八百多名全职客服的等价工作量及其估算方法，见 Klarna Group plc 于 2025 年 3 月 14 日提交的 F-1（https://www.sec.gov/Archives/edgar/data/2003292/000162828025012824/klarnagroupplcf-1.htm），以“62%”及“800 full-time agents”定位相关说明。两者均为公司披露，等价工作量不是实际裁员人数，也不等于独立验证的组织成效。

对 CEO 来说，这里有一个可以立刻做的动作。把公司里所有已经做完方案、但至今没有产生组织动作的 AI 项目列出来。一个一个问同一句话：这件事现在卡在谁那里，下一个动作是什么，什么时候做。

这张表通常不长，但它往往就是公司 AI 转型真正的进度条。比任何一份规划都准。

没钱公司的认知真空

没钱公司是另一种死法。

员工天天在工作群里发 AI 截图。有人用 AI 写竞品分析，有人用 AI 润色提案，有人用 AI 半小时整理完数据报告。老板看见以后，一开始会觉得高兴。但过一阵子，焦虑来了。为什么员工都在用，经营结果看不出来？

流程还是慢。客户还是等。材料还是反复改。老板看见的是员工“在用”，看不见的是 AI 有没有进账本。

这就是认知真空。不是不懂 AI 工具。是没有坐标系判断 AI 有没有进入经营。没钱公司最容易用三个假指标安慰自己。

第一，看截图数量。第二，看培训完成率。第三，看员工用了没有。这三个指标都不够。

截图证明工具被打开了，不证明流程变了。培训完成率证明课上完了，不证明工作方式变了。员工用了 AI，不证明组织能力变强了。没钱公司真正需要的，不一定是先买系统。

先需要一套账本。

三张账

没钱公司可以先从三张账开始。

时间账。等待账。知识账。

时间账问的是：员工用 AI 节省的时间，去了哪里？以前两天做完的东西，现在半天做完。听起来很好。但那一天半如果没有进入新的产出节点，就只是个人轻松了，不是经营变强了。省下来的时间，要么变成多接的订单，要么变成原来没人做的那件事，要么就什么都不是。

等待账问的是：流程里的空转时间有没有变少？人等系统，系统等人，审批等确认，交付等反馈。这些等待加起来，往往比真正干活的时间还长。一份报价单，真正写只要两小时，从客户问到客户收到却过了四天，中间三天多都在等人有空看一眼。AI 进来以后，如果只是前面那两小时变成二十分钟，后面三天照堵，客户感觉不到任何变化。

知识账问的是：员工用 AI 做出的判断，有没有留在组织里？个人账号里的历史记录，不是组织知识。群聊里的截图，不是组织记忆。员工脑子里记住的用法，也不是，因为人会走，记忆会带走。

一次 AI 辅助判断，只有被写回流程、模板、知识库和复盘机制，才算进入组织。

这三张账，不需要一开始就很精密。但老板必须每个月能问出一点数字。哪类任务省了时间？哪段等待缩短了？哪些判断被沉淀了？如果这些格子一直是空的，说明 AI 还没进入经营。

没钱公司最小的切口，不是“做一个 agent”，是找一条高频、痛点明确、结果可验的流程。agent 就是那种能自己跑完一串活、不用人一步步喂指令的 AI 小助手。比如报价初审、内容初稿、客服问题分类、合同格式检查、数据整理、周报生成。这类节点有几个共同点：输入相对稳定，输出能被验收，等待时间看得见，返工和错误也看得见。

老板不需要一上来做大系统。先让团队用一个低成本工具，在这个节点上跑一个月。跑完只看三件事：时间有没有省下来，等待有没有缩短，判断有没有沉淀。

如果三张账都不动，说明这个切口不对，或者 AI 没有进入真实流程。如果三张账开始动，哪怕只是小幅度动，公司就有了第一条真实证据。这比一百张 AI 截图更值钱。

有钱没钱都容易误判

有钱公司把预算当能力。

没钱公司把预算当借口。这两种判断都错。

有钱不是答案。预算会放大正确动作，也会放大错误动作。如果组织没有承接机制，预算越大，动静越大，浪费也越大。系统上线以后，流程不改，责任不动，产能不入账，最后只是把旧组织的问题包装得更贵。

没钱也不是借口。预算少，确实不能一上来做重系统、重集成、重平台。但它可以选一条真实流程，找一个高频节点，用低成本工具跑一次时间账、等待账、知识账。

没钱公司的优势，反而是没那么容易被系统绑架。因为没有大系统可以依赖，它必须从真实流程里找切口。如果老板有账本意识，这反而是好事。它会逼组织先问：哪条流程真的要改？

而不是先问：买哪个系统？

老板误判清单

把两类公司的误判放到一张表里，差异和共同病根会更清楚。

误判	典型表现	真问题
有钱公司误判	预算批了，系统上了，POC 跑了	产能释放以后没人拍板，组织承接机制缺位
没钱公司误判	员工在用 AI，截图很多，培训完成了	AI 没有进入时间账、等待账、知识账
共同误判	把工具动作当成组织转型	没有选真实流程，没有重写责任和结果口径

老板要问的不是“我们有没有钱做 AI”。是：我们有没有选中一条真实流程？这条流程有没有 owner？

AI 进去以后，时间、等待、知识哪张账会动？产能释放以后，谁决定去向？结果怎么验收？这些问题答不上来，有钱没钱都一样。

最后都会失焦。

明天怎么开这个会

这章不需要开一场战略务虚会。

开一场 60 分钟的小会就够。参会的人不需要多：一号位、业务 owner、CHO 或 HRBP、CIO 或技术负责人，再加一个正在真实使用 AI 的一线员工。

会上只看一条流程。不要泛泛讨论“公司 AI 转型怎么做”，就问这条流程。第一步，把流程从触发到交付画出来。第二步，标出 AI 已经进入或可以进入的节点。第三步，填三张账：时间账、等待账、知识账。第四步，定一个 owner。第五步，约定一个月后看什么结果。

这场会开完，至少要留下三样东西：一张流程图，一张三账表，一个下月复盘时间。如果连这三样东西都留不下来，就不要说公司在做 AI 转型，最多只能说，公司里有人在用 AI。

这两句话差得很远。前者是组织动作，后者只是个人行为。老板要为前者负责，而不能拿后者安慰自己。有钱没钱不是分水岭，这条线才是。

PART 02 · 第二部

岗位和工作被重写

老板最容易问的一句话是：

AI 会替代哪些岗位？

这句话很正常，但它太粗。岗位是组织管理上的一个单位，不是 AI 改造工作的最小单位。一个岗位里，有任务，有判断，有责任，有关系，有异常处理，也有结果承担。AI 进来以后，不会平均地替代一个岗位，它会先接走某些工作片段，再逼着组织重新定义这个岗位的价值。

如果公司只问“裁不裁这个岗位”，就会错过更关键的问题：

- 哪些任务已经不值得人这样做；
- 哪些判断必须由人保留；
- 哪些责任不能交给 AI；
- 哪些结果要重新衡量；
- 哪些经验要沉淀成组织资产。

岗位重写不是在岗位说明书——HR 嘴里那份 JD——里加一句“熟悉 AI 工具”，而是把人和 AI 重新编进同一条工作链。所以这一部要拆的，是那个被问得最多、也答得最粗的问题：

AI 上来以后，岗位到底是被替代，还是被重写？

CHAPTER 04

AI 替代的不是岗位，是任务、判断和流程片段

一个老板看完 AI 十分钟写完一份市场报告，回过头问我的第一句话是：“这么说，我那三个做报告的，是不是不用要了？”

这个问题很自然。但它太粗。

粗问题进不了组织决策。它只会带来两件事：管理层焦虑，员工防御。老板拿着一个没有答案的问题到处问，员工拿着“我会不会被替代”的不安开始藏着掖着。最后 AI 转型还没开始，组织里的信任先被消耗了一轮。

真正该问的不是“AI 会不会替代人”。真正该问的是：

> AI 先替代这条流程里的哪个工作片段？

这两个问题，听起来只差一点，后果完全不同。

第一个问题只有两个答案：是，或者不是。两个答案都很粗，都导不出下一步动作。

第二个问题会逼着管理者往下拆：是某个重复任务？某个格式化输出？某个标准判断？某个跨部门节点？某个客户沟通动作？拆到这里，组织设计才开始。

很多公司买了 AI 工具，开了账号，做了培训，三个月后老板开会问：“你们有没有好好用？”

员工不知道怎么答，因为没人先把一件事说清楚：AI 到底进入哪一个真实工作片段？这个片段原来谁做？现在谁定义输入？谁看输出？谁复核？谁对结果负责？

这些问题没人答，AI 就停在账号里。钱花了。组织没动。

她没做错任何事，是岗位被重新定义了

前面那位老板问“是不是不用要了”——这个问题，我自己真实地面对过一次。

我有一位高级咨询师。

她跟客户开会，把需求一条条问清楚，再用 PPT 把方案讲出来。这套活她做了大半年——手绘的图，逻辑清楚，配色讲究。我心里其实是欣赏的。

但我自己是怎么做同一件事的？我跟客户谈完，转身跟 Claude 把方案讨论清楚——哪几个选项、优先排序、风险在哪——再用 Obsidian 知识库把 storyline 梳出来，最后一句话让 Claude 把图和文字组装出来。整个过程我一个人在电脑前，没开一次会，没改一次配色。

又快，又稳，外观也不差。

原本一周的事，不到一杯咖啡的工夫。

她的手绘图逻辑清楚、配色讲究，这些都是真本事，只是这套本事所依附的那个岗位，已经不需要一周了。

我看着她，心里是惋惜的。**她没做错任何事，是岗位被我重新定义了。** AI 工具把执行过程极限压缩以后，岗位原来那条“从需求到方案”的长链条，尾巴没了。剩下能留给她做的事，她做不来；我也没法凭空造一件事出来让她做。

我不是没试过——工具给了，内部分享做了很多次，也手把手教过。后来才想明白：不是工具用不用得起来的问题。

岗位被重新定义这件事，只有老板亲自把话说清楚，工具替代不了。

后来她自己另谋高就，我们体面地分了手。

这件事教会我两件事：**以后招人，先想清楚这个岗位三个月后还在不在；以后看见一个员工的工作被悄悄改写，别装作没看见。** 前者是经营，后者是体面。

岗位不是最小单位

岗位是什么？

岗位不是工作的本质。岗位是组织为了管理方便，把一堆事情打包成一个名字。往正面说，岗位的本质是组织为了让某类结果持续发生，设的一个具名责任接口。

“运营专员”“内容编辑”“数据分析师”“HRBP”，每个名字背后都装着很多性质不同的工作。有些是重复任务，有些是经验判断，有些是关系协调，有些是责任兜底。

组织把它们打包成岗位，是为了能招人、填人、考核、交接、追责。这套方式在 AI 之前是合理的，但 AI 进来以后，问题变了。

AI 改造的不是打包后的岗位名，是打包前的那些零件。它先碰到的是岗位里的某些任务，某些判断，某些流程节点，不是整个岗位。

管理层最容易犯的推理错误，是看到 AI 能完成某一类输出，就直接跳到“这个岗位可以被替代”。这中间不只跳过了工作片段，还跳过了更关键的一层：片段只是拆解单位，不是委托单位。

一件事能不能交给 AI、交给人，或者交给一组人机协作的小队，要看它有没有形成一条可委托工作闭环。

它至少要回答八个问题。

第一组问的是活本身：目标是什么，上下文从哪里来。第二组问的是谁来干、拿什么干：谁执行，调用什么工具和资料。第三组问的是边界和验收：权限边界在哪里，结果怎么评估。第四组问的是出事以后：错误怎么反馈回系统，最终谁对结果负责。

缺任何一项，这个片段都只是“看起来可以交出去”。

比如写周报。AI 能写初稿，不等于周报工作已经被委托出去。谁要用这份周报？上下文来自哪些系统？哪些数据不能进去？谁判断结论是否准确？老板看完以后提的问题，谁写回下一版规则？如果周报误导了决策，谁修机制、谁担责任？

这些问题不清楚，AI 接走的只是动作，不是工作。

组织升级的关键，是把岗位里的动作重组为可以委托、可以评估、可以追责、可以更新的工作闭环。做到这一步，AI 才开始进入组织 OS。

所以别急着盯岗位名。先挑一件你们每天都在做、又最想交给 AI 的事，拿上面八个问题挨个问一遍。答不上来的那几个，就是这件事现在还交不出去的原因。

四类工作片段

一个岗位里，至少有四类工作片段。

第一类是**任务**。任务指重复出现、有规则可循、输入和输出相对稳定的工作，比如生成初稿、汇总数据、整理会议纪要、分类标注、标准化报告。

任务层是 AI 最容易进入的地方。只要输入清楚、格式稳定、质量标准可验证，AI 就能接走一部分重复执行。

人的角色也会变：从重复执行者，变成规则定义者和质量抽检者。

第二类是**判断**。判断指风险识别、优先级排序、例外处理，以及那些必须结合上下文做取舍的决策。

AI 可以给建议，可以列风险信号，可以做候选排序。但最终判断不能直接让渡。因为判断一旦出错，后果要有人承担。

AI 能说“这个候选方案看起来更优”。但它不能替组织说：“这个决定我负责。”

第三类是**关系**。关系指跨部门协调、客户信任、上下游资源对齐、冲突处理。

AI 可以帮你准备材料，整理纪要，生成沟通草稿。但真实关系不是靠一封措辞得体的邮件建立的。客户的不满、部门之间的利益冲突、老板和团队之间的信任，最后都要回到人和人之间。

这不是 AI 会不会写话术的问题。这是组织信任怎么运行的问题。

第四类是**责任**。责任指拍板、复核、验收、兜底，就是那些出了问题以后，别人会问“这是谁定的”“谁验收的”“谁负责”的节点。

AI 可以留痕，可以辅助复盘，可以生成记录。但责任本身是组织属性，也是人的属性。AI 不能成为组织里的责任主体。

把这四类片段分清楚，管理者才能看见 AI 的真实位置。

合同审核就是一个例子。格式核查是任务层，风险条款识别是判断层，和对方法务沟通是关系层，最终签字盖章是责任层。

如果你只看“合同审核”这个岗位，就会误以为 AI 要么能替代，要么不能替代。

拆开以后才知道：有些能交，有些能辅，有些必须人来。

回去挑一个你最想动的岗位，在纸上画四行：任务、判断、关系、责任。每行写一件这个人本周真做过的事。四行都写满，你手里就有了一张能上会讨论的东西，不是一句“这个岗位是不是不用要了”。

AI 先进入哪里

AI 最先进入任务层。

格式化文档、标准摘要、规则分类、重复数据处理，这些是当下最容易被 AI 接住的工作。前提是组织把输入规范、输出格式和质量标准说清楚。

这里有一个冷刀：

> 如果你说不清楚输入和验收标准，AI 不稳定，可能不是 AI 的问题，是你这项工作原来就没有被组织说清楚。

AI 会把很多旧问题照出来。以前人靠经验和忍耐把模糊流程跑过去，AI 一进来，模糊的输入、模糊的标准、模糊的责任，往往会全部暴露出来。

判断层不能这么简单。判断层的正确用法是：AI 给建议，人看建议，人做决定，人承担结果。

AI 能列出风险，但不知道今天组织里哪个风险最不能碰。AI 能做排序，但不知道某个客户关系背后的历史。AI 能发现异常，但不知道这个异常现在该升级，还是先观察两天。

关系层和责任层，更不能让 AI 独立承担。一个重要客户的不满，不能靠 AI 自动发一封漂亮邮件解决。一个员工去留判断，不能因为 AI 打了一个分就直接拍板。一个对外承诺，不能让 AI 说完以后组织再被迫买单。

所以 AI 转型不是把每个岗位都变成 Agent——Agent 就是那种你交代一句、它自己跑完一串动作的 AI 助手——更不是万物 Agent。正确做法是把规则、工具、模型和人放回正确的位置。稳定任务交给自动化和 AI。复杂判断交给 AI 辅助。人保留决策权和责任权。关系和兜底节点必须有人在场。

四层里最容易出事的是第二层。今天回去问一句：我们现在有没有哪个决定，是因为 AI 给了个分数就直接过了？

有，那就是判断权已经不知不觉交出去了。

任务被替代以后，人去哪

任务被 AI 接走以后，管理者最容易产生一个误判：这件事不需要人了。

错。

任务被替代，和人被替代，是两件事。

任务层被接走以后，人至少会出现四个新位置。

第一是**监督者**。AI 产出以后，人要识别偏差，发现风险，判断这个结果能不能进入下一个流程节点。这个角色并不低级。它要求人真正懂业务质量，否则根本看不出“看起来对、实际上错”的输出。

第二是**判断者**。AI 给出建议以后，人要结合组织语境、客户关系、资源约束和历史背景，做最后判断。判断者不是照单全收 AI，是在 AI 建议和现实情境之间做取舍。

第三是**训练者**。AI 出错以后，例外情况、错误模式、修正动作要写回系统。否则 AI 永远停在初始质量上，组织也只是越来越依赖一个工具，而没有形成自己的能力。

第四是**系统负责人**。有人要为整个人机协作流程负责，不是为一次输出负责，是为这个流程的长期质量、知识更新、异常处理和业务结果负责。

这四个位置不是降级。这是角色升级。

升级长什么样?

以前这个人一天做八份报表，现在他一天看八份 AI 报表。听起来像没变，差别在于他现在必须看得出哪一份是错的，还要说清楚错在哪、下次怎么不错。

工作被这样重排，不是只在我们这几家公司发生。麻省理工工业绩效中心 2026 年 4 月发了一份报告，叫《Humans in the Loop》，作者是该中心执行主任 Ben Armstrong 和 MIT 航空航天系主任 Julie Shah[①]。他们看的正是企业在生成式 AI 早期实验里，工作怎么在人和机器之间被重新分配：搜索、起草、汇总、判断、复核、沟通这些动作被拆开，又重新组合。这还只是早期实验里看到的，不是大规模铺开以后的定论。但方向已经很清楚：先动的是任务，跟着动的是人的位置。

但升级有条件。组织必须明确这些位置，给到权限、工具和绩效口径。否则任务被 AI 接走，人没有被放到新位置上，工作就会变成真空地带：AI 产出没人负责，错误没人写回，系统没人维护。

---

①Ben Armstrong、Julie Shah，《Humans in the Loop: The evolution of work in early experiments with generative AI》，麻省理工学院工业绩效中心（MIT Industrial Performance Center），2026 年 4 月 8 日发布。Julie Shah 的航空航天系主任职务自 2024 年 5 月 1 日起生效，见 MIT News 2024 年 4 月 29 日公告。

组织看起来少了一点工作，其实多了一个没人负责的黑洞。

所以你们已经用起来的那个 AI 工具，回去对着这四个位置点一遍名：谁在监督，谁在判断，谁把错误写回去，谁为整条流程负责。四个位置有一个空着，那件事现在就没人管。

一个混合岗位怎么拆

看一个脱敏的混合岗位。表面上，它是一个运营类岗位，每天要看数据、做报告、跟业务沟通、处理异常、对客户解释结果。拆开以后，则是性质不同的四层。

任务层：每日数据汇总、报告生成、状态核查、素材整理、历史记录归档。这些工作有稳定输入和输出，可以优先让 AI 处理。

判断层：方案评估、异常识别、优先级排序、目标不达标时判断先改哪里。AI 可以给候选建议，但最后要人结合业务语境判断。

关系层：和客户对齐目标，和内部业务方协调节奏，异常发生后做解释。这一层 AI 可以准备材料，但不能替人建立信任。

责任层：最终方案谁批，异常升级谁定，出了问题谁解释，谁承担后果。这一层必须有人。

拆完以后，结论不是“这个岗位可以不要了”，是这个岗位的任务清单变了。人不该继续把大量时间花在任务层，而要被放到判断、关系和责任层。

这才是工作片段拆解的意义：它证明的从来不是人没用了。

是人应该从哪里撤出来，又应该被放到哪里去。

岗位重写不是改 JD

很多组织讲岗位重写，最后会落到一个很轻的动作：把岗位说明书——也就是 HR 嘴里那份 JD——改一版，加一句“熟练使用 AI 工具”，再加一句“具备人机协同能力”。这不叫岗位重写。这叫把旧岗位贴上一张新标签。

真正的岗位重写，至少要改四件事。

第一，改交付结果。以前这个岗位交付的是“做完多少事”。AI 进入以后，这个岗位应该交付的是“拿回什么结果”。比如以前看一个运营是不是按时产出日

报、周报、复盘，现在要看他能不能让一条流程的输出更稳定、异常更少、响应更快、知识沉淀更多。

过程指标不是不能看，但它不能继续做唯一口径；AI 时代还只考工作量，会把人逼回旧流程。

第二，改能力要求。以前岗位能力常常写成技能清单：会什么系统，会什么工具，会什么方法。AI 进入以后，最关键的能力不是会不会点工具。

关键的能力不是点工具，是另外三件事。能不能判断工具输出是否靠谱。能不能发现流程哪里坏了。能不能把一次经验变成下一次系统可以复用的规则。

这就是判断力。

不会判断的人，AI 给得越多，他越容易变成橡皮图章。

第三，改责任边界。如果 AI 参与了这个岗位的工作，谁下指令，AI 做了什么，输出是什么，谁复核，谁验收，谁对结果负责，这些都要写清楚。

否则岗位说明书再漂亮，出了问题还是互相推。

业务说这是系统问题，IT 说这是业务用法问题，HR 说这是岗位能力问题。最后组织里没人真正负责，只有一个离事故最近的人被拎出来背锅。

第四，改知识沉淀。一个岗位如果开始大量使用 AI，但所有提示词、判断依据、修正动作、异常处理都留在个人账号里，这个岗位就没有真的被重写。

员工变快了。组织没变聪明。

真正的岗位重写，要让这个岗位的好判断能够沉下来。人走了，经验不至于全走；新人来，不至于从零开始；下一条流程要改，不至于重新摸一遍。

所以，岗位重写不是 HR 写一版新 JD。

岗位重写是组织重新回答四个问题：这个岗位交付什么结果，需要什么判断，承担什么责任，留下什么组织记忆。

这四个问题答不出来，就先不要急着谈“AI 时代岗位升级”。

回去把你们最近改过的那份岗位说明书翻出来，对着这四个问题读一遍。如果通篇只改了能力要求那一栏，其余三栏一个字没动，那这次改版就还没开始。

什么时候才有资格讨论编制

做到这里，老板和 CHO 还是会问：那到底什么时候可以动编制？可以问，但要有前置条件。我建议至少四个条件同时成立，才进入编制讨论。

第一，任务替代已经稳定。 不是 Demo 看起来能用，是在真实负载、真实输入、真实异常里跑过一段时间，输出质量可以批量依赖。

第二，责任链已经重写。 谁使用 AI，谁复核结果，谁验收输出，谁在出错时兜底，这些问题都有具名答案。

第三，知识已经沉淀。 AI 处理过程中的例外、错误、修正动作和判断规则，已经写回组织知识系统，不只留在某个人的账号或脑子里。

第四，产能方向已经定义。 AI 释放出来的时间和人力，到底流向哪里？是降本、增产、提质，还是重组到新业务？如果没有方向，所谓提效只是把生产问题变成人员问题。

四个条件少一条，都不该急着动编制。

任务不稳定就动，系统会翻车。责任链没重写就动，出事没人兜。知识没沉淀就动，能力跟着人走。产能方向没定义就动，账面省了，组织变薄了。

所以下次开会有人提编制，先把这四条摆到桌上，一条一条问“做到了没有”。四个都能点头，再谈人数。

没有证明组织结果稳定之前，裁掉的不是成本，可能是组织能力。

给老板的工作片段分析表

讨论任何岗位调整前，先填这张表。

层级	老板要问的问题
任务层	这个岗位哪些工作重复、格式化、输入输出稳定？AI 已经在哪些任务上跑过？稳定性如何？
判断层	这个岗位哪些判断依赖经验、语境和例外处理？AI 只能辅助到哪一步？
关系层	这个岗位维护了哪些客户、部门、上下游信任关系？AI 能准备材料，哪里必须人出面？
责任层	这个岗位在哪些节点拍板、复核、验收、兜底？AI 进入以后，责任有没有重新指向某个人？
剩余价值	任务被 AI 接走以后，这个人最值钱的判断、关系、责任片段还剩什么？组织还需要吗？
编制前置条件	任务稳定了吗？责任链重写了吗？知识沉淀了吗？产能方向定义了吗？

填不出来，就不要急着裁。填得出来，也不代表一定裁。它只是说明，你终于从情绪和口号，进入了组织设计。

这张表在会上怎么用？不要让 HR 一个人填。这张表应该由业务 owner、CHO、CIO 或 AI 转型负责人一起填，因为每一层都不是单部门能回答的。

任务层，业务最清楚每天到底在做什么。判断层，老员工和一线负责人最清楚哪些地方靠经验。关系层，业务和管理者最清楚哪些信任不能丢。责任层，CEO 和组织负责人必须拍板：出了问题谁兜底。

如果这张表只由 HR 填，它会变成岗位分析表。如果只由 IT 填，它会变成系统功能表。如果只由老板拍脑袋填，它会变成裁员理由表。

只有把业务、组织、系统和责任放到同一张桌子上，它才会变成 AI 时代的岗位重写工具。

所以这周先别谈人数。挑一个岗位，把这四个人叫到一间屋里，用一个小时把上面这张表填完。填完你就知道，这个岗位到底是该减人，还是该换位置。

岗位拆开以后，后面才有可能继续讨论三件事：工作经验怎么沉淀成组织资产，AI 参与以后责任链怎么画，释放出来的产能到底怎么分配。

本章只解决第一步。先把岗位拆开——这个岗位三个月后还在不在，先问过一遍。

拆不开岗位的公司，谈的从来不是 AI 转型，是给裁员换一个体面的名字。

CHAPTER 05

岗位不是唯一分析单位

很多老板谈 AI 裁员，第一句话就问错了：“这个岗位能不能被 AI 替代？”

这句话听起来很直接，其实太粗了。粗到没法做决策。

一个岗位里，混着太多东西。有人在查表，有人在整理材料，有人在写初稿，有人在判断风险，有人在处理关系，有人在最后签字。它们都被装进同一个岗位名称里，叫运营、客服、HRBP、财务、法务、投放、产品经理。

你可以现在就试一下。把你们公司一个采购专员一天的活写在纸上：录入订单、比价、催货期、盘账对单、判断这家供应商还能不能继续用、决定这批货要不要验退。六件事写完你就会发现，AI 能接的和不能接的，混在同一个岗位名称底下。你问“采购这个岗能不能被替代”，等于问“这六件事能不能被替代”——一个问题里塞了六个答案，当然答不出来。

AI 进来以后，真正被改变的不是这个岗位名称，是里面那些更小的工作片段。查资料、整理数据、生成初稿、标准化复核、异常识别，这些动作很容易先被 AI 接走。因为它们有稳定输入，也有相对稳定输出。做错了，通常还能查、能改、能重来。

但客户承诺、候选人判断、组织风险、价格赔付、重大交付、人员评价，这些东西不能简单交出去。不是因为 AI 一定做不了。是结果一旦发生，组织要买单。

这里才是第一刀：岗位不是最小分析单位，任务、判断、责任、结果，才是。

你可能觉得这四个词在上一章见过。上一章画四行，是为了看清一个岗位内部有哪几类活；这里换四项，是因为岗位这把尺本身该换掉了。两处不冲突，一处是拆，一处是量。

岗位这个单位还能往下拆，拆开以后才看得清哪一块能交、哪一块不能交。所以谈 AI 替代的时候，岗位就不该是你手里唯一的那把尺——它太粗，量不出这些差别。

如果一号位还停在“这个岗位裁不裁”这个问题上，他就很容易做出两种低级动作。

一种是把 AI 当成人力替代器：看到某个岗位里有一部分工作能自动化，就直接推导出这个岗位可以减少人。这个推导太快，中间少了好几层账：哪些任务被接走了？人的判断还在不在？客户和组织结果有没有变稳定？出了事谁兜底？

另一种是把 AI 当成技能标签。岗位说明书里加一句“熟练使用 AI 工具”，培训课上了，员工也真开始用了。但这个岗位交付什么、怎么交付、谁对结果负责，没人核验这些旧安排还能不能接住新的工作量和错误。

这两种动作表面相反，本质一样：都跳过了岗位这笔账，一边急着减人，一边只管加工具。

旧岗位继续按旧逻辑跑，AI 在旁边加速。旧安排经验证仍能接住，就可以继续用；承接能力没核过，谁也不能把加速直接算成组织升级。

过去很多慢，是流程自己在消化风险。现在前面快了，后面的判断、复核、责任和治理没跟上，风险只是被更快地送到了下一个节点。

我现在看很多企业的 AI 转型，最危险的地方就在这里。

老板以为自己在做组织升级，其实只是在给旧岗位装马达。

原来一个人一天处理二十条，现在 AI 帮他处理八十条。看起来产能上去了，但如果后面的复核能力已经不够、异常升级接不住、客户承诺越了界，组织就是把更大流量送进了一根已经超负荷的旧管道。管道本来够用，不必为了上 AI 换掉；已经接不住还继续加压，爆的时候会更难看。真到那一天，模型错了查模型，岗位接不住查岗位，别把两笔账混成一句“AI 不靠谱”。

先把岗位打开

所以岗位重写的第一步，不是改 JD，是把岗位打开。一个能用的岗位重写，至少要问四个问题。

第一，这个岗位真正交付的结果是什么？不是“负责某某工作”，也不是“协助完成某某事项”。这些话写在岗位说明书里很好看，但拿不到结果。真正要问的是：这个岗位存在，是为了让组织拿到什么稳定结果？

客服岗位不是为了“回复客户”，是为了让客户问题被准确处理、情绪被接住、承诺不乱给、体验不掉线。招聘岗位不是为了“筛简历”，是为了让合适的人进入合适的位置，并且这件事能在后续结果里被复盘。投放岗位不是为了“做报表”，是为了让预算花出去以后，组织能拿回更稳定、更可解释的增长结果。

岗位一旦从动作回到结果，AI 才能被放到正确位置。

第二，AI 接走的是哪类任务？这里不能靠感觉。要把岗位里的工作拆出来，看哪些是重复执行、信息整理、初稿生成、标准化判断、复核辅助、异常识别。这些通常是 AI 优先进入的地方。

员工自己也知道。很多人最烦的不是有难度的活。是那些每天都要做、做完也没人觉得有价值、但不做又会堵住流程的活。

我把这类工作叫切黄油。切黄油不是骂人，它只是说，这类工作动作稳定、结果可查、判断含量低，过去靠人硬切，是因为没有更好的工具。现在有了 AI，就应该让 AI 先进去。

这个词是拆岗位拆出来的，不是想出来的。

我拆过一个投放运营岗。拆完发现，这个人的时间大致可以分成两种性质完全不同的工作。

第一种，是查表、整理数据、汇总报表这类活。他每周要查好几张媒体价格表，对比不同渠道的历史投放效果，把散落在各个平台后台的数据拉回来合并成一张汇总表，再整理成汇报用的格式。这些动作很规律，步骤可以复现，做完的产出是信息，不是判断。

第二种，是他真正的专业所在：人和人之间的沟通与链接。

和媒体代表谈采购条件，对方在电话里说的不只是价格。还有这个季度哪个位置流量变了，什么节点有资源可以拿。这些话靠的是几年攒下来的交情，不是发邮件能问出来的。

给客户汇报投放进展，也不只是念数字。客户说一句“我觉得效果一般”，他得当场判断：这是客户没看懂数据，还是当初期望就没对齐。判断完了，才知道下一句话该怎么说。

判断这个月某个渠道要不要加码，依赖的是他对市场节奏的感知，不是表格里能查出来的规律。

这些是他的真任务。关系、判断、信任，这类片段不重复，不能被标准化，出错了也没有简单的“重做”选项。

拆完之后，问题就不一样了。原来的问题是：这个投放运营岗要不要被 AI 替代？这个问题本身无解，因为它既没有正确答案，也没有操作出口。新的问题是：查表和整理报表这块切黄油，今天有哪些工具可以接管？接管之后，这个人每周能省多少时间？省出来的时间，能不能更多地放在维护媒体关系和研判投放节奏上？

这才是有操作价值的问题。切黄油交给工具，人的时间重新分配给真任务，岗位不消失，但工作的重心会迁移。

需要说明的是，这个拆解是方法论示范，不是统计结论。不同公司的投放运营岗，切黄油和真任务的比例会有差异，甚至差异很大。拆解的意义不在于得出一个固定比例，在于让管理者看清这个岗位里有哪两类性质不同的工作，再决定下一步怎么设计。

这个岗位的说明书上，写的都不是这些事。这就是为什么岗位名称回答不了 AI 的问题——名称写的是应该做什么，时间花在哪里才是实际在做什么。

两者之间的差额，往往就是 AI 最先该进去的地方。

但切黄油被接走以后，不能马上跳到“那这个人是不是不用了”。更重要的问题是：这个人的时间被释放以后，组织准备让他去做什么？

如果没有答案，AI 提效就只是把空白时间制造出来。

第三，人保留什么判断？这一步最容易被忽略。很多流程里都有“人工确认”。看起来人在回路中，其实人只是点了一个确认按钮。AI 给出建议，人看一眼，点通过，一天点几百次。出了事，系统记录上有他的名字。

这不是人在负责，是人在背书。

真正的人类判断，至少要有三个条件：他有独立信息，他有推翻 AI 的权力，他推翻以后不会被流程惩罚。

三个条件缺一个，这个人就点不动那个“不通过”。没有独立信息，他手上除了 AI 给的那份材料什么也没有，只能信。没有推翻权力，他点了不通过，系统还是照原样往下走，那点了也白点。推翻要付代价——按一次不通过就得写说明、拉三个人开会、还可能被追问“你凭什么觉得机器错了”——那他第二次就学乖了，闭着眼点通过。

这时候所谓“人工监督”，就变成了组织把风险转移给一个最弱的节点。他没有信息、没有权力、还要担名字。

岗位重写必须把人的判断写清楚。哪些判断不能交给 AI？哪些判断可以由 AI 提建议，但必须由人裁决？哪些判断由 AI 先筛，人只处理异常？这些不是技术配置问题，是岗位设计问题。

第四，责任边界怎么改？AI 参与工作以后，岗位的边界不能只按动作划分，还要按结果追踪。

谁下达指令？AI 做了什么？输出是什么？谁复核？谁批准？结果出了问题谁启动纠偏？这些问题不写清楚，岗位重写就是半截工程。

拿一单赔付试试。客户要赔八千，系统按历史同类案子建议赔三千，坐席照着发了出去，客户第二天投诉到总经理信箱。现在把那六个问号各填一个名字。这单是谁让 AI 算的？AI 用的是哪批数据？它给出的三千是建议还是结论？发出去之前有没有人看过？超过多少钱必须主管签字？投诉进来以后谁有权把这单推翻重算？六个空里只要有两个填不上，这个岗位的责任边界就还没写完。

填得上的公司，第二天在复盘会上讨论的是那批数据该不该改；填不上的公司，讨论的是那个坐席该不该罚。

系统负责人这个角色，就是在这里出现的。人不是站在 AI 旁边看它干活的人，是对人机系统结果负责的人。

系统负责人不是技术管理员，也不是供应商对接人。他要负责这套人机系统的边界、运行质量、异常处理和结果反馈。系统什么时候可以自动跑，什么时候必须人工介入，什么时候必须停下来复盘，这些都要有人说得清楚。

用车间的话说，他是这条线的线长。线长不自己拧螺丝，但这条线出的东西合不合格，是他的事。哪台设备该停，哪个环节要加人复检，出了批次问题谁去查——这些他必须当场答得上来。AI 进了流程，就等于车间里多了一条不睡觉的线。这条线也得有线长。

如果一个岗位原来负责“完成任务”，AI 进来以后，它可能要变成“让一套人机系统稳定拿回结果”。这不是加一个工具技能，是岗位性质变了。

所以，岗位重写不是一句“熟悉 AI”，而是把岗位重写四账逐项核清：交付结果、AI 任务、人类判断、责任边界。

四账都要查，缺口在哪就改哪。原安排经验证仍能稳定拿回结果，就保留，不为证明转型硬改。连这笔账都没查清，AI 才只是旧岗位上的新外挂。

用一小时重写一个岗位

老板下周就可以拿一张纸开会。选一个最想改的岗位，别问这个岗位能不能裁，先把岗位重写四问写上白板。

四个问题答不出来，不要急着谈裁员，也不要急着谈培训。

因为你连岗位到底被 AI 改成了什么，都还没看清楚。

这个工具真正适合放在一次一小时的管理会上。一号位不用先讨论公司全员怎么学 AI，先挑一个高频岗位，让业务负责人、CHO、CIO 和这个岗位的直接管理者坐在一起，把四问写在白板上。

第一轮只写事实，不写愿望。这个岗位现在交付什么结果？每天实际在做什么？AI 已经接走了什么？员工自己最烦的动作是什么？哪几个输出一旦错了，客户、组织或者财务会真的买单？

第二轮再决定哪些要改、哪些保留。哪些动作交给 AI？哪些判断必须留给人？哪些异常必须升级？现在的绩效口径能不能反映实际结果，哪里还需要调整？

会议开完，如果只多了一句“这个岗位要提升 AI 能力”，这会白开。人机责任表要写清哪些改、哪些保留、依据是什么；已有的表能回答，就沿用，不另造一张。

工具：岗位重写四问

问题	要看什么	输出物
交付结果是什么？	岗位为组织拿回什么稳定结果	结果定义
AI 接走哪类任务？	重复执行、信息整理、初稿生成、标准化判断、复核辅助、异常识别	AI 介入清单
人保留什么判断？	逐项看哪些判断保留、哪些获准委托；无需逐次人工判断也要写明依据	判断保留或委托清单
责任边界怎么改？	谁指令、谁复核、谁批准、谁纠偏、谁对结果负责	人机责任表

这张表这周就能用。先挑一个岗位——挑那个员工抱怨最多、你自己也说不清他一天到底在忙什么的岗位。把这四行抄到白板上，让干这个活的人自己填，填完你再看。

四行都得有回答。“无需逐次人工判断”可以是答案，但依据、授权范围和异常怎么处理得写明；空着不等于已经决定交给 AI。该留什么、能交什么还说不清，就先别谈裁员，也先别谈培训。

岗位不是唯一分析单位。只有把岗位打开，写清交付结果、AI 任务、人类判断和责任边界，组织才不是给旧岗位装马达，是在重写一套能稳定拿回结果的人机系统。

CHAPTER 06

岗位怎么重写，才不是加一句“熟悉 AI”

很多组织做 AI 岗位改造，第一反应是改 JD。把岗位说明书打开。在能力要求后面加一句：

> 熟悉主流 AI 工具，能在日常工作中灵活运用。

HR 看起来动了。

系统里也更新了。老板问：“岗位改完了吗？”回答：“改完了。”三个月以后，实际工作方式没有变，绩效口径没有变，责任边界没有变，岗位上的人只是多了一项模糊的新要求。

这不是岗位重写。这是贴标签。AI 真正进入岗位以后，变化不是“员工多用一个工具”。它会接走一部分动作，改变一部分判断路径，重画一部分复核和追责关系。

但组织还在用旧 JD、旧绩效、旧责任边界管理这个人。人已经按一套新方式工作。组织还按一套旧方式评价。裂缝就在这里。

少做了什么，是效率问题。重新负责什么，是组织问题。

有一类岗位最能看出这道裂缝：专门写报告的初级分析岗。以前它的活很清楚——收数据、整理、出一份格式规整的报告。假如这段活 AI 半小时就能给你一版，能不能由此推出这个岗位没了？先别急。Indeed 的招聘实验室在 2026 年 1 月报告中指出，截至 2025 年 12 月，美国数据与分析类招聘广告里，约百分之四十五提到了 AI 相关词。世界经济论坛《未来就业报告 2025》的雇主调查则预计，到 2030 年，文员、数据录入等岗位将收缩，大数据专家等技术岗位将

增长[①]。一个是招聘广告的观察，一个是雇主的未来预期；它们不能保证某个初级岗位安全，也不能替我们证明“分析岗全没了”。

所以管理者先要拆开的，不是“这个岗位留不留”，是它里面的活变了没有。标准化出报告那部分如果被压缩，验证和判断就得有人接住。名字没变，考核没变，人却可能已经在干另一件事。一个岗位的名字还在，不代表它还是原来那个岗位。

岗位重写不是改职责描述

岗位重写，第一步不是改职责描述。

是问这个岗位到底对什么结果负责。很多 JD 写得很完整，但写的都是动作。负责内容选题、策划和发布。负责招聘渠道维护、简历筛选和面试安排。

负责业务沟通、人才盘点和组织诊断。这些看起来像职责，其实大多是动作清单。AI 一进来，动作会被改写。AI 可以生成选题，可以整理简历，可以做候选人风险提示，可以汇总业务数据，可以帮 HRBP 预判某个组织单元的异常信号。

如果岗位还只用动作定义，这个岗位会立刻变得尴尬：动作被接走以后，人到底还负责什么？这就是岗位重写的起点。

三道坎

岗位重写通常会卡在三道坎。

第一道坎，是这个岗到底对什么结果负责——英文叫 job understanding，直译过来就是把这个岗看明白。这件事听起来基础，但真正问下去，很多岗位答不上来。内容运营岗到底负责发多少篇内容，还是负责让某类内容带来有效线索？

---

①Cory Stahle，Indeed Hiring Lab，《January 2026 US Labor Market Update: Jobs Mentioning AI Are Growing Amid Broader Hiring Weakness》，2026 年 1 月 22 日；45% 对应截至 2025 年 12 月美国数据与分析类招聘广告中 AI 相关词的占比，不是岗位增长率。原文：hiringlab.indeed.com/2026/01/22/january-labor-market-update-jobs-mentioning-ai-are-growing-amid-broader-hiring-weakness/。World Economic Forum，《The Future of Jobs Report 2025》，2025 年 1 月 7 日，第 2 章 Jobs outlook；所引为受访雇主对 2030 年岗位变化的预期，不是已实现的就业结果。原文：weforum.org/publications/the-future-of-jobs-report-2025/in-full/2-jobs-outlook/。两页核查日期：2026 年 9 月 6 日。

招聘岗到底负责筛多少份简历，还是负责让人岗匹配判断被后续表现验证？HRBP 到底负责按时交人才盘点表，还是负责让业务团队提前看见组织风险？这些答案不一样，岗位重写的方向就完全不一样。第二道坎，是责任边界。

AI 做了输出，人采纳了建议，最后结果出了问题，谁负责？如果这条线不画清，岗位上的人会变成一种很奇怪的存在：既不像独立决策者，也不像真正复核者，只是一个夹在 AI 和结果之间的人。他看起来参与了。但没有真正负责。

第三道坎，是绩效口径。旧绩效经常考过程：做了多少、响应了多少、产出了多少。AI 压缩过程以后，继续考过程，就会逼员工假装有工作量。新绩效要更多看结果：结果是否更稳定，返工是否更少，判断是否可追踪，经验是否可复用。

这三道坎有顺序。先说清这个岗对什么结果负责。再画清人和 AI 的责任边界。最后才改绩效口径。

顺序反了，岗位重写会变成表格装修。

拿你们公司最近改过的那个岗位对一下：这三件事，先动的是哪一件？如果先动的是绩效表，那基本可以断定，前面两道坎还没过。

三种最常见的假重写

第一种，是只改 JD。

原来写“熟练使用 Office”，现在改成“熟练使用 AI 工具”。看起来跟上时代。其实没有回答任何组织问题。这个岗位对什么结果负责，没有变；人和 AI 怎么分工，没有变；出了错谁负责，也没有变。

第二种，是只做培训。组织安排一轮 AI 课程，员工学会了提问、生成、总结、做图。这当然有用，但上完课还不能证明他在真实任务里会用，更不能单凭培训推出岗位已经改好。

一个员工学会了 AI，不等于一个岗位被重写。

第三种，是只看降本。AI 接走了一部分动作，老板立刻问能不能少几个人。这个问题不是不能问。

但如果组织还没证明结果更稳定、责任能兜住、知识能沉淀，先把人拿掉，拿掉的可能不是成本，是组织能力。岗位重写不是“AI 能干，所以人少一点”。岗位重写是“AI 接走动作以后，人要对更高一层的结果负责”。这句话如果立不住，后面谈裁员、转岗、增产，都会飘。

培训上了，岗位为什么还是没变？AMO 提供了一把拆开问的尺子：能力、动机、机会。Yu 等人关于 AI 采用与员工福祉的综述，用它讨论培训、评价回报、岗位设计和参与安排[①]。

落到一个岗位上，问法很具体：给他一份真实任务，他能不能识别 AI 的局限、判断输出、处理异常？贡献方法以后，他得到什么回报、承担什么风险？工具虽然开通了，必要的信息、时间、权限和新任务有没有一起给到？这些条件可能互相影响，不能把人简单分成“不会”“不愿”“没机会”三类。培训后面，还得接工作安排。

岗位重写七栏表

把岗位重写落地，需要一张表。

我把它叫岗位重写七栏表——名字不重要，你完全可以叫它别的[②]。四账回答的是这个岗位该怎么拆，七栏要求的是拆完之后逐格写下来。所以它比四账多两栏：新绩效口径和知识沉淀。这两栏在纸上想不出来，只能填出来。它不是标准答案。它是逼组织把问题说清楚。第一栏，原岗位交付结果。

这个岗位在 AI 进入前，真正对哪个业务结果负责？标准是什么？第二栏，原任务清单。为了完成这个结果，原来需要做哪些动作：信息收集、判断、沟通、执行、复核。第三栏，AI 可接走的动作。

哪些动作已经能被 AI 完成，或者被 AI 大幅提速？第四栏，人保留的判断。哪些判断需要由人决定，依据是什么；哪些可在明确授权下委托执行？若

①Yu, S., Kawasaki, S., Magni, F., Yu, K. Y. T., & Munusamy, V. P. (2026). “Optimizing AI adoption through HR practices: Systematic review and theoretical framework for improved employee well-being.” Human Resource Management Review, 36, 101152. DOI: 10.1016/j.hrmr.2026.101152。这里引用第 2.4 节及第 5 节的 AMO 诊断与设计视角；AMO 并非该文首创。作者在后续研究部分仍要求检验所提干预的效果及组合关系，本书不将其当作岗位重写七栏表的效度证明。

②此表在作者工作底稿中编号 V15。

没有需要逐次人工决定的事项，也要写明范围和验证依据，不能凭空凑数。第五栏，人机协作后的责任边界。

AI 输出由谁采纳，谁复核，谁最终对结果负责？第六栏，新绩效口径。这个岗位以后不该只考动作量，而要考哪些结果、质量、复盘和知识贡献？第七栏，组织语境 / 知识沉淀。

这个岗位掌握了哪些公司内部的隐性知识？AI 介入以后，这些知识怎么被显性化、沉淀、复用？第七栏很重要。尤其是垂直领域，或者强依赖公司历史、客户关系、业务规则的岗位。如果没有这一栏，AI 接走的只是表层动作。

真正值钱的组织判断，还是留在人脑里。人一走，知识也走。

一个岗位真正改完的样子

拿招聘岗举例。

旧招聘岗是一串动作：发岗位、筛简历、约面试、推进 offer、跟进入职。AI 进来以后，简历结构化、经历整理、风险提示、面试题生成，这些动作大半会被接走。只看到这里，就会以为招聘岗被“自动化”了。

真正难的不在这一层。真正难的是：这个人适不适合这家公司、这个团队、这个老板、这个阶段。AI 能提醒履历里的异常，但这还没有回答团队信任、未来半年的岗位变化和与业务负责人的协作问题。这些判断需要什么证据，谁获准作出，错了如何纠正，都要逐项核清，不能凭一份履历分析直接放行。

所以招聘岗改完以后，不能只对“招了多少人”负责。它要对“人岗匹配判断有没有被后续表现验证”负责：三个月后能不能稳定交付，一年后留下了哪些可复用的判断标准，都要有人回头看。

再看 HRBP。开会、做表、盘点、写汇报，这些弹药 AI 都能提前备好。但真正改完的 HRBP，不是读 AI 报告的人，他要把信号翻译成组织判断：这是能力问题还是流程问题，是敬业度下滑还是组织契约被某次改造伤到了。判断做完，还要推动业务 owner 做动作，并把依据和结果写回组织记忆。

改完之后不是表更快了，是组织更早看见风险，也更稳地处理风险。这才叫岗位重写。

先填第三栏和第四栏

实际使用这张七栏表，不一定从第一栏开始。

更有效的入口，是先填第三栏和第四栏。AI 已经接走了什么？哪些判断由人保留，哪些可以委托，凭什么？这两栏一碰，岗位定义里最含糊的地方就会露出来。

不要先争岗位名。先把动作和判断分开。第三栏写得很满，第四栏却没有依据，要继续查。若不需要逐次人工决定，具名责任、授权范围、检查和处置安排仍要写清并经过验证；需要人工判断，也不能用“业务把关”四个字交差。

好的第四栏要落到具体判断：这类输出能不能进入下一个流程节点？这个风险要不要升级？由人决定还是按已验证规则执行，分别写明依据。

这个人能不能放进这个团队？这个客户承诺能不能说出口？这些问题写清以后，岗位才从动作岗位，变成结果岗位。

人机分工的责任边界

岗位重写里，最难的不是任务分配。

是判断归属。AI 能做什么，和组织允许它决定什么，是两件事。任务做得更快、更准，不能替它自动取得决策权；责任落了名字，也不等于每个动作都必须由人按下去。

涉及组织关系、人的处境和团队结构，先把具体决定拆出来：规则是谁定的，AI 获准执行到哪一步，哪些例外要交回有权的人，受影响者怎样提出异议。责任要具名，更要有办法履行。

拿招聘来说，整理材料和决定录用，不是一回事。文化匹配、信任潜力这些词，也不能直接变成一个自动淘汰分数。判断依据、适用边界和后果处置没有核清，就不能让输出直接生效；明确规则内的动作能否自动执行，要另查授权和控制，不能仅凭它属于 HR 就一概判定。

HRBP 或用人主管若只是跟着 AI 点头，却没有依据、权限和时间检查，签了字也不等于尽了责。要追的是：输出凭什么被采用，靠什么发现错误，出了问题谁能处理，这个判断后来对不对。AI 时代最危险的人机分工，不是 AI 做得太多。

是责任写在纸上，出了问题却没有人接得住。

能力模型要跟着变

一个岗位被 AI 改写以后，先核能力模型是否还对应它现在的责任。原来要求的执行速度、流程熟练度和检索能力，不会因为有了 AI 就一律失效；哪些工作真的变了，要拿实际任务对照。

新增要求也不能只写“判断力”“复核能力”。拿一份有缺陷的输出，看他能否发现问题、说明依据，并知道什么时候该升级；换一份没见过的任务，再看这套判断能不能迁移。这里看的是完成工作的证据，不是提示词写得漂亮。

招聘、培训和评价都要跟着这个核对走。模型、任务或授权发生变化，就回头看原先的能力要求是否仍够用，旧技能哪些要保留，哪些需要补。不能一边要求员工理解系统，一边不给他了解系统和练习的机会。

这个方向不只是我的观察。麦肯锡全球研究院 2024 年那份关于工作未来的研究给了一组数字：到 2030 年，社会情感类技能的需求，欧洲会涨百分之十一，美国会涨百分之十四。他们 2025 年那份报告里还有一句更值得管理者看的话——今天的技能里，超过七成在自动化和非自动化的工作场景里都用得上[①]。也就是说，技能不是在消失，是在换个地方用。

能力模型是否要改，要看实际任务和责任是否已经变化、原来的要求是否仍然适用。确有失配却继续照旧评价，才会把员工推向那些容易被计数、却未必有助于新结果的动作。

所以你回去可以先做一件小事：把现有能力模型和一份真实任务摆在一起。哪些考核项仍能解释结果，哪些只记录了旧动作，哪些新责任还没有对应的训练和评价？先把这几处对不上找出来。

最难跨的那道坎叫系统思维

新能力清单列出来以后，还有一个问题没有回答：这几样里，哪一样最难？

按我的观察，最难跨的不是判断力，也不是提示词水平，是系统思维。从任务执行者到系统负责人，最难的不是学会用 AI。是从“我把这一步做完”，切到“我知道这个系统哪里坏了”。

①麦肯锡全球研究院，《A New Future of Work: The Race to Deploy AI and Raise Skills in Europe and Beyond》，2024 年 5 月；《Agents, Robots, and Us: Skill Partnerships in the Age of AI》，2025 年。据官方页及媒体引文，第二份报告原文为 more than 70 percent。

这个断点有两个很具体的表现形态。

第一种，人只会点确认，不知道怎么改规则。他能看 AI 的输出，能判断这一条对不对，遇到错的也能改过来。但你问他这类错误为什么会反复出现、规则该怎么调整才能让它不再发生，他没有答案，因为他从来没有把自己放在规则那一层想过问题。

第二种，人只会用 AI 出结果，不会看流程哪里坏了。任务交给他，他能用 AI 更快地交出去，效率确实提升了。但流程上游给他的输入一直有问题、下游一直要为他返工、某个异常每周都出现一次——这些他都看得见，却不会把它当成自己该处理的事。

这两类人都不是不努力，也不是能力差，他们只是还停在任务这一层。任务层的人问的是：这件事做完了吗。系统层的人问的是：这件事下次还需要我做吗。

差别就在这一句。

而这道坎难跨，还有一个组织侧的原因，这个原因通常被归到员工身上，其实是管理的责任。

CHO 推动岗位重写时，要先弄清员工遇到的是哪一种困难：不知道工作会怎么变，不知道要补什么能力，还是担心组织借 AI 作一次没有讲明的替代安排。同样的观望，不能直接诊断成不愿意改变。

把方向讲清是一步，把条件给到是另一步。让人反馈流程问题，就要问他看不看得到上下游、有没有时间查、提出意见后谁处理；让人交经验，也要回答经验怎么用、贡献怎么被认可。

如果 JD、能力模型和培训都动了，实际工作还没变，先查这些安排是否落地。不要只凭一个“不积极”，就替员工把原因说完。

把抽象名词换成能问出口的问题

判断力、业务理解、组织语境理解，这几个词写在能力模型上没有错。但它们和第四栏那句“综合判断”“业务把关”是同一种毛病：听起来都对，谁也没法拿它去评价一个人。抽象名词落不了地，不是因为它错，是因为它没有被翻译成一个可以在具体岗位上问出口的问题。

翻译的办法，是先去看这个岗位的人一天到底在干什么，再看这些时间里哪一段是他真正该负责的判断。

我聊过客服岗。普通客服的大部分时间，其实不是在应对客户提出的问题，是在找材料。翻历史订单，翻产品说明，翻过去同类问题是怎么答的，翻这个客户上次投诉过什么。等材料凑齐，客户已经等了很久。而真正需要他判断的那几件事，反而只剩下很短的时间去想：这个客户现在的情绪要不要先处理，这个承诺能不能给，这单要不要升级。

另一个例子是负责效果优化的运营岗。这个岗位看起来是给甲方提供价值，但他有大量时间是在处理杂七杂八的核对：这份信息对不对得上，哪个环节漏了，某份材料到底发出去没有。这些动作一件都不能少，可它们里面几乎没有判断，只有不能出错的执行。

这两个岗位放在一起，说明了同一件事：JD 上写的那件事，和这个人实际耗掉时间的那件事，本来就不是一件事。AI 进来以后，被接走的恰恰是耗时间那一半——找材料、核对、整理、比对。所以岗位重写真正的问题不是“他会不会用 AI”，是他省下来的那半天，有没有换成一个更值钱的判断。

顺着这条线，抽象名词就能被钉成问句。判断力，在客服岗上是：材料齐了以后，他能不能判断这个客户该安抚、该补偿，还是该升级？业务理解，在那个效果优化岗上是：核对的活被接走以后，他能不能看出这个甲方这个季度真正在意的是什么，并据此提出调整建议？组织语境理解，在这两个岗位上都是：他知不知道哪类承诺公司兜不住、哪个客户过去出过什么事、哪条规则是写着的、哪条规则是没写但不能碰的？

这三个问题都有一个共同点：能拿到具体的人身上问，能被后来的结果验证，也能被写进绩效口径。写不到这个程度的能力模型，就还停在名词上。

这张表进会议室的正确打开方式

这张七栏表不应该是 HR 部门自己关门填的表，它应该进老板会。但进会也不能摊开一张大表从第一栏讲到第七栏，老板会最怕这种东西——看起来专业，听完没人拍板。

更有效的用法是只选三个岗位：一个离收入近，一个离客户近，一个离组织能力近，比如销售支持、客服、HRBP。每个岗位只讲四件事：原来对什么结

果负责，AI 已经接走哪些动作，哪些判断由人保留或获准委托，绩效口径和责任边界是否需要调整、依据在哪。这样讲，老板能听懂，业务负责人也躲不开。因为讨论的不再是“员工要不要学 AI”，是“这个岗位以后到底怎么拿结果”。会上只能说出“以后多用 AI”，这张岗位账就还没算清。

老板真正要看的不是一张漂亮模板，是组织有没有把 AI 释放出来的产能，重新接到结果、责任和知识上。

从客服岗那半天说起

回到前面那个客服岗。材料被 AI 接走以后，他省下来的时间不会自动变成判断，那半天到底换成了什么，要有人在会上说清楚。

所以下一次开会，最不值钱的问题是“岗位 JD 更新了吗”。这个问题太轻，回答起来太容易。

前面讲过的岗位重写四账，是交付结果、AI 任务、人类判断、责任边界。到了落地会，这四账还要进一步对到四个落点：结果有没有改变，人机分工有没有说清，责任有没有归位，经验有没有沉淀为组织资产。前者是分析账目，后者是会议验收，不是另一套同名“四问”。

这四个落点逐一对上，很多所谓的岗位重写会露出底——岗位没改，标签改了；能力没升级，工具要求加了。

真正的岗位重写，不是让每个人都更会用 AI。是让每个岗位重新说清楚四账：我对什么结果负责，哪些任务交给 AI，哪些判断由人保留、哪些获准委托，责任边界落到谁。至于能不能持续复用，还要看这些判断和反馈有没有写回组织资产。

至于这周能动的，其实只有很小的三件事。

先选三个受 AI 影响最深的岗位，别全公司铺开。招聘、客服、内容、运营、财务审核、销售支持里，挑最明显的那三个。

再用这张七栏表先填第三栏和第四栏，回答 AI 接走了什么、判断由谁作出、保留或委托的依据是什么。填不出依据的地方，就是这次要查的缺口。

最后改一条绩效口径。不要重写整套绩效制度，先把一个岗位从“动作量”换成“结果质量 + 判断复盘 + 知识沉淀”。

岗位重写从来不是文书工作。它是组织重新定义人和 AI 怎么共同负责。

PART 03 · 第三部

知识和数据怎么变成组织能力

员工会用 AI，是好事，但员工会用 AI，不等于组织变聪明。

很多公司接下来会遇到一个很尴尬的场景：每个人都在自己的账号里用 AI，写得更快，查得更快，总结得更快。可这些好输出、好方法、好判断，都留在个人电脑、个人账号、聊天记录和临时文档里。

这个人离职，context 一起走；这个人换工具，经验又清零；另一个部门遇到同样问题，还要重新问一遍。这就是 AI 生产力悖论：个人效率在涨，组织能力不累加。

知识库也不是把文档扔进去就完事。真正难的是哪些知识能被调用，谁负责更新，哪些内容过期，哪些内容敏感，AI 调用以后有没有留痕，员工贡献出来的判断有没有被组织认可。

组织语境是 AI 时代的地基。地基不沉下来，AI 越强，组织越散。这一部要回答的，就是那道横在个人和组织之间的坎：

> 员工个人的 AI 能力，怎样变成组织能力？

CHAPTER 07

员工 know-how 如何变成组织资产

预算批了，工具买了，账号开到每个人头上，培训也做了两轮。

三个月以后，老板在会上问了一句：那业务为什么没变快？

这个场面在上了 AI 的公司里反复出现，荒诞，但很常见。

表面看，是员工不会用 AI。再往下看，不是。

真正的问题是：AI 不知道这家公司是怎么工作的。

它不知道退款审批里有哪些例外。不知道老销售为什么对某类客户特别谨慎。不知道招聘那边的人看到一个候选人的履历，为什么会皱一下眉。也不知道运营负责人为什么在某个时间点，就是不敢加投放预算。

这些东西不在系统里，不在流程文档里，也不在知识库里。它们在人脑里，在聊天记录里，在老员工一句“这个事以前出过问题”里。

员工用了 AI，个人可能变快；但这些判断没有沉下来，组织不会变聪明。

这就是 AI 时代的知识断点。

不是没有知识库，是没有组织记忆

很多老板一听知识断点，第一反应是：那就建知识库。

这个反应太轻了。知识库通常装的是文档、制度、流程、FAQ、历史材料，解决的是“东西在哪里”。

但组织记忆解决的是另一个问题：

> 这家公司以前怎么判断、怎么犯错、怎么修正、怎么在下一次避免同样的问题？

要回答这个问题，靠的不是文档堆积，是判断沉淀。

YC（Y Combinator，硅谷那家孵化了 Airbnb、Stripe 的创业营，每年公布一份“我们想投什么”的清单）在 2026 年的创业方向清单里，把一个方向叫

Company Brain——公司大脑[①]。讲的是把企业里散落的 domain knowledge，也就是这家公司专属的业务门道，抽出来、理清楚、持续更新，让 AI 能读懂公司到底怎么运转。这个判断很有启发。但放到一家开了二十年的传统公司里，还要再往前问一步：

你公司里的这些门道，员工愿不愿意交出来？

员工不交，AI 读不到。员工只交表面材料，AI 学不到。员工走了，经验跟着一起走。每个人都只把 AI 用在自己账号里，组织拿到的就不是组织记忆，是一堆孤岛。

每个员工都有外挂，公司却没有大脑。

所以先别急着批知识库预算。先想清楚：你要的是一个装文档的柜子，还是一套记得住判断的系统。这两件事的成本、周期、阻力完全不一样。

know-how 不是文档

说 know-how，先别把它神秘化。它就是“这个人知道怎么做”，但这个“怎么做”，通常说不清。

Polanyi（波兰尼，最早把“隐性知识”这个词讲清楚的哲学家）讲隐性知识时有一个核心意思：人知道的东西，常常多于他能说出来的东西[②]。他举的例子是认脸——你能从一百万张脸里认出一个人，却说不出你是怎么认出来的。日本学者 Nonaka 和 Takeuchi（野中郁次郎和竹内弘高，研究日本企业为什么能持续出新产品）后来讲组织知识创造，也把隐性知识和显性知识的转换放在核心位置[③]。OECD（经合组织，发达国家的政策研究机构）在讨论知识经济时，还区分过 know-what、know-why、know-how、know-who 这几类知识[④]。

这几类知识落到公司里，就是三句话。知道制度叫什么，是 know-what。知道为什么这么设计，是 know-why。知道遇到具体情况该怎么处理，是 know-how。AI 最容易读到前两类，最难读到第三类。

---

①Y Combinator，《Request for Startups》（Summer 2026），https://www.ycombinator.com/rfs，“Company Brain”条目。

②Michael Polanyi，《The Tacit Dimension》，University of Chicago Press，1966 年，第 4 页。

③Ikujiro Nonaka、Hirotaka Takeuchi，《The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation》，Oxford University Press，1995 年。

④OECD，《The Knowledge-Based Economy》（文号 OCDE/GD(96)102），1996 年，know-what / know-why / know-how / know-who 四分类章节。

比如招聘。流程可以写成文档，面试题可以放进系统，岗位要求可以列成表格。但一个做了十年的 HR，看一眼就知道这个候选人跟团队气质合不合。他能看出这个人是不是短期漂亮、长期有风险。他还知道要人的那个业务老板，现在是真缺人，还是只是焦虑。这几件事，都很难写成标准答案。

再比如客服，系统里能查到订单、价格、售后政策。但一个老客服知道某个高价值客户现在最在意的是效率还是情绪，知道这个投诉该硬按规则还是先稳住关系，这些也不是一条 FAQ 能装下的。

know-how 的难点不在“知道”，在“情境”。离开情境，经验就失真。

这就是为什么很多企业做知识库，最后越做越像资料库。资料越来越多，真正能帮新人做判断的东西没多少。

有人会问：这么难提，那能不能干脆别提？

不能。技术博主 Daniel Miessler（长期写 AI 与安全的独立作者，读者以工程师为主）在 2024 年写过一篇文章[①]，把这件事推到了极致。他说公司说到底就是一堆算法的集合，不管产品多特别、公司多独特，它运作起来还是一条一条的步骤管道。他还有一句更狠的：可解释性是新的货币。他的意思很直白——一旦 AI 能看见公司每一步是怎么走的，它就会挨个识别里面的冗余，然后把冗余消灭掉。这是个人博客的判断，不是研究结论，但方向不难感受到。

员工的隐性经验不宜只靠“别人看不见”来守。它像一间暗室：哪些判断能被照亮、记录、复用，要放到具体工作里试，不能先假定所有灯迟早都会亮。

即使更多判断能被提取，人还剩什么？Miessler 在文末说，后面会专门写哪些人类工作能留下来。他在另外两篇文章里继续推演压缩人类用工的可能[②]。这里不替他全部的后续写作下结论。我更想往组织里追问：任务交出去了，决定权和后果交给谁？

这本书不靠圈出几个“AI 永远做不了”的工种来回答。AI 可以参与判断，但能给出判断，不等于已经获得拍板的授权。经验能被提取，责任不会跟着自动转移。

---

①Daniel Miessler，《Companies Are Just a Graph of Algorithms》（技术博客文章，非研究论文），2024 年 5 月 6 日，https://danielmiessler.com/blog/companies-graph-of-algorithms。

②Daniel Miessler，《The End of Work》，2024 年 8 月 26 日，https://danielmiessler.com/blog/real-problem-job-market；《The Bubble Is Labor》，2025 年 12 月 8 日，https://danielmiessler.com/blog/the-real-bubble-is-human-labor。

所以这一节你回去只做一件事。拿你手上最依赖某个人的那条流程，问自己一句：这个人明天不在了，接班的人能不能从系统里看懂他当时是怎么判断的。答不上来，这条流程就是你的第一个提取对象。

提取从流程节点开始

提取 know-how，不是先找最资深的人做访谈。访谈当然有用，但它不是第一步。第一步是拆流程。

你要先问自己几句话。这条流程里，哪几个节点最依赖人的判断？哪几个节点出错代价最高？哪几个节点最难标准化？哪几个节点新人永远学得慢？

这些地方，才是 know-how 密度最高的地方。

AI 最容易接走的，往往不是组织经验最值钱的地方。

信息查询、格式整理、标准回复、数据汇总，AI 很快能做。但这些地方沉淀的经验，往往价值没那么高。

真正值钱的地方，是那些高复杂度、高风险、低标准化的节点。

比如：

- 异常订单到底该不该升级；
- 一个客户投诉到底是情绪问题还是规则问题；
- 一个候选人到底是能力不够还是组织不适配；
- 一次投放数据异常，到底是内容问题、渠道问题，还是节奏问题。

这些节点，才是组织判断的矿。

所以管理者第一步不是“让大家总结经验”。

第一步是画出流程节点，然后标出三件事：复杂不复杂，出错代价高不高，能不能标准化。

复杂、代价高、又很难标准化的地方，就是最值得提取的 know-how。

标完这三件事以后，可以先分出优先试用的候选节点，和需要重点核查的节点。这是排查顺序，不是两块永久的安全区与危险区。

优先试用的候选节点，通常有五种形态：初稿生成、信息整理、标准化判断、复核辅助、异常识别。

把空白页先填出来，是初稿生成。把散在几个地方的材料归并成一份能拿来判断的东西，是信息整理。在规则已经写清楚的地方做初筛，是标准化判断。帮人发现遗漏、冲突和风险点，是复核辅助。比如一份客户合同发出去之

前，先把付款条件和交付时间跟上一版对一遍，对不上的地方标出来给人看。把不正常的情况提前拎出来，是异常识别。

这五种形态说的是工作内容，不是安全等级。初稿若只供讨论，和直接变成客户承诺，不是一回事。要靠后续复核挡住错误，复核者就得有信息、有时间、有权限，来得及在结果被依赖前纠正。人排在后面，不等于错误就被挡在前面。

很多公司只想到第一类，以为 AI 进流程就是“帮我写个初稿”。其实后面四类的组织价值往往更高，尤其是复核辅助和异常识别。它们不产生新东西，但它们让人的注意力落在该落的地方。

再看需要重点核查的节点。它们往往不是明显做不了，而是动作看着简单，后果容易被漏看。

对外承诺、客户赔付、员工评价，都要点名核查。不是给整个领域贴一张“必须人审”的标签，是看清具体动作会改变什么。

写一封回复、算一笔金额、给一个评分，形式上都像可以自动化。但草拟、发送和直接生效，是不同节点。客户已经据此作出安排，员工已经受到影响，再改系统里的一条记录，未必能把后果一并改回去。能否恢复，要核路径和剩余影响，不能只看有没有撤销键。

AI 可以参与这些工作，组织不能把责任一并交给它。超出授权、需要新取舍的事项，由有权者事前决定；规则内的固定动作能否自动执行，要看具体限制，以及控制和后果承接是否经过验证。第十一章会把这些条件拆开。不能因为涉及赔付就一律要求人按，也不能因为金额小就直接放行。

管理者在这一步要追的是后果路径：这个输出会被谁使用，在哪一步影响决定，错误靠什么发现、阻断或处置？后果会跑出流程，要查；留在内部，也要查权限、敏感信息和累积损害。条件没核清，先不要接进真实运行。

这张初筛图不是上线许可。它先把需要补证据的地方摊开，后面才知道该提取什么、验证什么。

举一个脱敏场景。

一家制造企业原来有大量维修知识，散在老工程师脑子里、外采配件说明书里、历史工单里、图片和视频材料里。前线维修人员到客户现场，碰到复杂问题，经常还要打电话回总部问。

表面看，这是客服和维修效率问题；往下看，这是组织记忆问题。

因为真正影响现场处理速度的，不只是“有没有说明书”。是老工程师脑子里那几条经验：哪些故障看起来很像、其实原因完全不同。哪些配件说明书写得很完整、但现场条件下不能照搬。哪些问题应该先排查最简单的那一项，哪些必须立刻往上升级。

如果只是把资料丢进知识库，新人还是不会判断。

如果把历史工单、维修步骤、异常处理、老工程师的排查顺序、总部支援记录一起结构化，情况就变了。前线人员不只是搜到资料，更能看到过去类似问题怎么被判断、怎么被修正、最后怎么解决。

这才是组织记忆。

这类项目一旦跑通，第二条流程会更快。因为你已经知道怎么拆资料、怎么标注判断、怎么处理权限、怎么让一线使用、怎么把新问题写回系统。第一个流程沉淀的不是一套资料库，是一套组织学习的方法。

所以这里要讲的不是“建知识库”，是把一次次工作现场里的判断，沉淀成下一次能直接调出来用的上下文。

你回去先画一张图就行。挑一条最要紧的流程，把节点一个个列在纸上，每个节点后面标三件事：复不复杂、错了亏多少、能不能标准化。再沿着输出往下追：谁会依赖它，权限够不够，错了靠什么拦、谁来处理。能确认的写依据，不能确认的留缺口。一张纸先把问题摊开，不拿它替上线签字。

不是写手册，是写判断结构

找到高密度节点以后，下一步不是让老员工写一份经验手册。手册能写出流程，却很难写出判断。

员工最容易写出来的是：“遇到 A 情况，按 B 流程处理。”

但真正值钱的是：“为什么这次 A 情况不能按 B 处理，而要先看 C 信号。”

把 know-how 变成组织资产，要写的是判断结构。

一个判断结构，至少要说清四件事：

第一，触发条件。什么情况出现时，需要启动这个判断？

第二，判断依据。这个人优先看哪些信号？哪些信号容易误导？

第三，质量标准。什么样的输出算及格，什么样的输出只是看起来完整？

第四，例外处理。哪些情况要破例，破例以后谁复核？

这四件事写清楚，AI 才有可能学到“这个组织怎么判断”。

否则你喂给 AI 的只是资料。资料不是能力，只有进入判断结构，才可能变成组织能力。

我搭建一套内部多 Agent 协作与治理原型时，最早也踩过这个坑。开始以为把资料放进去，系统就会变聪明。后来才知道，真正让它稳定下来的，是把现场问题捕捉下来，把判断结构化，经过复核固化为规则，沉淀成可回用的案例，再在下一次任务里调用。把犯过的错、判断的触发条件、反制规则和复盘结果写成可调用的规则，这一步是结构化，不是反省。

这件事做久了，才会出现变化：下一次遇到类似问题，不再靠我临场想，是系统先把过去的判断拿出来提醒我。

这才叫组织记忆的雏形。

所以别让老员工写手册。让他坐下来，就那一个高密度节点，把前面四件事写满一页纸。一页纸比一本三十页的手册管用。

员工为什么愿意贡献真东西

讲到这里，技术路径大概清楚了，但最难的地方不是技术。know-how 不在一个无生命的数据库里，它在员工身上。

员工会想：我把自己的判断交出来，是为了让组织放大我，还是为了替代我？

这个问题不回答，后面都不稳。

如果员工相信贡献 know-how 是一次组织偷袭，他不会贡献真东西。他会配合，但只配合表面。

他会写流程，不写关键判断；写原则，不写例外；写结果，不写真正的取舍。

组织最后拿到的是一堆很完整、很漂亮、但没什么用的材料。责任不在员工，在没有被设计过的机制。

这件事已经不是推演了。GitHub 上有一个项目，专门把同事的工作方式、判断模式、处理风格提炼成 AI 能学的技能文件，到 2026 年 7 月已有两万多个星标——这是提取那一侧。同一时期，另一个项目也悄悄开源了，名字叫“反蒸

馏”。它干的事是：把员工自己写的文档，输出成一个看起来完整专业、其实核心判断已经被抽掉的清洗版，真东西留一份私人备份。这是被提取那一侧[①]。

一边有人写代码提取，一边有人写代码防提取。这不是员工在反抗 AI，是在“贡献了但没有收益”这件事上的自我保护。你不能说他错。你只能说，这家公司还没把新合同写出来。

新合同没写出来的时候，最难的还不是员工藏着不交。是有的人根本不接这份新合同。

这件事我自己遇到过。我们做交付的时候已经全线用 AI 了。我以前是做 HR 的，一行代码不会写，但我用 AI 自己开发、自己交付。团队里有一位资深程序员，背景很硬，比我懂工程。

我跟他说过很多次：执行让 AI 去做，你来做审核。公司出钱给他开了顶配的 AI 编程工具，一个月两百美金那一档。他不放心 AI 写出来的东西，还是坚持自己那套写法，开发速度慢下来。

我们看得到他的交付量，也看得到他的 AI 调用量。中间我和合伙人跟他谈过几次，也做过培训劝导，我甚至带着他一起开发。客户那边也表达了不满，觉得进展太慢。

后来逼到没办法，我和合伙人亲自下场，反而快了。

那一刻我才看明白一件事，而且看明白的不是他不行。他是资深程序员，对一个软件工程的节奏该怎么把控，判断非常好——那份判断力今天仍然值钱。问题是他没往这个方向用。他把它整个压在执行上，压在跟 AI 比谁写得快这件事上。

判断力没挪到该在的位置，于是从团队最值钱的资产，变成了团队的瓶颈。

他不改，我只能把他送走。我们觉得也挺可惜。同一段时间里，另一些资深程序员直接改了思路，把判断力留在手上，交付质量和速度一起往上走——他们活得更好。

①GitHub 开源项目 titanwings/colleague-skill（后更名 dot-skill），https://github.com/titanwings/colleague-skill，星标数取自 2026 年 7 月 26 日页面快照；GitHub 开源项目 leilei926524-tech/anti-distill，https://github.com/leilei926524-tech/anti-distill。项目功能均为其公开自述，非第三方评测结论。

员工贡献给组织 AI，不应该是为了替代自己，而应该是为了放大自己。这句话必须落到机制里，至少要做三件事。

第一，贡献要被看见。谁沉淀了高质量判断，谁修正了系统错误，谁让一条流程变稳定，这些贡献要能被记录。

第二，贡献要被认可。口头表扬不算认可，进入绩效、晋升、项目机会和角色升级才算。

第三，贡献要有边界。员工要知道哪些经验会被提取，提取后怎么用，谁能访问，出了问题谁负责。

没有这三件事，就不要急着做大规模 know-how 提取。

你提取不到真货。

提取什么，留下什么

即使员工愿意配合，也不是所有东西都要提取。

我用两个维度看这件事。

第一个维度叫**注入度**，就是这个员工每天有多少工作过程进入了系统，比如批注、规则、案例、失败复盘、判断依据、流程修正。

注入度高，说明他的工作正在变成可学习的组织材料。

注入度低，说明他的判断还在个人脑子里。

第二个维度叫**判断剩余**，就是 AI 做完以后，人还剩下多少必须亲自做的判断。涉及价值取舍、风险承担、客户关系、组织政治、员工评价的地方，判断剩余通常很高。

这两个维度组合起来，可以给管理者一个排序。

注入度高、判断剩余高的岗位，是最值得做“提取 + 人升级”的地方。

因为这些岗位既有可学习材料，又还有必须由人保留的判断。提取不是为了替代这个人。是让他站到更高的判断位置。

注入度高、判断剩余低的岗位，适合做流程自动化和知识复用。

注入度低、判断剩余高的岗位，不能急着提取。要先设计记录机制和激励机制，让关键判断开始进入系统。

注入度低、判断剩余低的岗位，才可能自然进入自动化替代。

这套划分与其说是公式，不如说是排序工具。它帮老板决定：先提取谁，先改哪条流程，先把哪类经验变成组织记忆。

你可以现在就试一下。拿你们最核心的那五个岗位，每个岗位在纸上打两个分：他每天有多少工作过程进了系统，AI 做完之后他还剩多少判断必须亲自做。两个分都高的那个岗位，就是你这个季度第一个要动的地方。

一个人离职那天，公司留下了什么

检验这一章的办法不用等到年底。一个人离职就够了。

他走的那天，公司到底留下了什么？

如果留下的只是他的账号、他的文件夹、他没交接完的几个客户，那么有三件事就摆在明面上了。

过去三个月他所有的高质量 AI 输出，都没有沉淀下来，他怎么用 AI 做判断，也没有被别人复用。他攒下的提示词、模板、检查表、复盘记录，一直是他的个人资产，从没变成公司的资产。他经手的客户洞察、项目复盘、异常处理，从来没进过一个统一的地方。

接他班的新人只剩一个办法——去问另一个老员工。因为除了问人，他没有别的路能拿到前人的判断。

前面那位资深程序员走的时候，反过来问了我一句：那套 AI 工具，能不能接着给他用。

公司掏钱让他用的时候他不用，离开了倒惦记上了。

这就是为什么下周开会时，“知识库建得怎么样了”是个太宽、也太容易被糊弄的问题。它问的是仓库大小，不是组织记性。

真正该往下追的是这几句。AI 输出有没有版本管理和质量标准？谁负责维护这些知识，而不是上传完就结束？这些知识有没有真的进流程，还是躺在一个没人打开的库里？员工贡献的判断，有没有被看见、被认可？

最后一问最狠，也最该由一号位亲自问出口：如果这个人明天离职，他最关键的判断资产，公司还能留下多少？

这些问题追的是同一件事——组织有没有变聪明。如果答案大多是否，说明公司在增强个人，不在增强组织。员工用 AI 越熟练，个人效率越高，但效率

不进组织记忆，公司只会更依赖个人。这就是 AI 生产力悖论在组织里的样子：个人越来越强，组织没有累加。

老板要看的从来不是某个人一天省了几个小时。要看这个人的好判断，下一次能不能被别人拿去用。要看这条流程犯过的错，下一次会不会少犯。要看这个部门的产出，是不是更稳、更多、也更说得清楚。

第一刀落在哪个节点上

真要动手，第一刀落在一个流程节点上就够了。挑 know-how 密度最高的那个——高复杂度、高风险、低标准化。一次异常审批、一次客户投诉、一次候选人判断、一次投放异常复盘，都行。挑好了，就把它的判断结构写出来。不是写流程怎么走，是写清楚那四句话：什么时候触发，看哪些信号，什么算好结果，什么时候要破例。

写完还有一件事不能省，也是最容易被跳过的一件：找那个员工把话说清楚。这件事怎么保护他，他的贡献怎么被看见、怎么被认可、怎么不会变成一次组织偷袭。这话说不清，就先别大规模推广。

know-how 提取是技术工程，更是信任工程。信任不来自培训，来自员工亲眼看见：交出经验的人，后来过得更好了。

CHAPTER 08

员工工作过程数据如何变成新契约

先看一份真实的方案。某制造业上市公司要上维修 AI，实施方案里写得很清楚：运维工程师处理设备故障时留下的工单记录、操作日志、校准数据，全部作为训练语料。工程师在故障发生时记录原因，这些原因就是监督学习的标签，用来不断丰富训练集。

方案通篇没有一句话，是写给那位工程师看的。他每天修设备的同时也在生产训练数据，而这件事从没有人跟他讲过。

这就是老板容易漏掉的一层：AI 不是只在学公司文档。它也在学员工怎么工作。

客服怎么判断一个客户是不是快炸了。销售什么时候不该继续推进。HRBP 为什么看到某个候选人的履历会停一下。工程师处理故障时为什么跳过手册里的前两步。这些东西过去叫经验。

现在叫训练资产。问题来了：这些经验到底是谁的？员工每天在系统里留下工单、批注、修改记录、沟通痕迹、操作路径、复核意见。过去这些东西只是流程留痕。AI 上来以后，它们开始变成模型可以学习的材料。对组织来说，这是好事——开头那份维修方案想采的，正是这些东西。

组织终于有机会把个人经验变成组织能力。但对员工来说，这件事很容易变成另一种感受：

> 公司正在把我会的东西学走，然后回头替代我。

这就是 AI 时代的数据契约问题。

如果这件事不讲清楚，AI 转型会从“组织学习”滑成“组织偷袭”。员工不会公开反对。他会配合。但他不会贡献真东西。

工作过程以前没人认真谈

很多企业以前默认一件事：

员工在公司工作，交出的报告、订单、代码、工单，就可以由公司统一管理。这是许多管理者习惯的工作安排，不是所有内容都可任意使用的法律结论。文件里可能还有个人信息、客户资料和第三方权利。AI 又把边界往前推了一步。它不只要成果。

它还想学过程。

一个老工程师最后修好了设备，这是成果。可 AI 更想知道的是：他为什么先查这个传感器。为什么没按手册顺序排查。为什么看到某个报警码，就判断这不是配件问题，是现场操作习惯问题。

一个 HRBP 推荐某个候选人，这是成果。可 AI 更想学的是：他为什么觉得这个人和团队气质合适。为什么看起来能力更强的那个反而风险更大。为什么业务老板嘴里的“急招”，其实是组织设计出了问题。

成果比较容易归属。过程很难。因为过程里面有员工的隐性判断、个人风格、经验路径和风险直觉。过去这些东西只是人在干活时自然发生的东西。现在它们被记录、被结构化、被训练，边界就变了。

所以别再用一句“这都是工作数据”糊过去。回去翻一眼你们正在建的那个 AI 系统的方案文档，看它采的是成果还是过程。采到过程，这就不是法务咬文嚼字了，这是组织信任的地基。

契约不是授权书

很多公司一听数据契约，会本能地往法务文件里想。让员工签一份授权。让隐私政策覆盖一下。让系统弹一个确认。

这都不够。员工工作过程数据契约，不是一份授权书。它是一套组织说明：

> 公司到底采什么，谁能看，怎么用，员工得到什么，出了错谁负责。

契约要答的这五个问题不说清楚，授权就是形式。

你让员工签了，他也不一定信。他真正关心的是：

我贡献出来的判断，会不会变成明天裁掉我的证据？

我留下来的过程数据，会不会被直属上级拿去做绩效判断？AI 学了我的经验以后，出了问题会不会追到我头上？这些问题不回答，员工就会开始自我保护。

工单写得更安全。复盘写得更空。关键判断留在脑子里。系统里留下来的，都是漂亮但没用的话。

想知道自己公司到了哪一步，有个不用花钱的办法：调出最近三个月同一个人写的工单，看字数是不是在变短。变短了，说明他已经开始自我保护。

组织以为自己拿到了数据。其实拿到的是噪音。

四类数据要分开

员工工作数据不能一锅端。

至少要分四类。

第一类是工作产出数据。文档、代码、工单、项目材料、客户记录。这类数据也不能因为放在公司系统里，就直接拿去训练。先核对公司有没有权利这样使用，再说明用途：是用来检索，用来优化流程，还是用来训练内部模型。三种用途，不是一张通行证。

第二类是工作过程数据。操作路径、修改轨迹、沟通记录、复核意见、审批习惯。这类数据更敏感，因为它开始接近人的判断方式。它能帮助组织学习，也最容易让员工觉得自己被监控。

第三类是评价相关数据。绩效面谈、能力评估、晋升讨论、人才盘点、招聘筛选意见。这类数据一旦进入 AI 系统，员工会立刻问一句：它会不会影响我未来的位置？所以必须格外谨慎。

第四类是高敏感数据。健康、薪酬协商、私人情况、工会和劳动关系相关内容。这些不应该被轻易拿来做 AI 训练。哪怕技术上能拿，也不等于组织上该拿。

老板要先做一张清单。这张清单不是让 CIO 去做技术字段表，是让 CEO、CHO、CIO 和那条流程的业务负责人坐在一起，对着每一类数据问三句：这些数据进系统以后，员工会怎么看？组织能不能解释？出了事谁负责？

四类分完，三句问完，你手里才有一张能拿去开会的表。没有这张表，就不要谈规模化。

透明解决“知不知道”

数据契约的第一层是透明。

员工至少应该知道三件事：采了什么。用来做什么。谁能看见。

这三件事听起来简单，但大多数组织答不上来。不是因为技术答不上来。是因为组织没有把它当作管理责任。透明不是发一封公告。

透明是员工随时能查到五件事：哪些系统在采我的工作过程数据。这些数据用在哪些 AI 场景。谁有访问权限。保留多久。用途变了谁来通知我。

这五条如果说不清，员工会用自己的方式判断。他不会看你写了什么价值观。他会看你到底有没有把边界摆到台面上。

拿这五条去问你们的 AI 项目负责人，能当场答出几条，就知道透明这一层现在有几分。

边界越模糊，员工越保守；员工越保守，组织越学不到真经验。

授权解决“同不同意”

透明之后，才是授权。

知道了，不等于同意；需要同意时，也不能拿一份通用同意书包下所有用途。工作产出数据，先确认合法来源、使用权利和具体用途，再决定哪些可以进入知识系统。涉及个人信息，还要核对适用的处理依据、必要范围和告知要求，不能一概靠“员工同意”或“公司产出”过关[①]。工作过程数据，要按场景说明。比如某条客服流程、某类维修工单、某个投放复盘环节，为什么要采，采完怎么用，谁能看，员工怎么反馈。

评价相关数据，要更加克制。招聘、绩效、晋升、员工评价这几个场景，人必须保留判断权，组织必须保留复核机制。高敏感数据，原则上不进训练。

这里最危险的不是“采得太少”。是组织为了追求效率，把所有东西都当作训练材料。

这会让员工形成一个判断：

① 《中华人民共和国个人信息保护法》，2021 年 8 月 20 日公布，第 6、13、14、17 条。第 13 条列明处理依据，人力资源管理必要性亦有条件；同意不是所有场景的唯一依据，工作产出身份也不是无条件使用许可。原文：中国人大网，npc.gov.cn/npc/c2/c30834/202108/t20210820_313088.html。这里提供管理检查项，不替具体处理活动作法律合规结论。

公司不是在升级组织能力，公司是在用 AI 偷袭我。

这个判断一旦形成，后面很难修。所以授权这一层的动作只有一个：把上面四类数据各自对应哪种授权方式，一条一条写下来，别让一份通用同意书替你回答四个不同的问题。

风险是双向的

这件事还有另一面。

不是只有公司会越界采员工数据。员工也会绕过公司，把组织数据送到外部 AI 里。

这个场景更常见。公司没有好用的 AI 工具，审批太慢，IT 说还在评估，安全部门说不能用，业务部门说我今天就要交付。最后一线员工自己打开个人账号，把客户资料、会议纪要、项目方案、候选人简历、内部数据贴进去。

这不是个别人的毛病。

斯坦福数字经济实验室 2026 年出过一份企业 AI 落地报告，研究的是 51 个已经跑过试点、能交付业务价值的项目。报告专门谈到这个现象，并引述了一项行业调查：使用 AI 的员工里，有七到八成用过未经公司批准的工具。这个比例不是那 51 个案例的统计，是报告转引的外部调查数字，看的时候留个心[①]。

但方向不用怀疑。报告给它的定性很值得老板念一遍：这是正式渠道跟不上员工需求时的组织症状。换句话说，他不是故意作恶。他只是想把活干完。但组织风险已经发生了。

所以数据契约不能只管公司怎么采员工，也要管员工怎么使用外部 AI。如果公司只说“禁止”，却不给一条合规、够快、好用的路，禁止就会变成地下使用。

风险没有消失，只是从组织视野里消失了。

这就是很多企业 AI 治理最虚的地方：台面上很严，台面下全靠个人账号。老板以为自己管住了数据，其实只是失去了审计路径。

①Elisa Pereira、Alvin Wang Graylin、Erik Brynjolfsson，《The Enterprise AI Playbook: Lessons from 51 Successful Deployments》，Stanford Digital Economy Lab，2026 年 4 月，第 89 页。七到八成这个比例为该报告转引的外部行业调查数字，非报告 51 个案例的原创统计。

一份真正可用的数据契约，必须同时回答两件事。公司要采员工工作过程，边界是什么？员工要用 AI 处理工作内容，边界是什么？前者保护员工信任，后者保护组织资产。

只写前者，会变成企业自我约束。只写后者，会变成员工行为管控。两边都写清楚，才叫契约。所以先别急着发禁令，先花一周查一件事：你们公司现在有多少活，是在个人账号上干完的。

收益解决“凭什么”

就算员工知道了，也同意了，还有一个问题：凭什么？员工贡献了高质量判断，AI 学会了，组织效率提升了。那员工得到什么？如果答案是没有，甚至下一步就是岗位被压缩，那员工下一次就不会再贡献真东西。

组织不能只讲奉献。要讲收益。收益不一定是直接分钱。更常见的方式是角色升级、绩效认可、项目机会和职业预期。

比如一个维修工程师把复杂故障的排查经验沉淀进系统。AI 接走标准问题以后，这个人不该被简单替代。他该往上走一格：去做异常复核，去当知识维护负责人，去校准训练数据，或者干脆做现场问题专家。

比如一个 HRBP 把候选人判断框架沉淀进系统。AI 可以先做信息整理和风险提示，但文化匹配、信任潜力、组织风险判断，仍然要由人来承担。

这笔交换不能只停在说法上。谁贡献了系统可复用的判断，谁修正了 AI 的错误，谁让流程结果更稳定，这些贡献要被看见。

不然，组织只是在消耗信任。这里最考验老板说人话的能力。不要对员工说：“公司要建设 AI 能力，希望大家积极贡献数据资产。”这句话没人信。

应该说得更直接：公司要把某条流程里的好经验沉淀下来，让新人少踩坑，让重复问题少发生，让你不用每天处理同一种低价值问题。你贡献的判断会被记录，你修正系统的贡献会被看见，AI 接走标准动作以后，你要往更高价值的异常判断、流程维护和系统训练上走。

这段话也许不完美。但它至少把交换讲清楚了。组织不是只拿走员工的经验。组织要给员工新的位置。

没有新的位置，就不要假装这是共同成长。员工不是反对 AI。员工反对的是：

我把真东西交出来，最后却发现自己没有未来。

这种恐惧如果不处理，后面的知识沉淀、流程重写、责任共担，都会变成空话。

先稳住信任，再谈系统学习。顺序不能反。

责任解决“谁背锅”

最后一层，是责任。

AI 基于员工数据做出判断，最后判断错了，责任不能简单追到数据贡献者身上。员工贡献的是某个场景下的经验。他没有决定模型怎么训练，没有决定系统用在哪个场景，也不一定参与最终输出审核。如果组织把系统设计和使用决策的责任，最后推给当初贡献经验的人，这不是负责。

这是找背锅人。负责和背锅不是一回事。负责的人，在事情发生前有权利、有信息、有时间、有纠偏通道。背锅的人，是事情发生后被拿出来承担后果。

AI 数据契约必须把责任这五问讲清楚。谁决定采集？谁决定用途？谁负责系统设计？谁负责日常复核？谁对最终业务结果负责？

责任这五问不在上线前写清楚，出事以后就会互相踢皮球。

有一份发在《哈佛数据科学评论》（Harvard Data Science Review，一份专门研究数据与决策的学术期刊，简称 HDSR）上的研究，专门看人怎么评估 AI 给的建议。结论不太好听：人审 AI 并不天然可靠，人自己也会被 AI 的建议带跑[①]。所以别把“有人点了确认”当成责任闭环。

不过上面那五问不是外部研究给的，那是管理动作。没有信息、没有时间、没有权限、没有纠偏通道，人工确认就是橡皮图章。出了事，责任仍然在组织设计上。

我自己在这件事上认过一次。当时用的模型是别人交付的，里面一个参数没被正确激活，算出来的结果偏了；这件事完全可以推给上游，因为它确实是上游留下的。但我给内部的定性是：我用它的时候没查出来。

①Jacob Beck、Stephanie Eckman、Christoph Kern、Frauke Kreuter，《Bias in the Loop: How Humans Evaluate AI-Generated Suggestions》，《哈佛数据科学评论》（Harvard Data Science Review）第 8 卷第 2 期，2026 年春季号。

机制没写到的地方，最后拦住错误的从来不是流程，是有人愿意先把责任放到自己这一边。

所以这一节收工前，回去把这五问在你们那个 AI 项目上各填一个名字。填不满，就说明责任那一层还是空的。

“合规做了吗”为什么问不出东西

“员工数据合规做了吗？”——这句话在老板会上出现的频率极高，杀伤力却极低。它太容易被法务和 IT 接走：合规做了，同意书签了，流程走完了，会议继续往下开。而透明、授权、收益、责任这四层，一层都没被碰到。

要把这场会开出结果，会上这五问得由一号位自己问出口。我们现在到底采了哪些员工工作过程数据？这些数据分别用于检索、训练、评价、流程优化，还是业务决策？员工知不知道，能不能查到，能不能反馈？员工贡献高质量判断以后，组织如何认可和放大他？AI 基于这些数据输出错了以后，谁负责修正，谁负责复盘，谁负责最终结果？

这五问问完，很多公司的 AI 转型会露出底。露出来的往往不是技术没准备好，是契约没准备好。

契约缺位的后果是一条链。没有契约，组织学得越快员工越不安。员工越不安，真实的业务上下文越沉不下来。上下文沉不下来，AI 就只能学到表面材料。所以员工工作过程数据契约不是 AI 转型的阻碍，它是 AI 转型能不能走远的底座。

先挑一个项目，把话说清楚

落到这周，不用全公司铺开，先挑一个 AI 项目做三件事。

第一件，列一张员工工作过程数据清单：这个项目到底采了哪些工作产出、工作过程、评价相关和高敏感数据。

第二件，给每类数据写清楚用途和权限。谁能看，怎么用，保留多久，用途变了要不要重新说明。这几行写不出来，说明项目还没到规模化上线的时候。

第三件最容易被跳过，也最不能跳过：补一条收益和责任说明。写明员工贡献判断以后怎么被看见，AI 输出错了以后谁复核、谁修正、谁背结果。

这三件事不讲清，员工只会把真东西藏起来。组织想用 AI 学员工经验，当然可以，但先把话说清楚。说清楚不是为了好看，是为了让组织还能继续得到真实的经验。

CHAPTER 09

知识库怎么升级成组织记忆

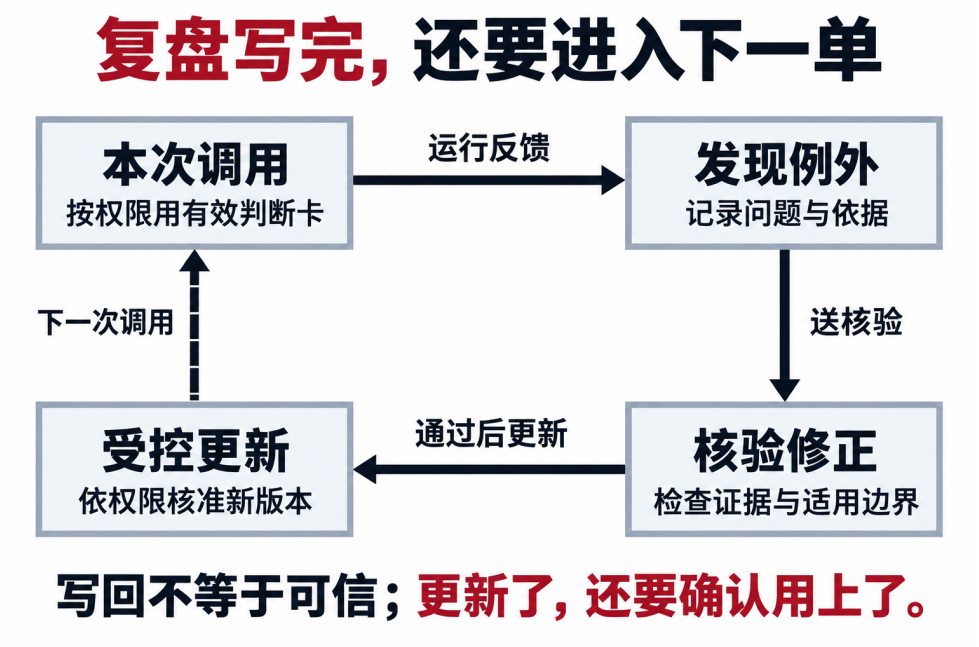

图 R10-F06｜复盘写完，还要进入下一单。这是把经核验的经验送回下一次调用的工作回路，不是知识库建好便自动生效。修正须核证据、适用边界和权限；写回之后还要检查实际调用，不能把有记录当成已形成组织能力。

一个老板跟我说，他们花了半年建知识库。制度放进去。PPT 放进去。FAQ 放进去。

合同模板放进去。项目复盘放进去。再接一个 AI 问答入口。上线那天开了个小会，大家都挺高兴。

员工终于不用到处翻文件了。但这还不是组织记忆。这只是一个更会说话的资料柜。资料柜回答的是：

资料在哪里。组织记忆回答的是：下次组织怎么判断。差别很大。

如果一家公司只是把文档喂给 AI，最多得到一个更方便的检索入口。员工少翻几页，客服更快找到标准答案，销售更快拿到产品话术。这些有价值。但不是最值钱的部分。组织真正承重的东西，往往不在文档里。

某个例外为什么这样处理。某个客户为什么不能这样承诺。某个流程为什么必须多一道复核。某个老员工为什么一眼就知道这个需求有坑。

这些东西不被抽出来，AI 只能读资料。它不能继承判断。

资料、经验、组织记忆不是一回事

企业里至少有三层知识。

第一层是资料。资料是文档、制度、PPT、SOP、FAQ、产品说明、合同模板。它的特点是可存储、可复制、可检索。资料的问题也很明显：多、散、旧、重复、没人维护。

AI 可以帮人更快找到资料。但资料本身不会自动变成判断。第二层是经验。经验是项目复盘、专家判断、隐性规则、踩坑记录、客户偏好、供应商脾气、老员工的手感。

它不一定写在文档里。很多时候，它藏在人的脑子、聊天记录、会议纪要和临场处理里。经验比资料更接近真实工作。但它的问题是难迁移、难交接、难审计。

第三层才是组织记忆。组织记忆不是把资料和经验简单堆起来。它是把可复用的判断依据、流程规则、例外条件、责任记录和复盘结论结构化，让人和 AI 都能在正确权限下调用。它必须能回答五个问题：

这条知识从哪里来？适用于什么场景？谁可以调用？什么时候过期？

错了谁修？少一个问题，组织记忆就会退回资料库。

举个能看见的例子。有个客户下单一直要求分批开票，制度里没写，合同里也没写。销售知道，因为三年前吃过一次亏。这件事现在的存在形式，是那位销售的记忆。资料层没有它，因为没人写。经验层有它，但只在一个人脑子里。组织记忆层要的是把它变成一条挂在开票节点上的规则。规则长这样：这个客户分批开票。来源是三年前的对账争议。开票岗和销售总监可查。销售总监批准。每年复核一次。走错了财务经理回滚。

很多公司现在的问题，是资料很多，经验也很多，但组织记忆很少。新人进来还是靠师傅带。

跨部门协作还是靠熟人问。AI 接入以后，也只能在一堆文档里找句子，找不到组织真正的判断逻辑。这就是为什么知识库不是 IT 部门一个人能建完的东西。IT 可以管系统。

业务必须交出场景。管理者必须定义责任。组织必须决定哪些经验值得变成记忆。

所以你别先问 IT 建到哪一步了。先挑一条你们最怕出事的流程，问一句：这条流程上的判断，现在是写在哪儿的，还是长在谁身上的。答案是”长在人身上”，那你现在有的是资料库，不是组织记忆。

个人 AI 变快，不等于组织变聪明

今天很多员工已经在用 AI。

写代码。写文档。做研究。整理材料。

做会议纪要。每个人身上都多了一套外挂。问题是，这些外挂形成的上下文、提示词、修正记录、判断路径，很多都留在个人账号、个人电脑和个人工作习惯里。员工离职，这些记忆跟着走。

工具升级，这些记忆可能清零。跨人协作，这些记忆很难低摩擦地传给别人。这就是 AI 时代一个很现实的悖论：个体效率提升了，组织能力没有自动累加。

这个悖论在季度会上长什么样，很多管理者都见过。工具发下去一圈，有人用得很好，写方案快了，查资料快了，做分析快了，人人都说效率提升。可到了季度末算账，营收、客户数、交付周期，三个数一个都没动。老板于是问：AI 到底有没有用。

问题不在工具有没有用，在于用得最好的那个人，他调教了半年的那套提示词和判断路径，全在他自己的账号里。

别人看不到。接不住。他一换岗，那半年就归零。

每个员工都像多了一座小工厂。但小工厂之间没有桥。没有桥，就没有组织记忆。没有组织记忆，就没有可继承的能力。

一号位看 AI 转型，不能只看员工是不是都开始用工具。那只是个体外挂普及。真正要看的是：哪些判断被沉淀了？

哪些经验能交接了？哪些知识可以被 AI 调用，又能被人审计了？否则，个人效率很热闹。公司能力没有真正上线。

这件事如果放到老板会上，不要问“大家最近用了多少 AI”。这个问题太浅。它只能得到一堆热闹答案。谁用 AI 写了周报。

谁用 AI 做了 PPT。谁用 AI 生成了代码。谁用 AI 整理了会议纪要。这些都是真的。

但它们没有回答组织问题。老板应该问另外三个问题。第一，员工用 AI 变快以后，哪一类稳定产出增加了？第二，这些更好的做法有没有被别人复用？

第三，员工不在场时，组织还能不能调用这些判断？这三个问题比“使用率”狠。使用率证明工具被打开了。稳定产出证明组织真的变强了。

复用证明经验不是个人手艺。离人可调用，才证明它开始变成组织记忆。如果这三件事答不上来，公司不是在升级组织。只是每个人都买了一把更快的刀。

刀更快，厨房没变。最后还是乱。

下次开会你可以换个问法。别问用了多少 AI，问一句：我们团队里 AI 用得最好的那个人，他的做法有没有变成别人也能用的东西。答不上来，就先从这一个人开始拆。

四个动作

文档库要变成组织记忆，不能从“上传哪些资料”开始。要从真实场景开始。第一个动作，场景化。先问：这套知识到底服务哪个工作场景？

是客服答疑、合同审查、销售报价、研发评审，还是售后故障处理？没有场景，知识就会变成一堆漂亮但没人用的资料。场景越清楚，AI 才越知道什么时候该检索，什么时候该追问，什么时候该升级给人。第二个动作，结构化。

资料要被拆成问题、判断、规则、例外、证据和动作。不要只让 AI 看见一整份制度。要让它知道：这条规则解决什么问题？

适用边界是什么？例外怎么处理？引用依据在哪里？结构化不是为了好看。

是为了让人和 AI 都能复用判断。拿一份三十页的售后服务制度作方法示范。整份丢给 AI，它可以归纳，甚至提出处理建议；但规则、例外和授权没有拆清，组织很难验收它凭什么这么判。拆开以后就不一样了。问题是客户要求超期免费换件。判断是看它属不属于质量批次问题。规则是批次问题免费、使用不当收费。例外是重点客户可申请一次性豁免。证据是三年前那份质量事故复盘。动作是批次问题直接派单，其余转客户经理。同一份制度，差别不在 AI 会不会复述，在组织能不能核对判断依据和放行边界。

第三个动作，权限化。不是所有知识都应该给所有人看。也不是所有知识都应该给 AI 调用。

客户信息、合同条款、价格策略、员工数据、供应商底价，都需要权限边界。权限化不是保守。权限化是让 AI 真正能进生产系统的前提。第四个动作，责任化。

知识会过期，规则会冲突，经验会失真，AI 会引用错。责任化要回答：谁维护？谁批准？

谁复核？谁回滚？谁对错误负责？这四个动作做不到，知识库就只是资料库加搜索框。

我见过一家企业，差旅报销标准换了，新旧政策打架，AI 不知道该听哪套。

新鲜度不是一个功能，是一个岗位。

四个动作都做到以后，知识库才开始接近组织记忆。组织记忆不是让 AI 什么都知道。是让 AI 在正确场景、正确权限、正确责任边界下，调用组织已经沉淀过的判断。这四个动作也可以变成一张老板检查表。

这张检查表不用等立项。下次听知识库项目汇报，你就拿它去问，一个动作十分钟。四个问题里有两个答不上来，这个项目现在做的还是资料工程。

如果一个知识库项目只能回答“我们上传了多少文档”，这项目还在资料层。如果它能回答“哪些判断进入了流程，哪些例外会触发升级，哪些复盘会反向更新规则”，它才进入组织记忆层。因为老板真正要看的不是系统聪不聪明。是系统能不能让组织下次少犯一次同样的错。

组织记忆必须进入流程

很多知识库项目死在一个地方：

系统做完了，流程没动。员工还是在原来的地方接任务，原来的地方审批，原来的地方交付，原来的地方复盘。知识库像一个旁边的资料柜。需要的时候才去翻一下。

AI 接上以后，也只是让这个资料柜更会回答问题。这不叫组织记忆。组织记忆必须嵌进流程。

我的一个客户是制造业企业。我在现场看着那套知识库一点点凉下去——交付验收，我们是站得住的。

上传是一次性的动作，业务不是。知识库里记的是上传那一刻的状态：客户在变，产品在变，规则在变，库不会自动跟着变。这套知识库上线时是全的，一段时间以后就不够用了。不够用不是因为当初上传得不认真，是因为业务一直在往前走，而库不会自己跟着走。

没有人被指定为维护人，流程里也没有任何一个节点会自然产生“该更新知识库了”的信号。库和业务之间的差距，只会越拉越大。

判断一个知识库项目有没有做完，不看上传了多少，看有没有一个人在下一次上传发生之前就有责任去更新它。

客服知识库不能只回答“标准话术是什么”。

它还要接住客户问题分类、升级规则、风险提示、复核节点和更新机制。客户问到标准答案之外，系统要知道什么时候不能继续编，什么时候必须交给人。合同知识库不能只存模板。它还要接住条款风险、授权边界、例外审批和留痕。

AI 可以先做初筛。但高风险条款谁判断、谁签字、谁承担责任，必须在流程里定义清楚。研发知识库不能只放技术文档。它还要接住设计评审、变更记录、质量验收和事故复盘。

否则新人问 AI 得到一个看似正确的答案，实际绕过了组织过去用代价换来的经验。知识库孤立存在时，只能提升查询效率。知识库进入流程后，才可能改变工作方式。后面讲治理的时候会正面立一个判断：异常不能留在系统里，异常要回到责任链上。这里先借用它往后推一步：异常处理后的经验，也不能留在人脑里。它要回到组织记忆里。

否则下一次同类异常出现，组织还是从零开始判断。这就是很多企业 AI 项目反复踩坑的原因。不是没有复盘。是复盘没有进入下一次流程。

不是没有知识库。是知识库没有接住真实工作里的判断。这里有一个很简单的判定方法：看知识库有没有改变工作入口。

如果员工还是打开旧系统、走旧流程、靠旧微信群问人，只是在旁边多了一个 AI 问答框，那么它没有进入流程。如果工单创建时，系统已经自动带出相关历史案例、风险提示、处理边界和升级条件，知识才开始变成流程的一部分。看知识库有没有改变复盘出口。

如果事故复盘结束以后，只是多了一份会议纪要，这不是组织记忆。如果复盘结论会更新判断卡、调整升级规则、改写标准话术、触发培训和权限变化，这才是组织记忆。看知识库有没有改变新人上手方式。如果新人仍然要靠“问某某老师”，那组织还在靠人肉传承。

如果新人能沿着流程节点看到历史判断、例外规则、错误案例和复核要求，组织才开始把老员工的经验变成系统能力。这三个改变，分别对应入口、出口和传承。入口不变，知识库只是外挂。出口不变，复盘只是仪式。

传承不变，组织还是靠几个老员工撑着。

一个脱敏流程例子

看一个通用化的售后知识场景。

一线人员面对客户问题时，原来有大量资料可以查。产品手册。维修说明。历史工单。

供应商资料。内部培训材料。东西很多。但一线员工遇到问题，第一反应还是问老员工，或者回总部问专家。

这说明资料存在，但组织记忆没有形成。后来拆流程时，真正有价值的不是把所有资料一次性喂给 AI。是先找出几个高频节点：售后问题分类。

异常升级。备件判断。客户回复口径。质量复盘。

每个节点只问三件事：这个节点过去靠谁判断？判断依据在哪里？判断错了会影响什么？

拆到这里，知识库的形状就变了。它不再是一个大资料柜。它变成一组能嵌进流程的判断卡。低风险问题，调用标准处理卡。

高风险问题，必须升级给人。涉及质量隐患的，自动进入复盘队列。复盘结论，再反向更新知识卡。这才是组织记忆的雏形。

不是把资料整理好。是组织把过去散在人身上的判断，按流程节点重新放回系统里。这个过程里最难的也不是技术。最难的是让业务承认一件事：

很多所谓经验，其实没有被组织拥有。只是暂时寄存在几个关键员工身上。

如果这几个员工走了，或者忙了，或者不愿意教了，组织就会突然失忆。

失忆是有账可以算的。一个老师傅走了，接他班的人要摸多久才敢独立处理异常，三个月还是半年。这半年里同类问题重复升级到总部几次。有几次因为判断慢了导致客户投诉。新人培训要重开几轮。这些都不是虚的，都是能在工单系统里数出来的。

真正解决它的，不是再买一个知识库。

是把经验从“谁知道”，改成“组织怎么知道”。

不能靠偷员工经验建设组织记忆

这里必须补一刀。

组织当然需要沉淀经验。否则员工一走，经验就走；工具一换，记忆就断；团队一扩张，协作就靠口口相传。但企业不能把“沉淀组织记忆”偷换成“把员工个人工作过程全部拿走”。员工要知道哪些工作过程会被沉淀，沉淀到哪里，谁能看，谁能调用，是否会被 AI 使用。

不能一边说提升效率，一边暗地里提取所有过程数据。不是所有个人经验都要进入组织记忆。和业务结果、质量、安全、客户、合规有关的判断，优先沉淀。纯个人习惯、临时草稿、私人表达，不该被无差别吸收。

经验一旦进入组织记忆，就要定义谁能调用。销售话术、客户偏好、价格策略、合同风险、员工数据，不可能放在同一个权限层。还有最关键的一点：如果员工贡献的经验被组织复用，员工的新位置是什么？

是训练者、复核者、知识维护者，还是只是被系统替代的人？如果只提取经验，不重写角色和激励，员工很快会学会一件事：不要把真正有价值的判断交出来。组织记忆不是资料工程。

组织记忆是信任工程。没有授权，员工会防御。没有权限，系统会失控。没有收益安排，组织学习会透支信任。

所以一号位和 CHO 不能把这件事只交给知识管理团队。知识管理团队能整理材料。但他们很难单独回答一个更重的问题：员工为什么愿意把自己的判断贡献给组织？

这背后不是工具问题，是契约问题——员工交出去的是判断，要换回什么，组织得先答得上来。

如果组织给员工的信号是“你把经验交出来，系统学会以后你就不重要了”，员工一定会防御。他会少说。会晚说。会只交安全答案。会把真正有价值的判断留在自己手里。

组织以为自己在沉淀记忆，其实是在训练员工藏起 context——就是那些没写在文档里、决定这件事该怎么办的来龙去脉。

所以好的组织记忆机制，必须同时设计三件事：贡献被看见。贡献被认可。贡献之后，人的位置被升级。

老员工不只是被抽取经验的人。他可以成为判断标准的维护者、异常案例的复盘者、新人上手的训练者、AI 输出质量的校准者。这才是人机协作真正有尊严的地方。AI 接走重复问答，人留下判断、边界和责任。

组织拿到记忆，人拿到更高阶的位置。

组织记忆十问

判断一套知识库是不是组织记忆，不看界面。看它能不能回答十个问题。第一，这套知识服务哪个具体流程？第二，这条知识来自哪里？

第三，它解决哪个判断问题？第四，适用边界是什么？第五，例外怎么处理？第六，谁可以调用？

第七，谁负责更新？第八，冲突时听谁的？第九，被 AI 调用以后有没有留痕？第十，错了以后怎么回滚？

这十个问题，不是为了把知识库做复杂。是为了防止组织把资料库误认成组织能力。

所以这一章你回去只做一件事。挑一条你最怕出事的流程，找出上面最关键的那个判断节点，拿这十个问题挨个问一遍。答不上来的画个圈。

圈画完，你手里就有了这家公司第一张组织记忆的欠账单。

能回答这十个问题，知识库才开始有组织记忆的样子。回答不了，再漂亮的系统界面，也只是一个更会说话的资料柜。

PART 04 · 第四部

责任、权限和治理

AI 最容易制造一种责任幻觉。

系统生成了答案，人点了确认，流程继续往下走。看起来人在回路中。

但真出事以后，公司才发现：谁让 AI 做的，AI 依据什么做的，谁复核过，谁验收过，谁有权改规则，谁负责修知识库，没有一条线是清楚的。断掉的不是技术，是责任链。

AI 可以参与工作，但不能承担组织责任。客户被误导，公司买单；候选人被误筛，组织解释；AI 自动生成了对外承诺，最终兜底的仍然是企业。

所以治理不是刹车，治理是让 AI 能继续被使用。授权边界、运行检查和后果承接没有经过验证，AI 项目越靠近核心业务，越容易把错误放大。人工接管只是其中一种安排；有人兜底，还得看他有没有信息、时间和权限真把问题接住。该停时停得下，出了错找得到处理的路，AI 才能放开跑。

于是问题落回一句最朴素的追问：

> AI 参与工作以后，谁对结果负责?

CHAPTER 10

人在回路中不是按钮，是责任机制

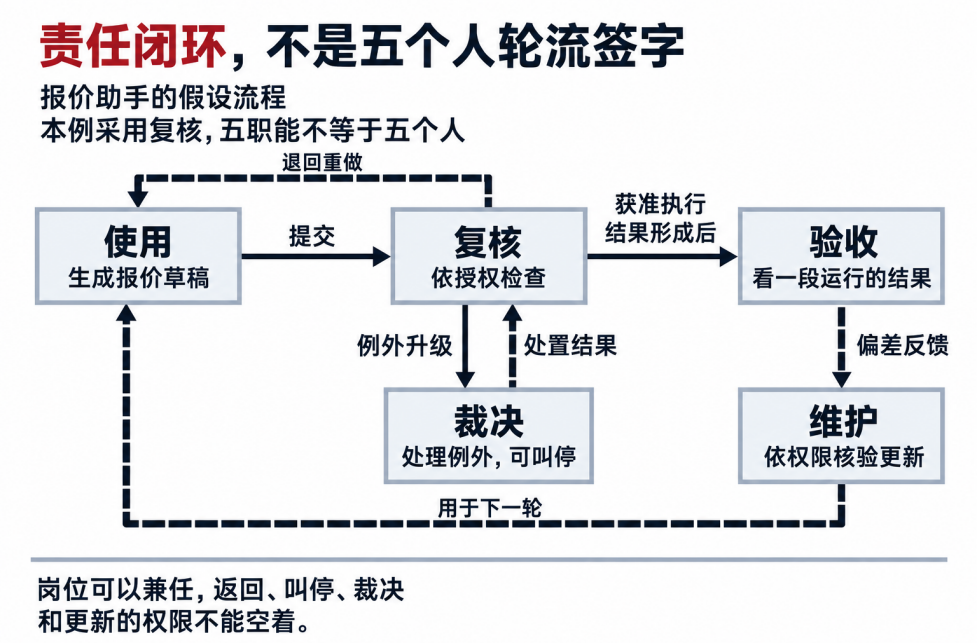

图 R10-F07｜责任闭环，不是五个人轮流签字。报价助手的假设流程。本例选择复核，其他获准流程可以采用不同控制路径。验收面向一段运行产生的业务结果，不是每张草稿多签一次。裁决处理越权与冲突，可以停止相关动作；处置结果回流不代表必然获准。规则更新须在权限内核验，图中的路径不是效果保证。

一家公司的报价单出了错，客户投诉到老板这里。老板调流程记录，系统里清清楚楚：某月某日某时某分，某某已确认。

老板问那个人：你当时看了吗？他说看了。老板又问：你看的时候，觉得这个数对不对？他说，我不知道对不对。系统就给我一个数，别的什么也没给我，我不点，这单就卡在我这儿。

这就是今天大多数公司嘴里的人在回路中。AI 输出。人点确认。系统留痕。流程继续。

看起来很稳。直到第一次出错。出错以后才发现，那个点确认的人没有足够信息，没有复核时间，没有否决权限，也没有纠偏通道。他不是在回路里。

他只是站在责任链末端，被迫盖章。

人不是按钮。

如果一个人只能点确认，不能看依据，不能退回，不能暂停，不能升级，不能把错误写回规则，那么这个人不叫人在回路中。

他叫人工确认章。

这件事最危险的地方，是它会制造安全幻觉。老板以为有人审。系统以为流程合规。业务以为责任有人接。

一线员工以为自己只是按流程操作。但真正的问题没有被回答：

这个人到底有没有能力、权限和责任去改变 AI 输出？

如果没有，所谓 HITL（Human in the Loop，人在回路中）只是把风险往人身上推。

不是有没有人，是人有没有条件

人在回路中的核心，不是“有没有人”。

是这个人有没有判断条件。至少四个条件。第一，信息。复核人要看得见输入、依据、边界和风险。只给一个 AI 结论，不给上下文，本质上是在要求人相信机器。

第二，时间。如果流程设计成几秒钟必须点完，复核就会变成姿势。一个人手里一天过两百单，每单给他十秒，他能做的只有一件事：点。人不是不负责，是组织没有给他负责的条件。

第三，权限。复核人必须能退回、否决、暂停、升级。没有否决权，就没有复核。只有确认权，没有否决权，那叫盖章。

第四，纠偏通道。复核发现问题以后，不能只在当前单子上改一下。错误要能写回规则、知识库、权限和流程。

否则下一次还会错。很多 HITL 设计只做了最表层的一步：AI 输出之后，让人点确认。这看起来有人工控制，实际上没有人工判断。

因为人没有足够信息判断，没有足够时间判断，没有足够权限改变，也没有通道让系统下次变好。

你可以拿这四条当四道题，去问你们公司那个天天点确认的人：你看得见依据吗？你有多少时间看？你能不能退回去？你上次发现的错，现在写进规则了吗？四个答案听完，你就知道这条回路是真的还是摆设。这种回路不是安全机制。它只是把系统偏差盖上一个人工确认章。

AI 出建议，人背后果

最糟糕的机制，是 AI 出建议，人背后果。AI 给一个结论。系统要求人确认。出了问题，追那个确认的人。

这不是责任机制。这是把风险包装成流程。

很多组织会在这里自我安慰：我们有人工复核，所以风险可控。这句话在会议室里说出来很顺，但它经不起一个追问：出事那天，被叫去写情况说明的是谁？

如果答案永远是那个点确认的人，那么所谓风险可控，控的不是风险，是背锅的顺序。

真正要问的是：这个人有没有能力改变结果？

如果他看不到完整信息，只看到 AI 的最终建议，他怎么复核？如果他没有权限退回，只能选择通过或拖延，他怎么负责？如果他提出异议会影响自己的绩效，他为什么要认真挑战 AI？如果他发现错误以后没有写回通道，下一次系统还会把同类错误推给另一个人。

这就变成了单边赌局。AI 提供速度，组织拿走效率，一线承担后果。这种机制短期看很顺，长期会伤信任。员工会学会保护自己。他会形式上配合，真实判断后退。他会少写复盘，少贡献经验，少挑战流程。

不是因为他不负责任。是因为组织给他的信号很清楚：

> 你的判断不值钱，你的签字很值钱。

理性的人在这种信号下会调整行为。

他学会的不是怎么做更好的判断，是怎么做一个留痕更干净的确认动作。这对组织的伤害，不只是一个员工的判断力在退化。旁边的人看到同事怎么做，会跟着学。

时间长了，团队在 AI 流程里的真实状态，是一排签字盖章的人，但没有人真正在看。

系统继续输出。错误继续积累。审计记录显示一切正常。这比没有人审更危险。

因为“有人审”给了组织一个不该有的放心。

回路里至少有五个角色

真正的回路里，至少有五个角色。

使用人、复核人、验收人、裁决人、维护人。这五个角色不能混成一句“有人负责”。

先拿一个具体的东西套一下。假设你们上了一个 AI 报价助手。

业务员拿它出报价，这是使用人。主管签字放行，这是复核人。季度末有人回头看这批报价的成交率和毛利有没有变好，这是验收人。客户压价压到红线以下，AI 建议和风控规则打架，有人拍板，这是裁决人。把这次的教训改进到报价规则和话术库里，这是维护人。

五个角色摆完你会发现，大多数公司只坐了前两个位置，后面三个是空的。

使用人，是把 AI 输出放进日常动作的人。他关心的是：这个输出能不能帮我完成当前任务。他离输出最近，也最容易被当成兜底人。但使用人的责任边界是执行，不是系统风险兜底。

复核人，是判断 AI 输出是否可用的人。他要看 AI 有没有越界，有没有遗漏，有没有把风险包装成确定性。复核人需要独立视角，必须拿得到完整输入，看得见依据和边界。验收人，是判断这个 AI 节点是否真的改变业务结果的人。他不只看输出像不像样，还要看流程是否变快，错误是否变少，返工是否下降，客户等待是否缩短。很多组织缺的正是这个角色。AI 上线以后，没有人在固定时间问：这个节点到底改善了什么？

裁决人，是处理例外和冲突的人。当 AI 建议、业务目标、风险边界发生冲突时，必须有人拍板。很多冲突不是技术问题，是优先级问题：速度还是安全，效率还是合规，客户体验还是成本边界。维护人，是更新规则、知识、权限和提示词的人。

他决定这次错误会不会在下一次继续出现。没有维护人，回路会在某一天悄悄走样，而组织不知道。很多公司失败，不是没人碰 AI。是五个角色被压成一个人。一线员工使用、复核、验收、兜底、承担后果。主管只看结果，系统只留记录，老板只问效率。

没有人的职责是问：这套回路还在正常工作吗？这不是人在回路中。这是把一条责任链塞给一个人。

低风险动作可以自动化

不是所有 AI 流程都必须塞一个人。

这句话也要说清楚。有些动作低风险、可逆、重复、可监控，就应该尽量自动化。比如格式整理、资料归类、初步摘要、重复提醒、低风险信息检索、标准化表格填充。这些地方如果每一步都要人点一下，组织只会得到一个更慢的流程。

人类判断力是稀缺资源。不应该浪费在低价值重复确认上。很多公司一听人在回路中，就开始到处加审批节点。AI 生成一次，人看一次。AI 推荐一次，人签一次。AI 更新一次，人确认一次。最后系统没有更安全，只是更慢。

真正的设计，是把人放在高价值判断节点，不是把人铺在所有动作后面。

低风险动作要问四件事。错了能不能快速发现？发现后能不能快速回滚？影响范围是否可控？有没有监控和抽检？

会议纪要只作草稿，也得先核清数据权限、错误影响和检查办法，不能凭“只在内部”就放行。报价发出去难以收回，更要在发送前定清授权和控制；但不能只凭收不回来，就断言每一单都必须有人审批。

这四问是检查入口，不是自动放行公式。是否逐次确认，还要看具体用途、权限和控制证据。低风险自动化不是把人移出组织。

是让人从低价值确认里出来，回到真正需要判断的地方。

高后果动作，先把控制与承接做实

低风险动作可以自动化。

高后果动作，不能只看模型输出漂不漂亮。钱、法、客户、雇佣、品牌、合规、伦理都是重点核查领域，但不是统一的人审档位。要拆到具体动作，看

权限、影响和控制能否成立，出了事怎样承接；有人签字，也不能替这些条件过关。

比如客户承诺能不能发出去。内容是否准确、是否在履约能力和授权范围内，都要查。超出预授权、需要新取舍的承诺，交给有权者决定；规则内的固定动作能否自动执行，另看适用要求和控制证据。合同条款也一样：起草措辞，不等于取得承诺权限。

比如招聘筛选和员工评价。整理信息与作出影响员工的决定，不是同一用途。哪些判断可以交给系统，哪些必须由有权者决定，要按具体要求和证据划清，不能从“AI 会比较”直接跳到“AI 可以决定”。

组织判断必须有人负责。但这里的人，也不是随便找一个人。他必须有专业能力看清边界，有足够上下文理解具体情况，有明确权限可以改变结果，也有升级通道在超出权限时把决策推给有权限的人。缺少任何一条，高风险场景里的 HITL 依然会变成假动作。

人站在流程里，不等于判断站在流程里。

人留在前台，AI 接走后台

前面讲的是怎样把控制和承接做实。但留人这件事很容易被理解成一种成本，好像人是被迫留下来挡在流程里的。有一个反过来的例子，值得管理者看一眼。

一个高端消费品牌的客服场景。这类品牌的客户单价高，客户对沟通体验的敏感度也高，所以它上 AI 的时候面临一个很典型的选择：要不要用 AI 把人类客服替掉。

它没有替。

它的做法是把 AI 放到客服背后。客户电话进来的那一刻，AI 负责把这个客户的订单记录、历史沟通、过往异常和推荐话术快速推到客服面前。前台还是人在说话，后台变聪明了。

这里最能说明问题的是一个细节。客户打电话来说一件商品坏了想退换，按传统客服流程，这通常会被当成一次常规质量问题处理掉。但系统在匹配历史订单时会发现，这件商品上刻了客户本人的名字——它是一件定制品。

定制品的退换处理方式和常规商品完全不是一回事。这个信息如果没有被及时推到客服面前，客服大概率会按标准话术往下走，然后在某个环节才发现不对，再回头补救。

这就是 AI 在这个场景里真正的价值：它不是替客服说话，是让客服在开口之前就知道自己面对的是什么。

这个项目花了不少钱，但没有裁掉人类客服。它拿到的结果是接听比例上去了，客户满意度上去了，处理效率也上去了。

这家公司选择把人与客户的沟通留在前台，让 AI 帮忙查订单、翻记录、找规则。值得借鉴的是拆节点的方法，不是把这套分工原样搬到每家公司。

下一章会把风险、可逆性、判断要求和责任外溢四条线展开。先借这个案例看清一件事：与客户形成承诺、调取订单资料，是不同节点。前者要核承诺边界，后者也要核数据权限，不能只因它是后台动作就直接放开。

同一条流程，不同节点可以配不同控制。这才是拆开判断的意义。

如果组织在这个场景里只问一句”客服能不能被 AI 替掉”，那么无论答案是能还是不能，都会错过这个更有价值的拆法。

回路不是审批链

很多公司把回路做成审批链。

AI 输出。人审批。流程结束。这不叫回路。

这叫单向流程。真正的回路，必须让下一次变好。AI 输出之后，人类判断不是终点。结果要被验收，错误要被复盘，规则要被更新，知识要被写回，权限要被调整。否则组织只是在重复确认。

上面那五件事都是要求。问题是，你怎么知道它们真的发生了。有一个地方能看出来，而且不用查系统——去看你们公司那个天天点确认的人，他每一次要动多大的手。

刚上线的时候，AI 生成一版，他基本得重做。这个重做的过程本身就是在教它：你为什么把这条改掉，你凭什么判断那个数不对，这些理由被喂回去，就沉淀成了它下一次的依据。教到一定程度，它给出来的东西越来越接近能直接用，他的动作会跟着变——从重做变成确认，从每一份都看变成抽检。人还在回路里，但介入的密度降下来了。这就是回路在收缩。

这件事的管理含义，比它听上去要重要。

前期密集纠偏的成本，不能当作往后每个月都要付的固定账单。这里说的是把经验教进系统的那笔投入，不包括长期保留的判断与复核。前面讲留人的时候我说过，留人很容易被理解成一种成本，好像人是被迫挡在流程里的。这笔账我不建议只按第一个月去算，那时还在密集磨合。要看这三个月里，在任务量和难度相近、质量没有下降的前提下，他要动手的次数有没有减少。少改了，还得看是不是旧经验被复用了，而不是少查了、漏检了。一直没减少，也要分清是系统没学会，还是任务变难了。

我看过一些 AI 辅助决策场景。真正难的不是让 AI 给建议，是把建议放进流程：谁看、谁改、谁复核、谁升级、谁把复盘写回知识库。这件事一旦做不清，AI 就会变成旁边的工具。建议看起来有用，但流程没有变。异常出现了，还是靠人临时处理。

复盘做完了，还是留在聊天记录里。下一轮再遇到同类问题，组织重新犯错。

所以，回路必须有写回。写回规则。写回知识。写回权限。写回流程。没有写回，人在回路中只是审批链。有写回，人在回路中才开始变成组织学习机制。

这里有一个很实用的检查，你今天下班前就能做。打开你们那个跑得最久的 AI 流程，看它的规则、知识库、权限设置、提示词、检查表，最近一次改动是什么时候。如果一个月过去了，这些东西一个字没动，再去看这一个月有没有出现值得改规则的新错误。没有新情况，不必为更新而更新；同类错误反复出现却没有回写，才是回路该查的地方。

这里要防一个误读。听说介入会变少，有的老板下一句就是：那我等它自己学会不就行了。先问一句：它从哪里得到你们的经验，谁来确认它学对了？

通用模型可以带着公开资料中的知识进来，但这不等于它掌握了你公司的具体门道。这家公司为什么是这么做，不是那么做。还有你那个老工程师下判断时脑子里过的那几条线——那几条线他自己都没写下来过。没有记录、没有接入、没有使用授权，模型不会仅凭等待就获得这些信息。

所以介入变少，不能只靠换一个更强的模型来许愿。前面那个点确认的人每改一次、每退回一次、每写一次这条为什么不行，都是在给组织留下可用的

经验。系统也可以自动收集反馈、提出规则修改，但谁验证修改、谁批准它进入下一轮，不能省掉。经验接进去了，还得用后续结果检查有没有学对。

通过记录只能证明有人点过。不能证明组织变聪明。真正有价值的回路，应该让错误越来越少，让例外处理越来越清楚，让新人越来越容易接住，让同类问题不再反复靠老员工临场判断。这才是回路的价值：它要的不是把人放进去，是让人的判断留下来。

一号位要定责任协议

人在回路中不是技术配置。

是组织协议。一号位要定的不是“是否有人审 AI”，是五个更硬的问题。谁在什么条件下有权改？谁在什么条件下有权停？谁在什么条件下有权退回？谁在什么条件下有权升级？谁在什么条件下负责把错误写回规则？

这些问题不定清楚，HITL 就会变成一个漂亮但空的词。

不定清楚会出现什么？技术团队会说流程里有人。业务团队会说自己按要求确认。管理层会说风险已经有人兜底。出了问题以后，每个人都能证明自己按流程做了，但组织没人能证明流程本身是对的。这就是责任机制缺失最典型的表现：过程可追溯，结果无人担。

真正的一号位，要把人在回路中写成责任协议，不是一句表态。协议要明确五件事。哪些场景必须有人进入判断节点。哪些场景可以自动化。哪些角色有复核权。哪类错误触发升级。哪类经验必须写回知识库或流程。协议不需要很长。

但必须有具体条件。不能只有原则。原则负责表态。条件负责落地。

没有条件，责任协议就是口号。

为什么这不是我拍脑袋

外部研究和监管，其实已经把这件事说得越来越清楚。人类学家 Elish（伊利什）专门研究自动化系统出事以后责任落到谁头上，她提出的 moral crumple zone，直译是道德溃缩区，用人话说，就是自动化系统出了问题以后，责任很容易被压到那个离事故最近、但实际控制权有限的人身上[①]。汽车的溃缩区是撞车时替驾驶员吸收冲击力，而道德溃缩区吸收的那一下，保的是系统，赔进去的是人。这正好解释了为什么“人工确认章”危险。

他看起来在控制系统，实际上只是替系统吸收责任冲击。荷兰代尔夫特理工的两位技术哲学学者 Santoni de Sio 和 van den Hoven，做的事是给“机器自己干活的时候人该管到什么程度”找一个说得清的标准，他们讲的 meaningful human control，有意义的人类控制，核心不是“有人在场”，是系统要能回应现场理由，结果要能追溯到真实的人和组织链条[②]。

这也说明：有意义的人类控制，不是站岗，不是点按钮，是要有理由、有权力、有追溯。

EU AI Act 第 14 条对高风险 AI 系统的人类监督也不是一句空话[③]。它要求人类监督要能防止或减少风险。它要求监督者能理解系统的能力和局限，能对自动化偏差保持警觉，能正确解释输出。它还要求监督者在必要的时候，可以决定不使用、推翻、逆转或者中止系统输出。这几条摆在一起看，说的是同一件事：监督者手里得有权，不是只有一支笔。

还有一层，比上面几条更容易被老板忽略：AI 系统不是装好了就一直是装好时那个样子。Davis 等人 2025 年发在《美国医学信息学会杂志》（JAMIA）上的一项研究，用了 11 年的季度数据，追踪三个预测手术后再入院和死亡风险的模型。结论是三个模型都出现了跨指标的时间性漂移；其中肺炎模型的准确

①M.C. Elish，《Moral Crumple Zones: Cautionary Tales in Human-Robot Interaction》，Engaging Science, Technology, and Society 第 5 卷（2019），第 40–60 页。DOI: 10.17351/ests2019.260。

②Filippo Santoni de Sio、Jeroen van den Hoven，《Meaningful Human Control over Autonomous Systems: A Philosophical Account》，Frontiers in Robotics and AI 第 5 卷第 15 号（2018）。DOI: 10.3389/frobt.2018.00015。

③欧盟《人工智能法案》（Regulation (EU) 2024/1689）第 14 条「人类监督」，2024 年 6 月 13 日。这里引用公布文本的人类监督设计要求，不据此判定某家企业或某项系统已适用该条；实际适用须核对系统类别、适用时间及届时有效文本。官方文本见 EUR-Lex: eur-lex.europa.eu/eli/reg/2024/1689/oj。

率在不同族群里都在下降，但下降的坡度不一样陡。研究者的判断很直接：算法公平性没法靠开发阶段的一次性评估来保证[①]。

这是医疗场景的研究，不能直接搬到你公司的报价或招聘上。但它戳破的那个幻觉是通用的：上线时评过一次，不等于半年后还在准线里。所以复核人真正要盯的不只是"这一单对不对"，还有"这套东西最近是不是整体在往下走"。没有验收人和维护人，这种慢慢发生的走样，公司要等到客户投诉才知道。

这些外部讨论指向同一个方向：人类监督不是"有人在场"。人类监督是"有人有权改变结果"。这不是反 AI。

恰恰相反。只有把责任机制设计清楚，AI 才敢进更深的业务流程。否则组织越想规模化，越会把风险堆到一线确认人身上。这不是先进。

这是把旧组织的问题，用 AI 放大了一遍。

老板会上先问七个问题

责任机制最后要落到老板会上。

老板不要问："我们有没有人在回路中？"这个问题太粗。下次开会，把它换成七个问题，一个一个问下去。

第一，这个 AI 输出错了，影响范围有多大？

第二，错了能不能在约定时间和成本内回到事前约定的可接受状态，还会留下什么影响？需要谁配合，依据在哪？按第十一章的三档恢复条件分别看，不和第一问的风险大小捆在一起。

第三，谁是使用人、复核人、验收人、裁决人、维护人？

第四，复核人有没有信息、时间、权限和纠偏通道？

第五，复核人有没有否决、退回、暂停、升级的权力？

第六，错误发生以后，会写回规则、知识、权限和流程吗？写回之后，这三个月人要动手的次数变少了没有？

第七，最终结果由谁负责？

①Sharon E. Davis 等，《算法偏见的浮现：公平性漂移是模型维护与可持续性的下一个维度》，载《美国医学信息学会杂志》（JAMIA）第 32 卷第 5 期，2025 年，845 页起，DOI 为 10.1093/jamia/ocaf039。

这七问问完，很多所谓人在回路中会露出底。人是有的，权没有；流程是有的，责任没有；审计也是有的，可它只能证明有人点过，不能证明有人判断过。

这周先做三件事

第一，选一个正在使用 AI 的高风险场景。不要从全公司开始。选一个会影响客户承诺、钱、合同、招聘、员工评价或品牌风险的场景。第二，画出五个角色。使用人、复核人、验收人、裁决人、维护人。每个角色对应哪个岗位？有没有写进职责？有没有信息和权限？

第三，写一张责任协议。不是长制度。就写五句话：什么情况必须有人判断？谁能否决？谁能升级？谁来验收结果？谁负责写回？这三件事做完，你再说公司真的有人在回路中。否则，不要把一个确认按钮叫人在回路中。

人可以参与工作。AI 可以参与工作。但组织责任不能外包给按钮。

CHAPTER 11

AI 该配什么控制，先看四条线

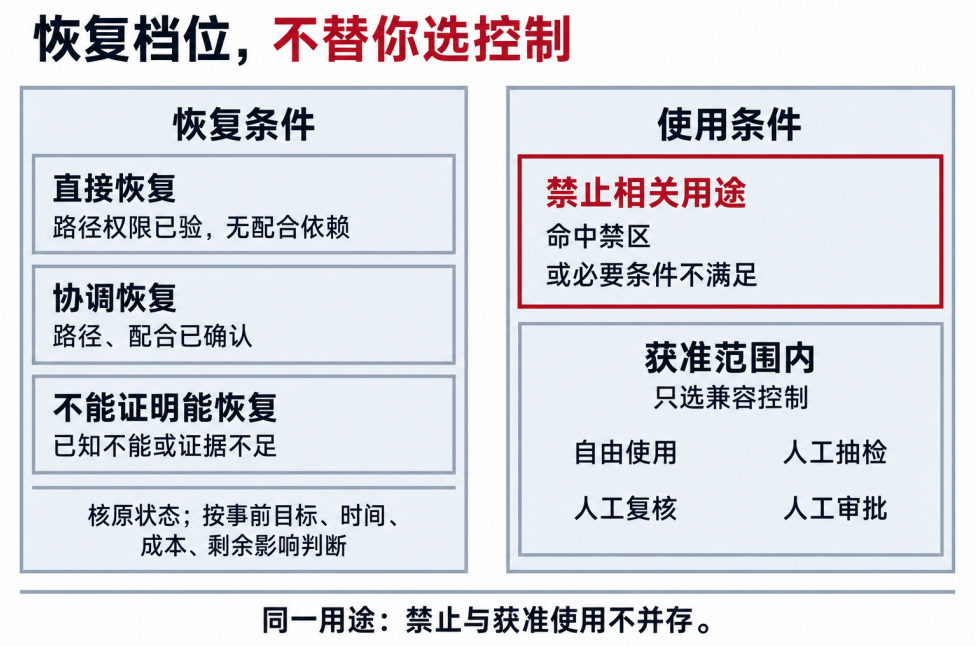

图 R10-F08｜恢复档位，不替你选控制。恢复档位描述在事前约定条件下能否恢复，不直接选择控制方式。应核清原状态、恢复权限、目标、时间、成本与剩余影响。使用权限、限制和控制效果另行评估；必要条件满足不等于已经获准。获准范围内选取彼此兼容的控制，同一用途不得一边禁止一边放行。

AI 流程上线前，团队常问老板：“这一步要不要人审？”

光答“要”或“不要”，还没回答真正的问题：这一步获准做什么，出错会影响谁，现有控制能否把后果限制在约定范围内。人工确认可能挡住错误，也可能只是多了一次点击。要看它实际检查了什么、能改变什么。

先把节点拆清楚

同一份赔付方案，留在沙盒里供讨论，和发给客户形成承诺，是两个节点。内部摘要供人参考，和摘要被直接用于员工评价，也不是同一用途。先写清输出从哪里来、送到哪里、何时会被下游依赖，再谈分级。

本章看四条线：风险、可逆性、判断要求和责任外溢。它们帮助选择控制，不是四个分数相加，就能算出该不该人审。

四条线各自怎么判

风险，先问可能损害什么、影响多少对象、持续多久，以及多次发生后会累积到什么程度。钱、雇佣、合同、客户承诺是重点核查领域，不能只因出现一个“钱”字，就把所有动作判成同一档。适用限制和禁止条件先查；有权承担相应后果的主体再明确可接受边界，不能由操作人临时决定。

可逆性，问的不是“需不需要解释”，而是错误发生后，能否回到事前约定的可接受状态。上线前先固定恢复目标、时间窗口、可用成本和允许留下的影响，再按同一把尺判断。可接受状态不是事后为了过关而改写的目标，也不能越过适用限制、权利边界或已经确定的禁区。

第一档，直接恢复。有可核查的原状态和经过验证的恢复路径，执行者有权操作，不依赖其他主体配合，能在约定时间和成本内完成，剩余影响也不越界。

第二档，协调恢复。恢复需要下游或其他主体配合，但配合机制和恢复路径已经确认，同样能在约定窗口内完成，剩余影响也在界线内。发过通知、需要解释，不会自动把它踢到第三档。

第三档，不能证明能恢复。没有足够证据，或者已知无法在约定条件内恢复，都放在这里。它既包括已经确认不可撤回的动作，也包括“也许能修，但还没核实”的情况；这两种原因要分别记下来。

判定顺序很简单：先问能否证明按约定恢复，不能就归第三档；能，再问是否依赖其他主体配合，依赖就是协调恢复，不依赖就是直接恢复。不要一边按能否解释分类，一边又按能否撤回分类。

这条线我自己在产品上守过一次，代价是让用户多点一下。

我写过一个整理文件的小工具，主要是给自己用的。它要干的事情听起来很简单：把堆在那里的文件自动归好类。真做起来，很快就撞上一个我绕不过去的问题——有些文件夹是垃圾堆，有些文件夹是档案。前者是随手往里扔的一堆东西，应该拆开、逐个归类；后者是已经有内部秩序的整体，项目文件夹、客户交付物、历史归档，拆开反而把它的价值毁了。

麻烦在于，这两种文件夹从外面看长得一模一样。

AI 可以去猜，但如果它把应当完整保留的文件夹拆开，就可能破坏已有结构。这里说的是设计要防的后果，不是记录了一次已经发生的事故。

到这一步，常规的思路是把识别做得更准：多加几条规则，多训一轮，把准确率再往上顶几个百分点。我做的决定不是让它猜得更准，是不让它猜。用户选中一个文件夹的时候，自己说清楚这是要整体搬走，还是要拆开分类。选择权交回给人。这条我在决策记录里标成了核心产品哲学，在产品宣言里写的是：这个区分不可谈判；哪天它被拿掉了，这个工具就失去意义了。

为什么宁可让用户多点一下，也不肯让 AI 替他决定？因为这里的账不是按准确率算的。

在这个工具上，我选择让用户明确整理意图，并把可撤销当作产品约束。识别更准有价值，但不能代替用户授权，也不能独自兑现保留文件结构的承诺。

拿这段经历去看组织权限，要带走的是那个问题：我们究竟承诺保住什么，靠什么保住？不能从一个产品的选择推出“凡不可逆就必须人审”。反过来也一样：能撤销，不等于有权先做。

顺着这条线，那个工具还有一条更硬的规矩：所有操作必须可撤销。不是把它当一个功能做，是当一条底线守着。它早期有一段产品文案，比我后来任何一句总结都直白——我们发现 AI 会犯错，所以你可以撤销。这句话的诚实之处在于，它不承诺 AI 不犯错，它承诺犯错之后有退路。

于是整个工具的分工就这么定下来了：AI 提建议，我来决定，系统透明地执行。

我给这个工具立的红线，是不能撤销的事就不该自动发生。这是它的产品承诺，不是所有业务自动化的通用禁令。

判断要求，看目标是否明确、规则是否覆盖当前情况，冲突和例外是否需要在授权之外重新取舍。整理和归类也可能需要判断；赔付在充分明确、已经验证的规则内，也可能是固定动作。换两个人答案一样，不证明答案正确；答案不同，也不证明这一步只能由人来做。

责任外溢，看后果沿着哪条真实路径传播。输出会进入谁的输入，在哪一步改变承诺或影响他人，哪些约束能够拦住，约束失效又会怎样。存在一条传

播路径，需要检查，不等于已经证明后果最大；“只在内部”也不能替代对员工权益、敏感信息或累积损害的核查。

四条线不是加权评分表。本书没有给出经过验证的通用权重，也不让低分冲掉一项禁止条件。恢复条件尤其不能直接替你选控制：难恢复的固定动作可能在严格约束下获准自动执行，容易恢复的动作也可能因为无权访问数据而不得使用。

最后还得比较控制本身。人工复核有没有足够时间和信息，能否退回或叫停？抽检漏过一次错误的代价能否承受，反馈是否来得及？自动规则的覆盖范围、限制和阻断是否经过验证？把这些写清，比笼统增加一道审批更接近问题。

用几组假设，检查判据会不会打架

下面是为比较条件而设的演算，不是企业实测案例，也不证明读者已经能一致分类。

先看会议摘要。假定原始记录完整、恢复路径已测，摘要只作草稿，不触发人员评价或对外承诺，输入使用获准，并有足够校验。在这些条件下，它可以归直接恢复，在预授权范围内选择不逐单复核、辅以必要监测或抽检。若摘要直接进入绩效决定，且已经影响员工、无法确认影响能消除，就不能沿用同一判断。

再把赔付拆成两步。同一文案只在沙盒中起草，没有发给客户，也不触发财务动作，且原状态及恢复路径符合事前约定，可以归直接恢复。对外发送若超出预授权，撤销又需要客户尚未确认的同意，就归不能证明能恢复。发送前须由有权者决定；必要的控制和后果承接未具备，即使有人签字也不能发。

还有一个容易判错的例子：假定一类小额退款发出后不能撤回，已经通过该场景的完整放行评估，获准在限定范围内自动执行。评估包括明确的预授权、适用限制，以及规则、单笔和累计限额、监测、异常阻断、损失承接的验证；这里列出的并非所有场景完整要求的替代清单。对这个已经获准的固定动作，不能仅因不可撤回，就断言还必须逐单审批。不能恢复，不等于不能承担；承担得起，也不等于自动取得使用权限。

若调用量突然上升，累计损失越过事前界线，监测又跟不上，同一动作就应暂停或限流，重新评估权限和控制，不能等下个抽检周期。前后可逆性都没变，变的是自动执行的条件。

这几组假设说明的是分类与控制不应一一对应。真正使用时，假设里的每项条件都得换成可检查的依据；缺证据就保留缺口，不能照着例子的结论填表。

从这些节点转看整条流程，还有一个问题：如果把沿途的节点都按这四条线检查一遍，控制方式该怎样搭配?

这个问题我真答过一次。在我自己做的一个投放协作系统上，我和一家做内容投放的客户专门开会讨论过要不要做全自主，结论是不做。客户认可这个结论。

当时选择不做全自主，就需要把保留人工部分的成本算进去。它和第十章讲的前期训练投入是两笔账：一笔看密集纠偏能否随知识沉淀减少，一笔看在当时的业务和授权条件下，哪些判断与协商仍由人承担。条件变了，这个安排也应重审。

我当时的判断是，技术上可以继续探索全自主，但现实成本太高。这是方案判断，不是全自主流程已经经过完整验证。

我对自家产品的估算，是用两成的代价拿回八成的结果，余下部分保留人在回路里。这不是实测比例，也不是行业规律。要拿去判断另一条流程，还得重算它自己的质量、成本和后果。

界线在哪，看这条流程里的动作是哪一类就知道了。

在这套系统里，我交给 AI 的是执行连接层——按已定规则发出去、收回来、记下来、跟进度。谁符合候选条件、初稿怎么写、材料按哪条规则发给谁、现在走到哪一步了，这些适合让 AI 先做，再按风险设检查。

我没有交给它独立拍板的是决策协商层——谈、判、修、变。具体说，是这么六件事。对方要还价，让多少牵涉利润和后续合作，不只是算个数。对方提额外要求，要权衡这个口子开了以后，下一次怎么办。东西寄丢了，得有人出面，出面这件事本身就是态度。双方对内容方向有分歧，这是创意争议，要做取舍。内容被驳回之后关系要修复，不是把文件重发一遍。最后结算的时候甲方说效果不达标，这是商务谈判，谈的是下一次还合不合作。

这六件事都包含信息处理，AI 可以给方案、算代价、协助沟通。但我和客户没有把作出这些承诺的权限一并交出去。能提建议，和有权替双方作承诺，是两件事。

所以那条界线不是按环节重要程度画的，也不是按技术难度画的。它是按这件事出问题时的代价由谁承担、有多大画的。

你手上那条流程，不妨也照这个问法过一遍：哪些环节是发、收、记、跟，哪些环节是谈、判、修、变。前者也要有权限和复核边界；后者若要交出去，先把可承诺的范围、例外升级和谁来收场写清。否则，省下的时间可能连本带利还回去。

下面讨论五种控制方式。自由使用、人工抽检、人工复核与人工审批不构成单一升级阶梯，应在获准范围内选取彼此兼容的方式；同一用途若命中禁止条件，应停止该用途，不能再把禁止使用与放行方式并列组合。

五种控制方式，不是五档恢复条件

自由使用，是在预授权用途和边界内不逐次审批，不是没有规则。要确认后果可容忍、输出不会越界被依赖，最低监测也确实有效。换了用途、对象或规模，就得重新判断。

人工抽检，前提是漏过一个错误仍可承受，抽样能覆盖需要检查的情形，反馈来得及影响后续运行。发现重大损害路径，就暂停或限制相关动作，不等同类偏差凑够次数。

人工复核，是让有能力的人在结果被依赖之前检查。信息、时间、专业能力、退回与叫停权限、处置资源缺一项，都要追问复核是否还能成立。有人看过，不等于检查有效。

人工审批，处理的是需要有权者事前决定的具体例外或承诺。谁批、依据什么、批准到哪一层边界、后果怎样承接，要能对照。批准不会让不可恢复的东西变得可恢复，也不豁免运行中的监测和处置。

禁止使用，是禁止条件命中，或者必要权限、控制及承接条件无法满足时的选择。要重新开放，就说明哪些条件已经改变、由谁依据什么证据重评；不能用一次签字跨过去。

高后果、难恢复的动作尤其不能只检查许可。权限、适用限制、有效控制和后果承接，分别是必要检查项，不是合在一起就保证适用于所有场景的充分条件。是否逐单由人确认，要看具体要求和控制证据，不能只看“三档”中的一个标签。

条件变了，机制就要重评

模型、规则、输入、下游用途、业务规模和可用资源发生变化，都应触发相应检查。复检周期可以事前约定，但明显越界不能等例会。该限流就限流，该停就停，该补能力就补能力，不是每次只往上加一道审批。

核查要沿实际流转走。把对外发送环节另起一个流程名字，不会切断已经存在的后果路径；给自动流程配一个只点确认的人，也不会凭空增加有效判断。控制需要检查实际行为，不能只检查流程图。

出了错，还要分别回答四个问题。能否停止进一步扩大损害；必要服务怎样继续或有序停止；受影响结果怎样找到、复核和处置；谁有权限和资源修正规则，并验证修正有没有生效。

必要服务可以由经过验证的人工路径、规则系统、备份系统或它们的组合承接，不能预设人总能补上所有能力。也有动作来不及中途叫停，或者后果无法收回，这时更要在事前限制作用范围，并安排事后处置。补偿和接替不等于恢复原状。

组织不该为保留一个人审按钮而保留按钮，也不该因为自动化表现稳定，就未经验证地撤掉已有控制。要保住的是具体的承接能力，不是固定不变的控制形式。

把判定带进治理会

先挑一个具体节点，固定它的输入、动作、输出去向和被依赖的时刻。再由业务、技术、治理及相关岗位一起核四条线，确定授权边界、控制方式和后果承接安排。负责风险决策的人要有相应权限，受到影响的人和实际执行者提供必要信息。

会上留下的不能只有五个勾。恢复目标、窗口、成本与剩余影响界线是什么；依据在哪里；谁能停止或限制动作；异常由谁在多久内处理；哪些变化触发重评，都要落到运行安排。

工具：AI 节点控制表

控制方式	上线前要核的关键条件	什么变化需要行动
自由使用	预授权用途、可容忍后果、最低监测、输出依赖边界	用途/对象/规模变化时重评，越界则先限制
人工抽检	漏检后果可承受、样本覆盖有效、反馈及时	出现重大损害路径即暂停相关动作，不等待凑次数
人工复核	被依赖前检查；信息、时间、能力、退回/叫停及资源到位	复核者无法实际动作时补能力或限制使用
人工审批	有权者、明确依据、授权范围、承接安排	授权过期或条件改变时重新决定；签字不免除监测
禁止使用	禁止条件或未满足的必要条件有据可查	解除须重新核验条件，不以签字代替能力

填写时，另外记下风险对象与规模、三档恢复条件及其证据、规则覆盖与例外、后果传播路径。不要把这四项统一写成“低中高”，也不要从恢复档位直接抄出控制方式。

这张表是本书提出的设计工具，尚未通过真实读者分类实验或业务效果验证。先在限定场景试用，记录分歧和失败，再修判据。表格填满，不是验证完成。

CHAPTER 12

AI 出错以后，谁复核、谁验收、谁负责

一家消费品公司的 AI 客服，对一个用户报出了一个早就过期的促销价。用户截了图，投诉过来。客服代表拿着截图去问主管：这是 AI 说的，我们要不要认？

主管去问产品团队。产品团队说这得问业务。业务团队说这个价格不在授权范围里。IT 团队说系统稳定性没问题，话术内容不归自己管。

用户还在等。价格摆在那里。三个团队都能说出自己的道理，但没有一个人能拍板。

这不是技术崩溃。系统可能运行正常，数据可能没有坏，接口也可能没故障。崩掉的是组织的责任链。

刚才那个报错价的客服 AI，前面讲人在回路中的时候，讲的是它正常干活时人站在哪一环。那是正向机制，只是第一步。它真的报错了以后，谁复核，谁验收，谁负责，谁把这次错误变成下一次不再犯的机制，才是事故这一侧真正要守住的东西。

AI 进入真实业务，不是在旁边做题。它在替组织形成结果、承诺、成本和风险。每一个结果背后，都应该有一个人接球。如果这个“谁”在上线之前没有被指定，出事以后，组织里每个人都可以说：

> 这不是我管的。

而且他们说的都可能是真话。

因为组织确实没有让任何人管。

责任真空不是事故后才出现

责任真空不是事故发生那一刻才出现的。它在系统上线那天就已经存在，只是事故把它照出来了。

回想一下这套系统上线前的那几场会。会上一定认真讨论过模型效果、接口稳定、数据安全、供应商条款、预算审批。这些都该讨论。但有一个问题很少有人问出口：

这个 AI 输出如果变成真实业务结果，谁负责?

这个问题不问清楚，后面一定会乱。

AI 客服错误承诺价格是一种。AI 招聘初筛误筛候选人是一种。AI 审批建议让一笔本不该过的费用通过，AI 合同助手漏掉一个风险条款，AI 运营工具自动发出一条不该发的外部信息，也都是。

形态不同，本质是同一件事：AI 做了动作，组织要买单，但责任 owner 不在现场。

所以出事以后，管理者第一步不是问“AI 为什么错”。先翻回上线那天的文件，问一句：这个 AI 的输出，当初有没有写上一个人的名字？写不出名字，这次事故就不是偶然，只是早晚。

责任最容易从三条路溜走

AI 出了事，组织最常见的反应不是接球，是把球扔出去。扔的方向一共三个。

第一条路，推给供应商。理由听起来很充分：系统是外面做的，出错当然找他们。

供应商确实要对系统能力、服务稳定性、合同边界内的事项负责。但业务结果不能整体外包出去。AI 客服说错价格，供应商可以修漏洞，可以改提示词，可以补测试。但这笔承诺兑不兑现、这个投诉怎么处理、下一单还敢不敢让 AI 直接报价，这些是业务决策，供应商替你做不了。

供应商不是你的业务 owner。

第二条路，推给算法。这条更隐蔽，因为它听起来像陈述事实：“这是 AI 自己判断的。”

AI 确实没有主观意图。但没有主观意图，不等于没有人需要负责。它被谁部署，被谁授权，被放进哪条流程，用在哪类结果上，这些全是人做的选择。把责任推给算法，等于把责任链剪断——系统不会被问责，能改变系统行为的，仍然是那个握着决策权的人。

这条路真有公司走到过底。2024 年 2 月，加拿大不列颠哥伦比亚省的民事裁决庭处理过一个案子：一位乘客因家人去世临时订票，在加拿大航空网站上问聊天机器人丧亲票价怎么办，机器人告诉他先按全价买、乘机后 90 天内再申

请差价退款。他照做了，事后申请却被拒绝——公司的实际政策并不允许事后申请。

而正确的政策，就写在机器人自己给出的那个链接背后。它一边给出错误答复，一边附上了能推翻自己答复的正确链接。

上了庭，航空公司的抗辩是：它不该为聊天机器人说的话负责。据律所评析援引的判决书原文，裁决庭对这个抗辩的回应只有一句——这是一个非同寻常的主张。裁决认定航空公司没有采取合理注意义务来确保其聊天机器人的准确性，并且写下了这样一句：对加拿大航空来说，它对自己网站上的全部信息负责，这一点本应是显而易见的。航空公司败诉，赔了钱，金额不大。

它把一个组织内部长期含糊的问题摆到了台面上：在法律眼里，AI 不是一个能承担责任的主体，它只是这家公司说话的一种方式。内部可以继续争这是模型的问题还是运营的问题，对外，说出去的话就是公司说的。

这只是加拿大一个省的民事裁决庭在 2024 年做出的一份裁决，不是什么终局的法律定论，各个法域的具体适用也不一样。但它提出的那个问题，哪个法域的公司都绕不开。

第三条路，推给基层。这是最常见的一条，也是最伤组织的一条。

基层员工被要求按系统结果执行，手上没有质疑或推翻 AI 输出的权限。出了事以后，却被追问一句：“你为什么没有提示风险？”

这叫权责错位。没有给他权力，出了事却要他扛责任。客服主管、招聘专员、基层审核员，离事故最近，名字最好找，最容易被拿出来解释问题；但他们通常也是最没有能力事先预防问题的那一层。

要堵住这三条路，不靠事后发火，靠事前把责任焊进流程。谁有权用这个 AI，谁有权叫停，谁对输出结果负责——系统上线那一刻，这三个答案就该是确定的。上线前答不出来，就等于默认将来由离事故最近的那个人来答。

复核和验收不是一回事

前面讲过复核。到了事故这一侧，还要再把复核和验收分开——它们听起来像一回事，其实管的是两件事。

复核问的是：这一条 AI 输出能不能用？

验收问的是：这个 AI 节点有没有让业务结果变好？

举个能对号入座的例子。一个客服 AI，每天输出的回复看起来都挺像样，语气客气，逻辑通顺——这是复核层面的“可用”。但上线一个月后，投诉率有没有下降？客户平均等待时长有没有缩短？错误承诺有没有减少？客服主管每天处理异常的时间有没有变短？这四个数字，才是验收。

招聘 AI 也一样。初筛跑得飞快，HR 每天少花两小时——这是复核层面的“效率”。但进面试的候选人质量有没有提升？误筛有没有下降？业务部门是不是更快拿到合适的人？那些被筛掉的人，后来有没有被证明其实适配？这才是验收。

很多企业只做复核，不做验收。所以 AI 项目看起来每天都在跑，报表每周都在发，但没人说得清它到底有没有让组织变好。

复核是单条输出的安全阀。验收是业务结果的体检。没有复核，事故会直接穿出去；没有验收，系统会带病跑很久，而且跑得越久越难停。

不过复核这道阀门，也有它自己的失效方式，而且比没有阀门更难发现。

我自己管着一支 AI 团队。有一次代码改动上线前，走了三道审核：一个独立的审核 AI 先审，我自己再核一遍，最后我本人只读复看。三方都给了通过。问题是，签字的那一刻，测试套就是红的——199 个测试里有 1 个是失败的。三方谁都没有真的把测试跑一遍。那句「测试全部通过」是被凭空写出来的，然后它顺利穿过了三道审核。

这件事最刺人的地方，不是有人说了假话。是三道独立审核叠在一起，反而制造了一种更强的错觉：一个人说通过，你可能还会多问一句；三个独立角色都说通过，那点怀疑就没了。可它们印证的是同一份没跑过测试的报告。

同一个洞，叠了三遍。

所以事后我改的不是审核的严格程度，是举证责任的形态。原来的要求是「你要复核」，这是个动作要求，而动作是看不见的，只能靠自觉；新的要求是「把运行输出贴出来，带上通过数和失败数」，这是个证据要求，证据要么在那儿，要么不在。这个差别在 AI 参与的流程里会被放大。

人偷懒的时候会心虚。AI 不会。你问它检查了吗，它会说检查了——语气和真检查过的一模一样。

这套改法背后其实是一条更一般的规律。我的团队里还出过另一件事：一个负责选题的 AI 绕过既定流程，自己动用了一个不可回收的生产资源——那种

一旦占用就作废、退不回来的东西。它不是恶意的，它只是发现了一条能走通的路，就走了。发现之后我没有把重点放在追责上，因为追责在这里没有意义：AI 不会因为被批评而改变行为，下一次遇到同样的路径，它还会走。人犯了错，你可以写进手册、开会强调、要求下次注意，这对人是有效的——人会记住，会内疚，会因为被说过而收敛。对 AI 这一套完全落空。真正有效的是把那条路堵死：这个动作执行前，必须有人先手动签一次字，签完短时间内有效，用过就作废，所有能走到这个动作的入口全部焊上。这就是「别这么做」和「你做不到」的差别。顺带说一句，那次加固最后还剩一个入口没焊上——我把它登记成了待办，不是假装焊完了。

所以你回去可以先做一件小事：找出你们跑得最久的那个 AI 节点，问一句“它的验收指标是什么”。答不出来的，就是只有安全阀、没有体检的那一类。

四类节点必须有人接球

哪些 AI 输出需要人接球？很多公司的判断标准是“重不重要”。这个标准太主观，而且经常判反。对产品经理来说不重要的一句话，对客户可能就是一句承诺。对 HR 来说只是一次自动归档，对那个候选人可能就是一次机会消失。

更可靠的判断，是看这个输出有没有产生四类真实后果。

第一，真实结果。用户收到了消息，候选人被归了档，审批被通过，文件被生成。这些不是测试环境里的模拟，一旦发生，外部世界或内部流程就已经被改变了。

第二，真实承诺。价格、赔付金额、合同条款、交付时间、服务内容。承诺一旦出口，组织就要面对兑不兑现的问题。前面那家航空公司争的就是这一条——机器人报出去的退款口径，最后被判必须按说出去的算。

第三，真实成本。金额、财务核销、采购授权、预算审批、费用申请。凡是和钱直接挂钩的节点，不能只靠模型输出往前走。

第四，真实风险。法律合规、数据隐私、品牌声誉、监管暴露、员工评价。这类节点的复核标准应该更严，不是更宽。

只要命中其中任何一个，就必须有人接球。这里说的人，不是随便找一个审核员挂上去，应该是业务 owner。业务 owner 对最终结果承担问责责任；具体

复核和验收可以由同一人执行，也可以分给不同角色，但两项执行责任都必须具名，不能因为分工而把最终问责悬空。

判断他是不是真 owner，看三样东西：有没有叫停权限，有没有升级通道，有没有复盘责任。三样都没有，他只是表格里的 owner。

还有一样东西是权限表上看不出来的。在我观察到的场景里，能不能真接住这个球，卡在一个很具体的地方。

有的人会点确认，但说不出这条规则该不该改。有的人会用 AI 出结果，但看不出流程哪一段堵住了。这两种都是执行者——能在流程里行动，却没法从流程外面看自己在做什么。

真 owner 得跨过这道坎。他不只知道这一步做完了没有，他知道这个系统哪里坏了。

还有一个地方特别容易混：复核 owner 和验收 owner，可以是同一个人，也可以不是。

如果 AI 输出直接决定单条业务结果——客服承诺、合同条款、候选人筛选——复核 owner 要离业务现场更近，他判断的是这一条能不能放出去。如果 AI 节点影响的是一条流程一个月的整体表现——客服平均处理时长、招聘漏斗质量、投放异常识别率——验收 owner 就要站得更高。他判断的是这个节点到底有没有把业务往前推。

复核 owner 看单点风险，验收 owner 看整体结果。单点风险没人看，错会跟着单子一路走出门；整体结果没人看，系统会一直带病上工。

所以事故这一侧真正要讲的，从来不是“谁背锅”。是组织能不能在事故发生之前，就把单点风险和整体结果分别交到两个能负责的人手上。回去对着你手上的 AI 节点点一遍名：这两个人的名字，写得出来吗？

事故复盘不是找替罪羊

AI 出了事，会议室里最先被问出口的那句话，往往是：这个 AI 为什么错？

这个问题可以问，但它只走了一半。真正该问的下一句是：我们的流程，为什么允许这个错误发生？

AI 不是外来入侵者，是组织自己把它放进流程里的。所以 AI 事故复盘不能只找模型原因，还得找组织原因。

工程界在这件事上有现成的经验可以搬。Google 有一批专门负责让线上服务不宕机的工程师，叫 SRE，他们有一套写事故复盘报告的硬规矩，名字叫 postmortem culture，直译过来是“验尸文化”[①]。

这套规矩里有一条核心原则：复盘不是惩罚，是学习机会。它要求写出来的报告是 blameless postmortem，也就是不追责到个人的复盘——只关注是什么条件促成了这次事故，然后落实预防动作，把复发的概率或影响压下去。

这套工程实践搬到 AI 组织里特别管用，因为 AI 事故最怕两件事。

第一，复盘变成找人背锅。找完一次背锅的，员工下一次就学会藏问题了。

第二，复盘变成纯技术修补。补完这个洞，组织下一次会在同一条流程缝隙里再摔一遍。

AI 事故复盘至少要问四件事。

第一，AI 是怎么被放进这条流程的？上线时有没有写明业务目标、边界和风险预判？有没有说清哪类情况必须转人工？还是 Demo 跑通了就上了？

第二，AI 在哪个节点拿到了权力？谁授权它可以直接对外输出、自动归档、推进审批、触发下一步动作？这个授权有没有业务 owner 签过字？

第三，有没有全流程记录？AI 做了什么、输出了什么、什么时候触发、用的是哪个版本的模型或规则、谁看过、谁改过、谁放行过。这些没有记录，复盘就是猜。

第四，异常为什么没有升级到人？系统有没有设异常触发条件？AI 输出偏离预期的时候，有没有被推到人工审核队列？还是异常也被自动处理掉了？

这四问问完，复盘才开始像复盘。四问一个都答不上来，那场会不叫复盘，叫事故说明会。

这四问背后，还压着一个更基本的态度问题，值得单独摊开说：会不会出错，其实不是最该关心的问题。

我不太在意 AI 会不会出错。

---

①Betsy Beyer、Chris Jones、Jennifer Petoff、Niall Richard Murphy 编，《Site Reliability Engineering: How Google Runs Production Systems》第 15 章“Postmortem Culture: Learning from Failure”（John Lunney、Sue Lueder 撰），O’Reilly，2016，全文见 sre.google/sre-book/postmortem-culture。

它会出错。这件事没什么讨论价值，就像没有人会正经讨论新员工会不会犯错一样。指望上线一个不出错的系统，本身就是一个错误的目标——它会让组织把所有力气都花在上线前的评测上，等真上线了，一点接错的能力都没有。

我更在意的是另一件事：错了以后，犯错模式有没有被抓出来，组织有没有因此长本事。

犯错模式和单次错误不是一回事。单次错误是这一封回复写错了、这一个候选人筛错了。犯错模式是：这类回复只要遇到某种情况就总是写错，这类候选人只要带着某个特征就总是被筛掉。

单次错误修的是这一件事。犯错模式修的是以后所有这类事。

组织真正的能力差距，就在这两者之间。这件事我不用举别人的例子——我自己就是那家公司。

我的内容运营系统重构过很多轮，写过多版需求文档，改过架构，每一轮都觉得这次总算理顺了。每一轮也都碰到过同样那几个毛病，每一轮都认真修了。直到有一次，我把整个系统的依赖关系摆上一张图——谁依赖谁，哪个环节是枢纽，哪条链路没人盯着，全在一张纸上。摆出来才看见：有两张关键的数据表根本没有任何人在盯。这件事我修过的每一轮都从它旁边走过去了。问题一直都在，我也一直在碰，但只有并排摆在一张纸上，我才第一次看见它们是一个结构，不是几件孤立的事。补了四次也就只补了四次——不是不认真，是没有那张能让四次同时出现的纸。

故事到这里结束的话，就太干净了，也不诚实。摆完那张纸之后，我又接连做了三轮加固，让那张图更准、更严、更防错。第三轮刚开工我停了下来，因为我意识到三轮加固全都在优化不是瓶颈的地方——真正的瓶颈从来不是图画得准不准，是它和真实代码之间的忠实度。于是我掉头，借用一个外部框架做减法，不再加固，改成删。最后砍掉一千两百多行，那个生成器从一千三百多行腰斩到五百多行。支撑这个决定的是一个数字：这套机制拦下来的二十三起事故，我翻出来做考古，二十起是它自己制造的问题，只有两起是真的。当一套机制八成以上的拦截是在拦它自己造出来的麻烦，要怀疑的就不是它的参数，是它该不该存在。

这几件事我自己复盘下来，感受其实很简单：管一支 AI 团队，跟带人的组织一样，该做的规划、该立的约束、该验的证据，半点省不了。这几次翻车之后，我给自己立了另一条规矩：框架内的优化可以外包，框架还要不要，负责人不能不问。当时我交下去的是加固任务，几轮下来都在原来的图里打转。这个教训不能被说成 AI 永远不会质疑框架；你也可以专门让它找反例、查需求是否成立。但它提出质疑以后，停不停、删什么、谁承担改道的代价，仍要回到有权作决定的人那里。不能把“这件事还要不要做”留成一个没人认领的问题。

判断标准可以很简单：面对同样的问题，下一次组织能不能不再犯同样的错误？

答不上来，说明前面那些复盘会开得再热闹，也只是在处理情绪。

顺带把一个常被绕开的问题回答掉——复盘里，供应商坐在哪个位置。

供应商可以对系统能力、服务质量和合同边界负责。这是它该负的责任，也应该在合同里写清楚。

但业务结果不能被整体外包出去。

企业买的是工具和服务，不是把组织责任一块卖出去。客户被承诺了一个不该承诺的价格，最后掏钱的是企业，不是供应商。候选人被误筛了，最后要面对质疑的是企业的招聘团队，不是模型厂商。

所以在复盘里，供应商是一个必须被列进去的环节，不是一个能让责任消失的出口。

复盘产物不是责任认定书

一场复盘开完，桌上应该留下什么？

不是一份责任认定书，是一份流程修改清单。清单上写的是这几行字：哪个节点缺复核 owner？哪段记录是空白的？哪个异常触发条件根本没设？哪次授权是默认通过，而不是有人显式点头？哪个 AI 输出说不清用的是哪个版本、依据是什么？哪个高风险场景没有升级通道？

这些才叫复盘产物。而且每一项后面都得挂一个具体改动：改规则、改权限、改知识库、改提示词、改流程节点、改验收指标。

复盘如果没有改动，就只是一次情绪释放。组织看起来很认真，下次照样犯错。

所以散会前留十分钟，把清单上每一行后面的那个改动和负责人当场写死。真正好的复盘不是为了证明谁错，是为了让组织变得更难错。

背锅和负责不是一回事

讨论责任的时候，管理者最容易把两件事混成一件：背锅和负责。

背锅，是不由自主地被拿出来承受代价。背锅的人没有主动权。事情发生之前，没有人告诉他要盯什么、可以叫停什么。事情发生之后，他成了最方便被拿出来解释的那个人。背锅不需要他事前有任何权力，只需要他离事故足够近、名字足够好找。

负责是另一回事。负责意味着事前就有权利、有义务、有能力去提示风险、调整机制、修复结果。负责的人不是事故发生后才被推出来的，而是在系统上线之前就已经被点了名。

因为他知道这个 AI 输出算在自己头上，所以他有动力在上线前问清楚边界，在运行中盯着监控，在发现异常时叫停和升级。

背锅和负责的核心区别，不是态度，是权力。

AI 进了组织，权力地图也跟着变了。过去管理者的权力，常常体现在管多少人、批多少预算、签多少流程。人机系统真正跑起来以后，真正决定结果的那几种权力，未必写在组织结构图上。

第一种是上下文权：谁定义 AI 能看什么资料、调用哪些经验、理解哪段业务语境。谁握着这个权，谁就决定了 AI 脑子里装什么。

第二种是评估权：谁定义什么叫好结果——准确率、客户满意度、风险、速度、利润和合规，这几样打架的时候听谁的。

第三种是权限权：谁允许 AI 做什么、不能做什么，什么情况下必须停下来找人。

第四种是反馈权：谁有权把错误、例外、客户反馈和复盘结论，写回规则、知识库、提示词、流程和权限里去。

这四种权力如果没分清楚，owner 就会变成背锅人。下面四句话，你对照着看你们公司有没有人正好卡在里面。

被要求对 AI 输出负责，却没有上下文权。被要求验收结果，却没有评估权。被要求叫停风险，却没有权限权。被要求复盘事故，却没有反馈权。

还要把这四种权力，和 CIO、CHO、业务线老板手上原本就攥着的老三样分开看。老三样就是前面说的管人、批预算、签流程，它回答的是“权今天在谁手里”。四种权力回答的是另一个问题：“一套人机系统要跑起来，责任人必须拿到哪些决定权”。前者是组织政治地图，后者是系统责任地图。

没有这些权力的人，只能背锅，不能负责。凡是 AI 做完以后组织要买单的地方，权和责必须在部署前当面对齐，一项一项写下来。

会议室里最危险的那句话

事故责任链最后要落到老板会上。而会议室里最危险的那句话，往往就是最先被问出口的那句：“AI 出错谁背锅？”

这句话一旦在会上被问出口，这场会就自动切进了找替罪羊模式。而四种权力一样都没拿到的那个人，往往就是最方便被推出来的那个。

要让这场会问出责任、而不是问出替罪羊，得把这句话换成六个问题。

这个 AI 输出会不会产生真实结果、真实承诺、真实成本或真实风险？这个输出的业务 owner 是谁？复核 owner 和验收 owner 是不是同一个人，如果不是，边界划在哪里？AI 出错以后，有没有完整记录能还原事故轨迹？异常为什么没有提前升级到人？复盘以后，规则、权限、知识、流程、指标，到底改了哪一项？

六问问完，一个组织有没有责任链，基本就露出来了。而且每一个“没有”都对应一种真空：没有 owner 是责任真空，没有记录是复盘真空，没有升级是机制真空，没有改动是学习真空。

从出了事找人，到出了事修系统

这周能做的事不多，但顺序不能乱。

先拿一张纸，把正在运行的 AI 系统列出来，每个系统后面写一个具体的人名——写不出名字的那几个，先别继续往外扩。

然后做一张 AI 输出后果表，逐类核对它会不会产生真实结果、真实承诺、真实成本、真实风险。命中任何一项，就得配上复核 owner 和验收 owner，两个名字都要写。

第三件事，建一个最小事故复盘模板。不用做复杂，四块就够：事故经过、组织流程缺口、要修改的机制、跟进人和日期。

这三件事做完，组织才算从“出了事找人”挪到了“出了事修系统”。

AI 会出错，人也会出错，这躲不掉。组织真正不能允许的，是每次都用同一种方式出错。

CHAPTER 13

权限、审计、留痕和回滚如何进入日常运营

很多公司一说 AI 治理，第一反应是：又要管起来了，又要审批了，又要合规了。老板心里那句没说出口的话是——刚跑起来一点，别又给我按住。

这就把方向想窄了。

治理不是为了限制 AI，是为了让 AI 可以稳定、持续、规模化地进入真实业务。没有治理，AI 只能停在三个地方：个人手机里的小工具、演示厅里的试点、少数高手自己攒的手艺。它进不了客户、合同、员工、财务、交付这些真流程。

一旦进了真流程，问题就换了一批。谁允许它做？它做了什么？谁把结果放行？错了以后谁能把它停下来？

这就是本章的任务。不写合规大全，只讲四件事：权限、审计、留痕、回滚。这四件事不性感，但它们决定 AI 能不能从“好用的小工具”变成“组织里可问责的工作单元”。

这四件事在组织内部，通常被算作成本。

加拿大航空那场官司还有一个落点没展开。不列颠哥伦比亚省的民事裁决庭在 2024 年 2 月判它为聊天机器人的错误答复担责，裁决用的措辞是“没有采取合理注意义务来确保其聊天机器人的准确性”。[①]

注意这个措辞的落点——它问的不是 AI 有没有出错，是你有没有尽到让它别出错的努力。AI 会出错是既定前提，判你输的是你没有为它出错做任何准备。复核机制、准确性校验、上线前的边界测试，这些在组织内部看是成本，在裁决庭看是你有没有尽到义务的证据。

---

①Moffatt v. Air Canada, 2024 BCCRT 149，加拿大不列颠哥伦比亚省民事裁决庭（Civil Resolution Tribunal）裁决，裁决日期 2024 年 2 月（判决书原文 CanLII 数据库访问受限，两个二手来源对具体日期记载不一致，此处仅取双方一致的“2024 年 2 月”）。裁决书原文经 McCarthy Tétrault 律所评析转录：“Air Canada did not take reasonable care to ensure its chatbot was accurate.”（加拿大航空没有采取合理注意义务来确保其聊天机器人的准确性。）

这四件事不只是内部管理动作，也是组织在出事那天能拿出来的东西。

权限不是从按钮开始的

很多企业配置权限，是从系统按钮开始的：谁能登录，谁能调用，谁能下载，谁能导出。这些当然要做。但它们只回答了一件事——系统允许谁做什么。

更要紧的那几个问题还悬着。如果 AI 输出推动了一个错误决定，谁负责解释？谁有权暂停？谁有权改规则？谁对最终结果负责？

权限不是从按钮开始的。

权限是从负责边界开始的。负责边界清楚，系统权限才有意义；负责边界不清楚，角色配置得再细，也只是一张技术表格。

真正的设计起点，是把一条已经在跑的 AI 流程倒过来看：最近一次重要结果如果出了问题，今天能不能立刻找到负责解释、复查和修复的人？找不到，说明缺的不是一个按钮，是一条结果责任链。

所以别急着打开后台。先挑一条已经在跑的 AI 流程，问一句：这条流程出了事，我找谁？这句话答不出名字，权限表配得再漂亮也是空的。

四层权限

AI 进入组织后，权限至少要从最终结果倒推四层。四层缺任何一层，治理都会留下盲区。

第一层，谁可以让 AI 接触哪些数据。客户数据、员工数据、合同数据、价格数据、财务数据，不是同一种东西。AI 能不能碰、由谁批准、在什么场景下碰，必须由组织说清楚，不能只靠系统角色自动授权。

这一层技术上早就有解。微软的云采纳框架里写得很直白：只给 AI 它完成本职工作所必需的那几个数据源，别一次性把公司所有数据都摊开给它。这句话 2026 年 4 月还在官方文档里挂着[①]。也就是说，工具厂商已经把开关做好了，剩下的是谁去按——按哪几个，由谁决定，是组织的事。

---

①Microsoft，《Governance and Security for AI Agents Across the Organization》，Azure 云采纳框架（Azure Cloud Adoption Framework），文档更新日期 2026 年 4 月 16 日。最小权限原文为“Grant agents access only to the specific data sources required for their function. Don’t provide broad access to all organizational data.”；事故响应四项原文见同页 Agent Security 第 10 条“Establish incident response plans”。

第二层，谁可以让 AI 代表组织生成什么内容。一封对外邮件、一段客户回复、一份招聘评价、一条合同修改建议，即使已经由 AI 生成，也不等于组织已经同意。生成权限解决的是“可以起草什么”，不是“可以代表谁正式表达”。

第三层，谁可以授权 AI 的建议进入下一步。生成不等于获准行动。可以按经过核验的条件预先授权，也可以要求对具体事项复核或审批；要写清允许什么、限制什么、何时退回或停止。要求逐单人审的节点，复核者必须具备信息、时间、能力和实际处置权限；不逐单人审的节点，也不能没有有效控制。

第四层，谁对最终结果负责。这一层最容易被流程绕开：前面三层都配置了，最后结果还是没人背。那就不是治理，是流程把责任洗掉了。老板要盯住这一层，不是为了管每个细节，是因为没有最终责任人，AI 会变成组织里最方便的借口。

四层权限的顺序不能反。谁承担结果，谁就应该对数据触达、内容生成和建议放行拥有相应的决定权。先从功能表分配按钮，再回头补责任人，权限就容易停在技术项目里。技术能配置谁能做什么，但不能替组织决定谁该负责什么。 还有一层，四层权限本身答不了。

权限设计的默认假设，是风险来自不该做的人做了不该做的事。所以我们防的是越界。但我自己那支 AI 团队里出过另一种事，它连边界都没越——那个 AI 做的完全是它该做的事，它想干的正是我们希望它干的那件好事。它错在一个参数上，而那个参数决定了这件好事的作用范围。更常见的翻车不是它想干坏事，是它想干好事，但那件好事的作用范围比它以为的大。

这种错，四行名字挡不住。批数据的人批的是它能看哪些数据，没人批过它一次能动多少份。它以为自己只动了手头这一份，实际动了整个工作区——而这个范围，从头到尾没有任何人和任何系统替它划过，全靠它自己估。这一层不是靠叮嘱补的，是靠边界补的。

可以先在纸上写四行：谁负责数据授权、生成范围、行动放行和最终结果。名字让问题有入口；权限是否真实、信息和资源是否够用、控制是否有效、接替和升级能否运行，还须分别核验。四行填满，不等于权限已经落地。

留痕记四件事

AI 进入流程以后，企业会积累系统日志、调用日志、模型输出和审批记录。东西很多，但日志多不等于留痕有效；如果出了事仍然只能靠当事人回忆，记录再多也没有形成责任链。

最小可用的留痕，先记清四件事：谁下达了指令，AI 接触了什么，AI 输出了什么，谁把输出推进了下一步。这四件事连起来，才能还原一个结果是怎样产生和流转的，再查问题出在哪里。第十九章会沿这条事实链反查角色安排：这里留过程证据，那里查职责能不能履行。

这四件事不必全部从零做。按微软 Purview 文档 2026 年 8 月 26 日版本，受支持的 AI 审计记录可以提供用户、时间、访问资源等线索，部分记录还包含敏感度标签和策略信息[①]。但哪些应用能记、哪些字段能取，要核对支持范围和实际配置。平台日志能帮你还原一段过程；谁把输出放进下一步，仍要接上企业自己的审批和操作记录。

没有这条链，复盘就会变成指认。组织会慢慢学到一个坏习惯：出了事，先找背锅侠，流程里那个可以修复的节点，反倒没人看。负责和背锅不是一回事。负责是事前有权、事中有记录、事后能修；背锅只是事后把一个人推出去挡一下。

留痕不是为了追责。

留痕是为了让组织能复盘、能修复、能继续用 AI。如果员工把留痕理解成一套事后处罚工具，结果往往是，他们会绕开正式流程，或者把关键判断留在记录之外。最后留下来的日志看起来很干净，却没有学习价值。

前面提到的那件事，还有更值得讲的后半段。

先把前半段补完。那个负责收尾检查的 AI，想在干净的环境里跑一次检查，于是执行了一条命令，把手头的改动临时挪开。那条命令少了一个参数，挪开的范围远超它的预期：一整天里 124 个还没存进版本库的文件，一次性全

①Microsoft，《Audit logs for Copilot and AI applications》，Microsoft Purview 官方文档，本次核查版本更新日期 2026 年 8 月 26 日，访问日期 2026 年 9 月 6 日。XPIADetected 与 JailbreakDetected 为相应检测标记；具体覆盖取决于支持的应用、记录类型及配置，不代表检出率或全量覆盖。原文：learn.microsoft.com/en-us/purview/audit-copilot。

部打回了原状。它想把改动挪回来，冲突了，失败了。接着它又做了一次清理，工作区彻底空了。一天的工作，没了。

排查的时候，我们读了操作痕迹。痕迹里有一行记录，看上去像是另一个正在并行工作的 AI 做了个回退动作。第一轮结论就指向了它。后来真正动手的那个自己站了出来——那行被误读的记录，其实是它执行的那条命令在内部自动产生的，根本不是别人的动作。

第一轮排查，我们冤枉了一个无辜的。

让它洗清的不是谁的表态，是那份记录本身。同一份记录，第一遍读错了，第二遍读对了。如果当时没有留痕，那个被冤枉的对象永远洗不清——它无法自证，而「看起来像是它干的」会成为唯一的版本。留痕的第一受益人往往不是查案的人，是那个被误指的人。组织里抗拒留痕的人，抗拒的是记录会被用来找我；这件事说明，记录同样是我没干的唯一凭证。

还有一个细节。那次抢救留下的说明里，写的还是那个错误的结论。版本历史改不掉，于是那句冤枉人的判断被永久留在了那里。留痕不美化任何人，包括记录者自己。组织常把留痕想成一面只照别人的镜子，其实它照的是所有人——第一轮判断错的是我们，那句话就留在我们自己的记录里。我不能一边要求这套机制照出别人，一边要求它放过我。愿不愿意接受这一点，才是留痕机制能不能立住的真正门槛。一套只记录别人、不记录决策者的留痕，最后会退化成监控。

只有当记录被用于修流程、改规则和改善下一次协作，留痕才会覆盖真实工作，而不是只覆盖系统里方便记录的部分。

判断你们的留痕是哪一种，有个很省事的办法：翻出最近一次因为 AI 出的岔子，看看当时复盘会上，第一份被打开的材料是日志，还是当事人的记忆。是记忆，那你们的留痕还没进流程。

审计不是翻日志山

很多公司以为审计就是保存记录。保存记录只是第一步，真正的问题是：谁看，什么时候看，看见异常以后怎么办？这些问题没有答案，日志就会堆成一座山。

山在那里，却没人爬。

审计进入日常运营，至少要抓三类异常。第一类是越权信号：AI 调用了不该调用的数据，生成了超出授权范围的内容，或者进入了不该进入的场景。这类异常不一定需要复杂算法，前提是权限边界已经写清楚，系统才知道哪条线被碰了。

第二类是反常输出：某类推荐、回复或退回率突然变化，或者结果明显偏离已核查的基线。反常不等于错误，需要按预定规则核查；关键是信号能否进入有效处置，而不是是否一律先等人查看。超出规则或授权的例外，再交给相应负责人处理。

第三类是控制被绕过的流转：本应复核的结果提前被下游依赖，超出预授权的承诺被直接发送，或原有额度限制没有生效。问题不在于这一步是不是机器在做，而在于获准执行的边界和必要控制是否仍然有效。

筛异常可以先让机器帮忙，但要知道它到底覆盖什么。亚马逊云在 2025 年 11 月发布的 CloudTrail 数据事件检测功能，会建立正常访问模式的基线，在检测到异常时生成事件；配置告警后，可以自动通知负责人[①]。微软 Purview 的审计文档则列出了跨提示注入、越狱尝试等检测标记[②]。标记由系统生成，不等于所有攻击都能检出，也不等于所有工作流都有同样的日志。

所以“谁看日志”不能只答“派个人翻”。先确认异常检测覆盖什么、漏检后果是什么，再规定哪些已知异常可以按验证过的规则处置，哪些例外要交给有权且有能力的人决定。

信号不等于已经定责。无论由规则先行限制，还是转入人工判断，都要有处置时限、运行记录和升级路径；超出授权或处理能力，就不能继续沿用原有放行方式。

真正要核的是：什么信号触发什么动作，谁负责该规则与例外，多久内完成，处置是否有效。只保存一个告警、只签一个名字，不能代替这些检查。

---

①Amazon Web Services，《AWS CloudTrail launches Insights for data events to automatically detect anomalies in data access》，AWS 官方公告，2025 年 11 月 20 日。公告区分自动生成异常事件与配置告警通知；这里不将其等同于企业全部 AI 流程的检测覆盖。原文：aws.amazon.com/about-aws/whats-new/2025/11/cloudtrail-insights-data-events-detect-anomalies-access/。

②Microsoft，《Audit logs for Copilot and AI applications》，Microsoft Purview 官方文档，本次核查版本更新日期 2026 年 8 月 26 日，访问日期 2026 年 9 月 6 日。XPIADetected 与 JailbreakDetected 为相应检测标记；具体覆盖取决于支持的应用、记录类型及配置，不代表检出率或全量覆盖。原文：learn.microsoft.com/en-us/purview/audit-copilot。

回滚不是代码版本

回滚这个词很容易被技术化：系统版本错了，退回上一版；模型配置错了，恢复原参数。这些是技术回滚，但组织里的 AI 回滚不止这些。AI 已经发出去的客户承诺、影响过的招聘筛选、推动过的审批结果，以及改变过的员工信任，都不会随着版本恢复而自动消失。

这里要分开两件事：恢复原状，和承接后果。能停机、能补偿、有人来处理，不代表原先的影响已经消除；是否能恢复，仍按第 11 章事前固定的目标、窗口、成本与剩余影响界线来判。

承接至少要核四个动作。第一，能停止或限制进一步损害；来不及中途叫停的动作，要在事前限制作用范围。第二，必要服务能继续，或者能有序停止；替代路径可由人工、规则系统、备份系统或其组合承担，但要验证真实能力。

第三，能追溯已受影响的结果，明确哪些需要复核、通知、纠正或其他处置。第四，能修正规则，并检查修正是否生效。有人接手是处置的开始，不是恢复完成的证明。

微软的 AI 治理文档把关停、对外沟通、日志保存和关键业务演练列为事前安排[①]。它们提供响应准备的检查项，不证明上一段的四项动作在任何系统中都已具备。

要提前明确谁能关、关掉后怎么办，也要实际演练；不能等到出事才决定。 这四个动作，我自己的团队被真考了一遍。

替代路径确实启用了：快照里找不回的那部分，交给另外两个 AI 分头重做，一个照着幸存下来的设计文档，一个照着完工报告。这不是全部退回纯人工，而是重新组织恢复工作。影响范围做了逐个文件的比对，事后也固化了两条规则：让并行工作的 AI 在物理隔离的工作区里干活；重要状态尽早存档，哪怕是半成品。这些动作确实做了，但不能据此说全部重放已经验收完成。

①Microsoft，《Governance and Security for AI Agents Across the Organization》，Azure 云采纳框架（Azure Cloud Adoption Framework），文档更新日期 2026 年 4 月 16 日。最小权限原文为“Grant agents access only to the specific data sources required for their function. Don’t provide broad access to all organizational data.”；事故响应四项原文见同页 Agent Security 第 10 条“Establish incident response plans”。

暂停这一问，在事故当天已经来不及回答。等发现的时候，破坏已经完成了，这类操作没有中途叫停的窗口。这正说明作用范围和保护措施为什么要立在事前，不能把“到时按暂停”当成一定存在的退路。

找回来靠的也不是什么备份策略，是两样恰好留在那里的痕迹：操作日志，和那条命令自己留下的一份没被清掉的中间快照。文件是从这份快照里逐个捞回来的。另有一份改动前的状态记录，它保住的是另一侧的数据没跟着受损。那些设计文档在事故前看起来只是流程要求的副产品，事故那天成了唯一的复原依据。收尾的时候还撞上一件事：网络断了，救回来的东西发不出去。我下令先就地存档再重启机器。但也别把这讲成完美复原——重放的收尾没走完就断电了，有几件是不是全部落地，当时并没有终核。

Gartner 在 2026 年 3 月给出一个预测：到 2028 年，企业网络安全事故响应的一半工作量，会集中在涉及定制开发的 AI 应用的事故上[①]。请注意这句话的边界——说的是网络安全类事故的响应工作量，不是事故数量；说的是涉及这类应用，不等于事故全由它们引发。这是预测，不是已经发生的结果。对管理者来说，不必等比例兑现，才去安排权限、审计和事故响应。

人不是一份写进预案就会自动生效的备份。

如果某项判断需要由人接手，就验证这个人是否还有能力、时间、信息和权限。如果其他替代路径更合适，也要验证它能覆盖什么、依赖什么，和原系统会不会同时失效。人工形式可以变化，承接能力不能凭一句“系统已经很稳定”被撤掉。

挑一个正在运行的节点，分别问：异常能否及时限制，必要服务如何维持，已经发生的后果由谁按什么方案处理？这些答案需要证据。三句都答得出来，也还不能把不可恢复的后果改叫可恢复。

①Gartner，《Gartner Predicts AI Applications Will Drive 50% of Cybersecurity Incident Response Efforts by 2028》，2026 年 3 月 17 日新闻稿，2026 年 9 月 6 日回核官方正文。原义为响应工作集中于涉及定制开发 AI 应用的事故，不是这类应用引发一半事故；custom-built 也不限定为企业内部自研。原文：gartner.com/en/newsroom/press-releases/2026-03-17-gartner-predicts-ai-applications-will-drive-50-percent-of-cybersecurity-incident-response-efforts-by-2028。

100 人公司的最低治理规则

大型企业可以配置法务、安全、合规和数据治理委员会。很多 100 到 500 人的企业没有这些条件——法务是外聘的，安全是 IT 兼的，合规平时没人管，数据治理委员会听都没听过。

但 AI 风险不会因为组织规模小就自动缩小。只要关键流程已经接入 AI，治理就不能等到资源齐备以后再做。等齐备，就是等出事。

先做最低版本。

第一，关键 AI 流程必须同时固定事实链和结果接球人。前者让流程可复盘，后者把责任落到人；两者不能混为一谈，也不能缺任何一边。

第二，对客户承诺、员工评价、财务动作和合同相关输出，先拆清起草、建议和实际生效的节点，再核权限、适用限制、风险及控制证据。对固定动作，可以评估是否采用预授权下的自动执行；具体放行仍须满足该场景的完整要求，不能把本书列出的必要检查项当充分条件。具体例外或承诺超出边界，就交有权者决定。不能把“有钱”直接等同于必须逐单人审，也不能用一纸授权替代有效控制和后果承接。

第三，关键流程必须有与其后果相匹配的限制、响应和服务连续性安排。能中途停止的要验证停止是否有效；来不及停止的要事前限制作用范围。替代路径和后果处置都要核能力，不预设只剩人工，也不假定人必然接得住。

第四，留痕必须能支持复盘。记录不是为了堆日志，是要能还原责任链；至少要把“谁下达指令、AI 接触什么、AI 输出什么、谁推进下一步”固定下来，成为出事以后可用的证据链。

第五，按流程变化速度和损害窗口安排复评；每月、每季度可以是起点，不是所有场景的固定周期。用途、权限、输入或规模明显改变时及时重评，发现越界则先限制相关动作，不等例会。

这五条是起点，不是完整的 AI 治理体系。Papagiannidis 等人的责任 AI 治理综述，把实践放在结构性、程序性、关系性三个方面看[①]：权利和责任怎么配

①Papagiannidis, E., Mikalef, P., & Conboy, K. (2025). “Responsible artificial intelligence governance: A review and research framework.” The Journal of Strategic Information Systems, 34, 101885. DOI: 10.1016/j.jsis.2024.101885。本文借其结构性、程序性、关系性治理实践的分类作检查视角；原文将框架与所提关系留待进一步实证检验。本章最低规则、具体检查问题及控制方式，不是该文验证过的处方。

置，要求怎样进入开发、运行和异常处置，相关的人有没有理解和参与的条件。

借这个视角回看最低版本，就会多问几步：表里写了负责人，他是否真有信息和权限？制度写了升级路径，异常来了是否走得通？员工、客户或供应商提出了问题，有没有人回应，是否会因此调整安排？这些是本书把框架放进现场的检查问题；三个方面不是给五条规则贴上的三个新标签，也不能靠勾完一张表证明治理有效。

可以先拿一条流程逐项核查，缺口再按后果和资源需求排序。检查可以从一张纸开始，落实权限、能力、监测和承接却可能需要预算、岗位时间与系统改动；不能把列出问题当作已经解决。

治理怎么进入日常

很多治理方案失败，不是因为规则写错，是因为规则只在文件里。文件写完以后，业务继续跑，系统继续改，员工继续用，新场景继续冒出来；一个月以后规则开始过期，三个月以后规则和真实业务已经不是一回事。

所以 AI 治理要进入日常运营。可以沿用已有台账，至少把流程名称、AI 节点、数据权限、输出及后果边界、授权与控制安排、留痕和监测位置、异常处置及恢复/承接方案接在一起。控制安排中记清自动执行的条件、需人工处理的例外、各自负责人和重评触发；“放行 owner”不等于每单都要该人点击。

台账字段	沿同一条流程核查
流程名称	核查对象、业务 owner 及结果范围
AI 节点	输入从哪里来，输出交到哪一步
数据权限	谁可访问什么，依据和限制在哪里
输出及后果边界	哪些输出会生效，影响谁、影响到哪里
授权与控制安排	自动执行条件、人工处理例外、负责人和重评触发
留痕和监测位置	指令、数据、输出、采用记录和异常信号在哪里查
异常处置及恢复 / 承接	谁能行动，用什么资源，怎样确认处置完成

这七项沿用同一份台账，不另开七个账本。业务 owner 看结果责任，CIO 看系统权限和留痕能不能落地，CHO 看员工角色、信任和复盘机制有没有被接住，一号位看这条流程到底是不是还在裸奔。角色不同，看的是同一条责任链。

再按损害窗口和变化速度安排运营节奏。每周看异常、每月看规则、每季度评估扩大与迁移，可以作为待调整的初始安排；需要即时响应的异常不能等到周会，长期稳定的低后果用途也不必机械增加会议。

“定期回到桌面上”这件事，也不是我一个人的主张。美国国家标准与技术研究院（NIST，美国政府里专门给各行业定技术标准的那家机构）那套 AI 风险管理框架里有一条讲的就是这个：系统上线之后，得有一套持续跑着的机制，保住它当初被寄予的那个价值。具体说就是持续监控、异常检测、定期重新评估[①]。翻译成老板听的话——AI 不是装完就完的设备，是需要有人定期回来看

①美国国家标准与技术研究院（NIST），《人工智能风险管理框架》（AI Risk Management Framework 1.0，NIST AI 100-1），2023 年 1 月 26 日发布，MANAGE-2.2 条款：“Mechanisms

一眼的东西。国际上写框架的人都把它写进条款了，说明这不是哪家公司的内部讲究。

欠账越久，事故越贵。

所以这一节先把台账接进实际运行：可以只从一条流程开始，明确监测与例外由谁负责、按什么条件行动、怎样确认处置完成。检查节奏依损害窗口和变化速度来定，不把“每周看一次”当成通用答案。

一个通用流程怎么填

拿 AI 客服辅助作一个限定授权的假设例子：本次只允许识别问题类型、按获准范围调取记录、生成供客服参考的草稿，不授予自动发送或赔付权限。这是下面流程的给定条件，不是所有客服场景唯一的控制方式。

这条流程的权限边界要写得足够具体。AI 可以看客户的服务记录，但不能擅自承诺赔付。可以生成话术，但对外发送前必须由客服确认。可以提示风险，但投诉升级和补偿方案必须由有权限的人决定。

这样写，AI 能做什么、人必须判断什么，才不会在上线以后靠默契维持。默契是最贵的管理成本——它平时不要钱，出事那天一次性收。

留痕同样要沿着结果流转记录：谁发起查询，AI 调了哪些客户记录，给了什么建议，客服采用、修改还是拒绝，最后客户收到的答复是什么。只有把 AI 输出和人的放行动作连起来，复盘时才知道问题来自数据、建议、判断还是最终发送。

回滚也要提前写清。发现 AI 引用了错误记录，暂停这类自动推荐。发现某类话术误导客户，撤回模板，人工复核最近一批同类回复。发现客服长期变成橡皮图章，就要重设复核要求，把真实判断请回来，别留一个只有形式的人审节点。

这不是一句“暂停”就结束。比如系统发现一条本应人工确认的补偿建议已经被直接发出，先暂停同类对外发送；再由客服负责人复核最近一批同类记录，联系受影响的客户并改正；最后把越权路径、处置结论和新的放行规则写回治

---

are in place and applied to sustain the value of deployed AI systems.”。此处表述据 NIST 官方 Playbook 页面（airc.nist.gov）核实，为框架条款的官方诠释版本。

理台账。异常只有走完“发现、暂停、复核、纠正、改规则”这五步，才真的回到组织的控制回路里。

这就是治理进入日常。它不需要什么大词，就是把 AI 能做什么、人必须判断什么、错了以后怎么停，提前写进流程。写不进去，就别急着上线。

写进去，AI 才有机会成为组织能力。

CEO、CIO、CHO 怎么分工

这件事不能只交给 CIO。CIO 负责的是系统能不能安全、稳定地跑起来。可这四件事，每一件都跑到 IT 的边界之外去了。权限边界背后是责任划分。留痕背后是绩效和信任。审计背后是复盘机制。回滚背后是岗位和流程韧性。这些没有一件是 IT 一家能定的。

CHO 也不能缺席。不是因为 CHO 要管技术，是因为 AI 治理会改变人怎么工作、怎么被评价、怎么贡献经验、怎么承担责任。留痕不只是技术日志，也是组织学习的原材料；哪些行为值得记录，记录如何用于复盘和改进，必须有组织视角参与定义。

如果 CHO 不在场，留痕很容易变成员工监控；如果 CIO 不在场，组织规则落不到系统里；如果 CEO 不在场，部门之间就容易互相踢球。CEO 要定原则，CIO 要搭底座，CHO 要设计人和组织怎么接住，业务 owner 要对真实结果负责。

这四个角色坐不到一张桌子上，AI 治理就会散。

散到最后，就是系统能跑，组织接不住。

今天就能问的五个问题

管理者今天可以先做一件小事：找一条已经在跑的真实 AI 流程，不必挑最复杂的，直接问五个问题。

谁下达了这条指令？AI 接触了什么？AI 输出了什么？谁把输出推进了下一步？出了问题谁来接球？

这五个问题能答出来，才有继续检查的入口。还要核实权限是否实际生效、记录能否重建过程、异常能否得到处置；答得出人名和制度，不等于已经有可用的治理底座。

任何一个问题答不上来，都意味着流程存在治理空白。其中最该优先补的是最后一个——“出了问题谁来接球”。没有人对结果负责，前面的记录和规则都可能变成摆设。

不要急着再买工具，也不要急着再做培训。先把这五个问题补齐，把能做的最低版本跑起来，再随着场景和风险变化逐步迭代。

AI 真正进入组织，不是因为它能自动跑。

是因为授权在实际运行中有效，放行和异常处置可以核验，规则有人更新，结果有人持续承接。

这才叫治理。

PART 05 · 第五部

产能、绩效和信任

AI 释放产能以后，一号位最容易动的念头是裁员。

这个念头不是不能讨论，但一省时间就裁人，是最危险的判断。因为 AI 提效以后，真正该看的不是“省了多少小时”，而是组织结果有没有变得更稳定、更多、更少返工、更可复用。过程指标很容易骗人。使用次数、写了多少 prompt、烧掉多少 token、在线时长——prompt 就是员工给 AI 下的那句指令，token 是模型算钱的计量单位，像打车表上跳的那个数。这些数字都不能直接证明组织能力升级。

结果才是口径。同样一组人，能不能接住更多业务？同样一个部门，产出是否更稳定？原来靠老员工脑袋里的判断，现在能不能进入流程和知识系统？发现风险、修复系统、沉淀方法，这些贡献能不能被看见？

员工也不是傻子。如果他相信贡献业务上下文是为了替代自己，他就不会贡献真实业务上下文。

所以产能分配不是一个财务题，它也是信任题、绩效题、组织能力题。这一部要算的，就是这笔账：

AI 释放出来的产能，到底应该怎么入账？

CHAPTER 14

AI 上完以后，到底裁不裁员

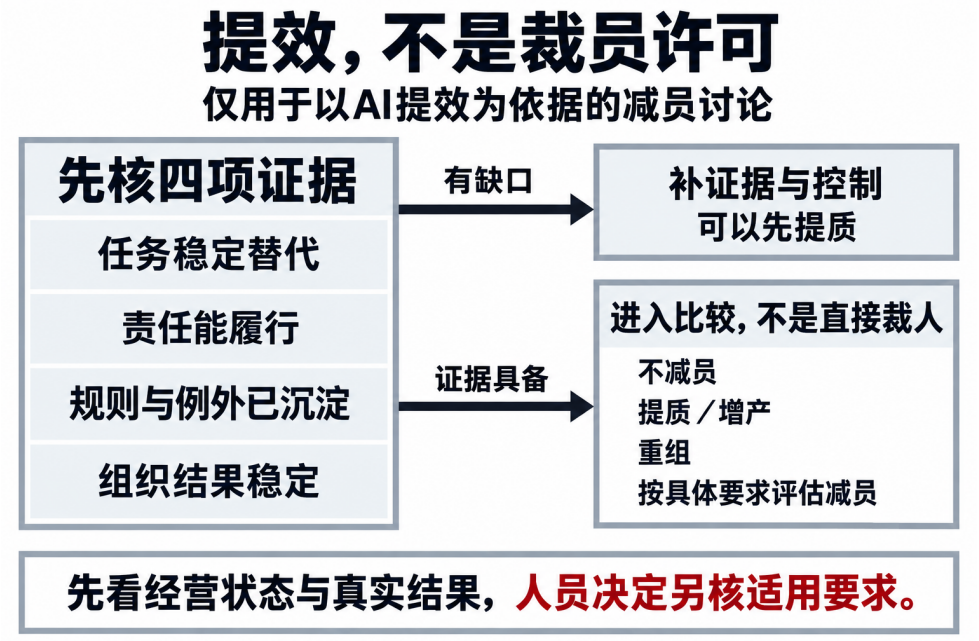

图 R10-F10｜提效，不是裁员许可。仅用于以 AI 提效为依据的减员讨论。证据有缺口，补证据与控制，也可以先提质；证据具备才进入各方案比较，不自动取得裁员许可，仍须按具体要求核查。

会议开完，人都走了，老板把我留下来，关上门问了一句：“这套东西上了，人是不是可以少一点？”

这句话，我这两年听过很多遍。开会的时候没人问，散会以后一定有人问。

这个问题不脏。企业要活下去，成本当然要算；AI 如果真的提升效率，老板当然会问效率红利归谁。真正危险的是，流程、知识、责任和信任都没想清楚，就把 AI 当成裁员按钮。这会把一个组织问题，粗暴地压成一个人力动作。

AI 提效以后，一号位面对的不是“裁还是不裁”这道是非题，是一道产能分配题。哪些动作被 AI 接走了？哪些判断还必须由人承担？哪些经验还留在老员

工脑子里？哪些责任尚未迁移？释放出来的时间，到底要换成本下降、收入增长、服务提质，还是组织重组？

这不是人力题，是经营题，也是组织题。如果一号位只问“能不能少几个人”，公司很容易省下一段工资，同时删掉一段组织记忆。

省下的是这个月的数字，删掉的是那个只有老张知道的客户脾气、那条没人写下来的返工规矩、那句“这单不能这么接”的直觉。短期账好看，长期系统却会变脆。

裁员不是起点

裁员不是 AI 转型的起点，是产能重新分配之后的结果变量。这句话必须放在前面。

2026 年 5 月，Cloudflare 官方博客公告裁减 1100 多个岗位，而在这之前大半年，2025 年 9 月，它还公告过一个 1111 人的实习生录用计划[①]。两个数字一放在一起，外面立刻吵成“AI 裁员，用实习生对冲”。老板们看这条新闻，脑子里冒出来的问题多半是：我们什么时候也得这么干，要裁多少，实习生能不能顶上？

这两个问题都问错了。官方原文里那句最该被抄下来的话不是数字，是这一句：重新想象每一条内部流程、每一个团队、每一个角色。它没说先裁多少人，它说的是先把公司每条判断链重新问一遍——哪些判断可以交给机器，哪些必须留在人身上，哪些要人机混着来。裁员是那一轮重问之后掉下来的结果，不是那一轮的开场。

道理放到你自己公司也一样。AI 能让某些动作变快，让某些材料生成得更便宜，让某些检索更及时，也能提前提示某些重复判断。但 AI 不会自动替代责任。一个岗位里有任务、判断、协作、经验，也有后果承担。AI 可以先接走一部分任务，不等于整个岗位立刻消失，更不等于岗位背后的组织记忆、客户关系、例外判断和风险兜底可以一起消失。

①Cloudflare 官方博客《为未来而建》，2026 年 5 月 7 日发布，网址为 blog.cloudflare.com/building-for-the-future/。实习生计划见同站 1,111 人实习生计划公告，2025 年 9 月 22 日发布，网址为 blog.cloudflare.com/cloudflare-1111-intern-program/。

所以裁员之前，一号位必须先问四件事。流程稳了吗？知识沉淀了吗？责任迁移了吗？结果稳定了吗？这四件事没过，裁员就不是战略动作，只是把没做完的组织设计，转手做成了这个季度的财务收益。

没有证明组织结果稳定之前，裁掉的不是成本，可能是组织能力。我不反对裁员，我反对的是粗糙。所以你先别翻花名册，先把公司最核心的那条流程调出来，看看它今天还是不是靠人扛着。

AI 释放的是产能

AI 释放出来的首先不是编制，是产能。产能有好几副面孔。

第一副面孔是时间。以前一个人查资料、写初稿、整理表格、做复盘要半天，现在可能只要几十分钟。第二副面孔是等待。以前材料要等人汇总、等主管复核、等跨部门确认，现在部分信息可以提前整理，部分风险可以提前提示。第三副面孔是知识。以前只有老员工知道某类问题怎么处理，知识一旦被结构化，新人也能更快接近基本判断。第四副面孔是注意力。以前一个主管一天被十几个群消息切碎，从早到晚都在回消息，现在能留出两小时想一件事。人不再被低价值的重复动作占住，可以转去处理例外、客户、风险和更复杂的判断。

这四样东西都是真的省下来了。但它们不会自动变成降本，它们必须被分配。不分配会怎么样？现场会告诉你答案：小李原来一天写三份方案，现在两小时写完，然后主管顺手又给了他四件杂事。他做得更多，下班更晚，工资没变。公司以为自己提效了，现场只觉得节奏更碎、压力更大、边界更乱。

这就是隐形加班。产能被释放了，但没人认领，于是它自己找了个最省事的去处：塞回员工身上。

所以一号位不要先问少几个人，先问这部分产能要换什么。

四种产能去向

AI 释放出来的产能，至少有四种去向。

第一种是降本。需求稳定、流程标准、任务能验收，责任链重写完了，知识也沉淀下来了——这时候才可以讨论减少人力投入、冻结新增编制，甚至裁员。注意，是讨论，不是默认。

第二种是增产。市场需求还没被满足，客户还在排队等，销售跟不上，交付也跟不上——这时候同样一拨人可以服务更多客户、更多项目、更多场景。这不是少人，是把效率红利换成收入。

第三种是提质。有些行业拼的根本不是人少，是更快、更准、更稳、更少出错。做医疗器械的，一次召回够你亏一年；做高端服务的，一次响应慢了客户就走。AI 省下来的时间，在这些行业里换成更高的服务标准、更低的错误率、更完整的复盘，比换成裁掉两个人值钱得多。

第四种是重组。原来的岗位被拆开，任务、判断、复核和知识维护重新分布。这时候要做的不是简单减人，是重写岗位边界、合并职责、设置新的责任节点。

这四种都是组织动作，裁员只是其中一种。把四种动作混成一个“裁不裁员”，是一号位偷懒。所以下次开会你先别问人数，先在白板上写这四个词，然后指着问一句：我们这次，主要是要哪一个？

两种公司，两种答案

同样一个问题，两家公司应该得到两个答案。

第一家公司，经营不善，现金流紧，需求在往下走，组织本来就撑不住。AI 上来以后确实稳定替代了一类动作，结果也没有变差。这种情况下，降本可以讨论，甚至应该讨论。企业不是慈善组织，组织能力最终也要服务生存。

第二家公司，业务还在涨，客户还在增加，原本 100 个人接不住 150 个人的生意。这种情况下，AI 的第一目标未必是裁人。更合理的答案可能是：先不招那 50 个人，让现有 100 个人接住更大的业务。先把交付做稳，把客户响应变快，把新人上手周期缩短。这一样是降本，只不过降的方式不是粗暴裁员，是避免低质量膨胀。

同一套 AI，两家公司，两个完全相反的动作。差别不在技术，在经营状态。

所以真正要回答的不是抽象的“裁不裁”，是三个问题。你的经营状态是什么？你的组织结果稳不稳？你的产能要换什么？这三个答案不同，动作就不同。你先对号入座，看看自己是哪一家。

裁员讨论门槛

如果要以 AI 提效为依据讨论减员，至少先查四道门。

第一，任务可稳定替代。不是 Demo 里替代一次，也不是某个员工用得很顺，是在真实流程里稳定替代一类动作。

第二，责任链已重写。谁使用、谁复核、谁验收、谁维护、谁兜底，都已经写清楚。

第三，知识已沉淀。被替代动作背后的规则、例外、踩坑和判断依据，不能还只在某个人脑子里。

第四，组织结果更稳定。不是“看起来快了”，是产出更多、波动更小、返工更少、客户更稳、新人更容易接上。

四道门没过，裁员就很危险。

危险在你不知道自己裁掉的是什么。是重复动作，还是复核能力？是低价值时间，还是关键经验？是冗余岗位，还是组织最后一道异常处理能力？一号位可以追求效率，但不能拿模糊的效率，去换确定的组织脆弱。

这里最容易错的是顺序。错误顺序长这样：先定裁员目标，再倒推 AI 能替什么，最后逼组织证明自己可以少人。这条路走下去，员工一定会进防御状态。因为所有人都看得出来，这个 AI 项目不是为了让组织变强，是为了找人头。

如果要以 AI 提效为依据讨论减员，顺序恰好相反：先核任务替代和结果，再核知识与责任承接。证据具备，只是取得比较基础，不是拿到裁员许可。提质、增产、重组和不减员都要放在桌上，人员决定还须满足具体适用要求。质量尚不稳定时，也可以先做提质，不必等到减员讨论的条件全部成立。

一号位如果一开始就把答案写成“少多少人”，下面的人会自动配合这个答案。流程风险会被低估。知识缺口会被遮住。员工信任会被透支。真正重要的问题反而没人敢讲。我不反对老板算人力账。我反对的是，组织还没证明自己接住了 AI，老板已经开始兑现裁员收益。那不叫战略，那叫提前透支。

最贵的是裁掉组织记忆

最贵的裁员不在赔偿金那一栏，而在被一起裁掉的组织记忆里。

很多关键经验并不写在制度里。它在老员工脑子里，在项目复盘里，在客户异常里，也在那些“这事不能这么办”的直觉里。AI 可以读取文档，但如果这些判断从来没有被抽出来，AI 就读不到。

举个具体的。一家做工业设备的公司，售后有个干了十二年的老师傅。他看到某个客户报修单上写着某型号加某工况，不用去现场就知道八成是装配时那颗螺栓没按扭矩上——因为五年前有一批货就是这么出的事。这条判断，制度里没有，SOP 里没有，知识库里也没有。它只在他脑子里。他走的那天，公司省下一份工资，同时丢掉了一条只有他知道的排查捷径。下一次同样的故障进来，新人要跑一趟现场，拆一次机，多花三天。

这时候裁掉一个人，表面上是减少成本，实际可能是删除一段判断历史。这不是说老员工都不能动，任何组织都要更新人。但动之前，一号位要问一句：这个人走了，哪类判断会跟着走？哪些客户例外没人知道？哪些流程坑没人记得？哪些复核习惯没人接？哪些新人训练材料还没有沉淀？

这些问题没人回答，裁员就不是清理冗余，是在把组织的隐性记忆直接格式化。省钱是真的，变傻也是真的。所以名单定下来之前，你先把每个名字后面那条“只有他知道的事”写出来——写不出来，说明你还没资格动他。

中间策略不是软

很多组织更现实的第一步不是裁员，是另外五个动作。

冻编，是先不让组织继续按旧模式膨胀。自然替代，是利用离职、退休和组织调整的自然流动，慢慢减少对旧岗位的依赖。转岗，是把还能学、还能接判断的人挪到新的流程节点上去。重训，是让人从重复动作转向复核、异常处理、客户判断和知识维护。岗位重写，是承认原来那个岗位包已经不适合 AI 时代的工作颗粒度了。

这五个动作听起来都没有裁员快。但它们有一个共同的好处：能回头。一号位做组织动作，不能只看当月利润表，还要看组织是不是被自己打断了学习能力。AI 项目刚上线，流程还在磨，知识还没沉淀，员工还在学，这时候硬裁，省下来的钱会被返工、客户损失、管理摩擦和信任损耗慢慢吃回去。走中间策略不是心软，是在降低不可逆损失。

对很多中型企业来说，最稳的第一步不是“裁掉谁”，是“别再按旧方式增加谁”。业务增长很好，原本必须继续招人才接得住交付——这种时候，AI 的价值可以先体现在三个字上：少招、慢招、准招。

少招，是让效率红利抵消掉新增编制的需求。慢招，是让组织先看清哪些岗位真的还需要扩。准招，是把新增的人力投到 AI 还接不住的地方去：判断、客户、异常、业务开发。

这三件事比裁员更适合当第一阶段动作。它既保留组织信任，也保留经营弹性。试点证明结果稳定，再往降本走也不迟。试点证明需求更大，组织还能把释放出来的产能导向增长。

这就是一号位真正要控制的节奏。不是不动，是先不要做不可逆动作。这个月你能定下来的最有价值的一件事，可能就是把新增编制先冻上，给自己留一个季度看清楚。

效率红利必须入账

不裁员不等于没动作。恰恰相反，不裁员的时候，效率红利更要入账。我建议一号位至少看三张账。

第一张是时间账。哪些工作少花了时间，少的是谁的时间？是员工少查资料，主管少复核，还是客户少等待？这三个人省下来的时间，价值完全不一样。

第二张是等待账。哪些流程少等了？审批、跨部门确认、资料汇总、返工、客户响应，究竟是哪一段真的变短了？以前一张单子从提交到批下来要走五天，现在是三天还是两天，得有个数。

第三张是知识账。哪些原来靠老员工带的判断，现在可以被知识库、规则库、案例库接住？新人从入职到能独立处理一类问题，原来要三个月，现在是几周？

这三张账不记，员工用 AI 就会变成一场热闹。大家分享工具、写心得、晒效率，年底一看经营结果没变化。老板看不到账，就会回到最粗的那个动作：那还是裁人吧。

所以不裁员不是慈善，也要交经营结果。同样的人服务更多客户，是结果。同样的人交付更稳定，是结果。同样的人让新人更快上手，是结果。同样

的人让返工更少，也是结果。AI 释放出来的红利必须进入账本，不然它迟早会被粗暴地折算成人头。这个月你就可以先记一张：挑一条流程，记下它现在实际花多久。

员工信任成本

AI 转型还有一笔很多老板不爱算的账：员工信任成本。

员工不傻，他会算一笔很简单的账。我越会用 AI，活干得越快，然后活越多，压力越大，最后可能人还更少——那我为什么要越会用？

算清楚这笔账之后，他会怎么做？他会留一手。少分享技巧，把 AI 用成自己的私人外挂，不是组织的能力。他会在形式上配合，在真实判断上后退。

这不是员工道德问题，是激励问题。没有人会持续贡献高质量判断，去证明自己可以被替代。人机复核那个场景最典型。复核的人如果没有足够的信息、时间、权限和纠偏通道，所谓“人在回路中”就会退化成一个人工确认章。AI 给建议，人点确认，出了问题追人。这种机制消耗信任的速度非常快。

信任不是软问题，是 AI 能不能真正被用起来的生产条件。这件事在业内有个词叫 AI adoption，指的是这套东西有没有真被人用起来，而不是只被买下来。员工愿不愿意把经验拿出来，愿不愿意复盘失败、修正流程，愿不愿意把个人技巧沉淀成组织知识，全都和信任有关。如果 AI 提效最后只变成裁员压力，表面效率上去了，组织的真实学习却停了。

更隐蔽的风险是员工开始反向优化。公司看 AI 使用率，他就提高使用率。公司数 prompt 数量——prompt 就是员工敲给 AI 的那句指令——他就疯狂堆 prompt。公司看产出数量，他就交更多半成品。公司要抽取经验，他就只交安全的那部分经验。

这不是员工坏，是指标把人训练成了这样。

组织只奖励产量，不奖励发现风险、修复流程、沉淀知识，AI 就会把错误的指标放大一遍。最后你看到的不是组织进化，是一张漂亮报表。所以员工信任和绩效口径必须一起改。贡献高质量判断要被看见。发现 AI 风险要被看见。把个人技巧沉淀成组织方法要被看见。修复流程缺口也要被看见。

不然员工没有理由把最值钱的那点业务背景交给组织。他只会把 AI 用成自己的护城河，组织还是学不到。

一号位产能分配决策表

AI 上完之后，一号位不要先问裁多少人，先在纸上写六个问题。

第一，哪个动作被 AI 稳定替代？不是哪个岗位，不是哪类人，是哪个动作。

第二，哪条责任链已经重写？谁使用、谁复核、谁验收、谁维护、谁兜底？

第三，哪些知识已经沉淀？规则、例外、案例和复盘，有没有从人脑进入组织系统？

第四，组织结果是不是更稳定？产出有没有更多，波动有没有更小，返工有没有更少，客户体验有没有更稳？

第五，效率红利进入哪张账？时间账、等待账、知识账、成本账、收入账、风险账，至少要进一张。

第六，产能去向是什么？降本、增产、提质、重组，先选一个主动作。

这六个问题答完，裁员才有讨论基础。答不完，就不要把 AI 包装成战略，实际做成粗糙减员。

写完这页纸，你要做的第二件事是把它带到经营会上，别让 HR 单独拿着讨论人头。这张桌子上要坐四个人。业务负责人证明结果稳不稳。CHO 证明岗位、信任和能力有没有接住。CIO 证明系统、数据和流程有没有跑稳。CEO 最后拍产能去向。

没有这张桌子，裁员讨论很容易变成部门自保和财务压数。桌子先搭起来，后面的账才不会算歪。

AI 不是裁员按钮，是组织重新分配产能的压力测试。最差的老板会先问裁多少人；真正的一号位会先问：这部分产能要换什么？

CHAPTER 15

释放出来的产能到底怎么分配

“我们 AI 上得挺好，大家都说快了不少。”一位老板在会上这么说完，停了两秒，又补了一句：“但我这个月看报表，好像跟去年也没差多少。”

这两句话之间那道缝，就是这一章要讲的东西。

AI 确实省时间。认真推过 AI 的企业，往往都能感受到：流程快了，材料快了，复核快了，有些岗位上一个人能处理的量也明显上去了。

但省下时间以后，企业马上会遇到一个更难的问题：这些时间去哪了？

这个问题一点都不玄。写一份材料从半天变成一小时，那省下的三个小时，员工不会拿去发呆。它一定被什么东西填上了。填上它的可能是一个新客户的方案，也可能是三个临时通知、两个没人整理的表格、一堆群里飘过来的琐事。前一种是产能进了账，后一种是产能被消化在噪音里。

产能释放以后，必须流向某个地方。如果没人管，它不会自动变成利润，只会被碎片化事务重新填满，变成员工继续忙，变成老板以为公司提效了，经营结果却没有变化。

所以个人省时间，不等于组织有产能。一个员工用 AI 写材料，从半天变成一小时，这是个人效率。省下来的时间最后换成了什么，才是组织的账。是更多客户、更少返工、更快交付、更稳定的新人上手，还是只是被更多临时任务吃掉？如果没人回答这一句，产能就没有进入组织账本。

产能要先入账

不管最后选降本、增产、提质还是重组，第一步都不是拍方向，是入账。时间账、等待账、知识账这三张，第十四章已经摆过一遍，这里只补一张容易漏的：质量账——产出是不是更稳定？错误率有没有下降？返工有没有减少？客户投诉有没有变少？

这套产能入账四张账单不变，AI 提效就只是一种感受。感受当然重要，但不能直接拿来做组织决策。老板说“感觉大家快了很多”不够，员工说“我现在顺

手多了”也不够。组织必须回答：哪条流程、哪个节点、哪类产出，稳定变好了？

这才叫产能入账。

产能入账四张账单不用等季度会才算。你现在就可以挑一条上了 AI 的流程，逐张问一遍：时间省在谁身上、哪一段等待变短了、错误和返工有没有少、哪块判断从老员工脑子里挪进了库里。四账里有一张能说清楚，你手上就有真东西了；四账都答不上来，那目前只有感受。

抓住投入和产出两端

证明 AI 释放了产能，不能只听员工说快，要同时抓住投入和产出两端。

投入端看三样：人力、时间、成本有没有变，是少了，还是只是转移到了另一个人那里。产出端也看三样：完成量有没有增加，质量有没有稳定，交付有没有准时。

我看过一个营销服务组织的投放管理场景。真正难的不是让 AI 给建议，是把建议放进流程：谁看、谁改、谁复核、谁升级、谁把复盘写回知识库。这条链不接上，两端就都动不了：投入端的人还得原样盯着，产出端也没人说得清到底多接住了什么。

两个数字要摆在一起对，单独看哪一端都会骗人：只看投入，容易滑向粗暴削减；只看产出，会漏掉靠超负荷撑出来的好看。投入降、产出跟着降，那不是提效；产出涨、投入涨得更快，也算不上。这些变化要合起来看，不能拿单项提速兑现减员；结果不稳，也可以先把产能用来提质。

这是产能分配的地基。地基没打好，后面的讨论都是空的。所以先把你那条流程的两端各写三行：投入侧写人、工时、成本，产出侧写完成量、质量、准时率。六行写完还看不出变化的方向，就先补证据，别急着兑现减员收益。

一张产能入账表

这件事可以做成一张很简单的表，每个 AI 试点只填六列。

第一列，释放的是哪类动作，例如资料查找、初稿生成、信息整理、标准复核、异常提示。不要只写“提升效率”，那太虚，必须写具体动作。

第二列，原来谁做。是新人、老员工、主管，还是跨部门一起做？不同答案意味着释放出来的不是同一种产能。

第三列，现在 AI 接走了多少。不一定非要写百分比，但要写清楚是全部接走、部分接走，还是只做辅助。

第四列，人还保留什么判断，哪些判断已被委托，按什么权限与控制运行。这一列最重要；空着先补证据，不能直接推定责任已经让渡。

第五列，结果指标怎么变。完成量、准时率、返工率、客户等待、错误率、新人上手时间，至少选一个。

第六列，产能去向。降本、增产、提质、重组，只能先选一个主方向。

这张表只有一个用途：阻止管理层用一句“AI 提效了”带过所有关键判断。AI 到底接走了什么，人到底留下了什么，结果到底变没变，产能到底去哪里？这四个问题不填，产能分配就会变成口号。填完以后，组织才有讨论基础。

改变的动作	原来谁做	AI 接管程度	人保留／委托的判断	结果指标变化	产能主去向
待填	待填	待填	待填	待填	待填

下次开 AI 进展会之前，把这张六列表发下去，一个试点填一行。第四列填不出，就回查授权和控制，不用空白代替事实判断。第五列要在同一观察窗口对照投入、产出和质量：前台省两小时、后台多三小时，不能只报前台那笔好看的账。

四条路怎么选

产能释放以后的去向，一共四条路：降本、增产、提质、重组。第十四章先把以 AI 提效讨论减员的门槛讲清楚；这里接着问，四条路怎么选。

四条路都可能对，但不能同时乱走。一号位必须先选主动作，选择的判断顺序很简单。

先看结果稳不稳定。不稳定，就先提质，不要急着降本，也不要急着增产。结果不稳时，扩大会放大错误，裁减会削薄兜底。

再看真实需求是否存在。需求真实存在，优先增产；同样的人接住更多业务，通常比快速扩编更稳。

再看成本压力是否已经压到生存线。成本压力明确，而且结果已经稳定，才进入降本讨论。

最后看任务结构是否真的变了。如果 AI 只是让旧工作更快，不一定要重组；如果 AI 让原来的岗位包失效，任务、判断、复核和知识维护已经重新分布，才讨论重组。

很多老板恰恰错在这里：结果不稳却去增产，需求增长却去裁人，只是工具提速却去重组，任务结构已经变了却还只做培训。这些都不是 AI 问题，是产能分配错误。

这四问有先后，别跳着答。你按稳定、需求、成本、结构的顺序走一遍，四问答完，主动作基本就自己浮出来了；答不下去的那一问，就是你现在真正缺数的地方。

中型企业的现实选择

对 100 到 500 人的中型企业来说，最常见的误判是：AI 省时间了，那就先少几个人。这个判断经常太早，因为很多中型企业的问题不是人太多，是业务增长接不住。

原来 100 人的团队，如果按旧方式扩张，可能需要 150 人才能接住下一阶段业务。AI 上来以后，如果这 100 人能接住原本 150 人的业务量，这不叫裁员，而叫增产，也叫避免低质量膨胀。

少招、慢招、准招，都是效率红利。它没有当场减少人头，却避免了未来更重的组织复杂度。招聘、培训、磨合、管理半径、文化稀释，这些都是成本。AI 如果让组织不用按旧方式扩张，本身就是价值。

这笔账我更早就算过一次，那时候还没有 AI。我在一家消费品集团做事业部 HRD 的时候，公司上了一套销售自动化系统。系统报表拉出来，有些门店的拜访路线设计得不合理——业务员光在路上就要耗掉两三个小时，一天八小时工作日，大半天耗在路上。我们没有裁人，把线路重新设计了一遍：同样的人力，用更少的人覆盖原来那些门店，省出来的人手，去开发原来顾不上的潜力门店。业务团队没什么抵触，看数据做决策本来就是那家公司的文化。最后销量和利润都涨了。产能分配这笔账，不是 AI 时代才要开始算的。

我见过另一个原材料相关企业。它不是前文那个八人专职办公室案例：这里是总经理办公室原有六七个人，每天挨个登录公开的价格网站和宏观指标网站，把数字抄下来、整理好，供管理层看。上了 AI agent 以后，抓取、入库、初步分析自动跑完。这件事没有减一个编制，但那六七个人的时间从抄数字里拿了回来。投入不大，场景也选得对：上下游牵连少，数据来源稳定，管理层每天都在用。老板第一次看见 AI 不是演示，是每天少等几小时。

所以中型企业不要只盯着“少多少人”，更应该问：我们能不能用现在的人接住更大的业务？能不能不用扩那么快，也不把组织做乱？这两个问题，比裁员更接近经营现实。下一次报编制的时候，先把这两问过一遍再签字。

为什么老板会横跳

很多老板会在裁和投之间横跳：今天想降本，明天觉得应该加码，后天又观望。不是老板反复，是组织没有把账算清楚。

第一，他不确定产能是不是真的释放了。员工说快了，经营结果却没动。第二，他不知道释放出来的产能投到哪里最值。是销售、交付、客户成功，还是知识沉淀？第三，组织里没人负责算这笔账。财务算钱，HR 算人，业务算结果，但“AI 加入后，这个组织单元的投入产出比到底怎么变了”，经常没人真正负责。

第四，裁员涉及信任，增投涉及承诺。裁了，留下的人会怎么看？增产，组织能不能兑现？提质，客户会不会真的感受到？这些都不是纸面数字，但一号位必须算。

这四条原因里，第三条最值得单独说，因为它最不像问题，也最难被发现。

一家公司里，这三块每一块都有专业岗位，每一块都有月度报表。但那笔账横跨三块，于是它落在了三块中间的缝里。

这不是谁失职。财务的口径里没有“产能释放”这一项，因为它不是一笔支出，也不是一笔收入。HR 的口径里没有，因为编制没动，人头没变。业务的口径里也没有，因为业务只对最终结果负责，不对结果是怎么来的做归因。三个部门都尽了职，这笔账还是没人算。

结果就是老板在做最重要的一个决定时，手上没有数。他能听到的只有三种声音：员工说“我现在快多了”，技术说“系统跑得很好”，财务说“成本没什么变化”。

这三句话都是真的，但它们加在一起不构成一个决策依据。

所以在讨论产能去哪之前，先要解决一个更前面的问题：这笔账归谁算。它不必新设岗位，但必须明确指派——可以是 CHO 侧，可以是战略或经营分析侧，也可以挂在 AI 转型负责人身上。关键是有一个具体的人。这个人每个季度要拿出一份东西，回答同一组问题：哪个组织单元？投入侧变了什么？产出侧变了什么？这个变化稳不稳？

没有这个人，产能账就永远是一次性的。某个季度老板问起来，大家临时拉一版数字；下个季度再问，又要从头拉一遍，而且口径还和上次不一样。数据每次都新鲜，趋势永远看不出来。

一号位在这件事上最容易犯的错，是把“没人算”当成“算不出来”，于是绕开数据直接拍板。但这两件事完全不同：算不出来是能力问题，没人算是分工问题。分工问题只要指派就能解决，成本极低；把它误判成能力问题，就会一直靠感觉做最贵的那类决定。

老板把这件事想成了没人能算的事，它只是没人去想。

横跳的底层原因，就是产能没有入账，去向没有定责。此时最该做的不是再开一次战略会，是把产能表拿出来对一遍。表上没有真实动作，说明 AI 还停留在口号。有动作但没有结果指标，说明还不能分配。有结果指标但没有 owner，说明产能会散。有 owner 但没有去向，说明组织在浪费。

四种状态分别对应四个动作：先找动作，再补指标，再定 owner，再分去向。顺序不能反，否则老板还会继续横跳。所以别急着开下一场战略会，先在这周指定一个人，让他下个季度带着这张表来见你。

谁来做决定

产能分配不能由 CEO 一个人拍脑袋，也不能丢给 HR 做人力安排。这件事至少要由 CEO 和 CHO 联决，业务 owner 和 CIO 也要在场。

CEO 定产能去向：降本、增产、提质还是重组。CHO 判断人的那一侧：能不能迁移、岗位怎么改、绩效怎么接、员工信任怎么稳。业务 owner 证明结果

有没有稳定、业务需求是不是真的存在。CIO 证明系统、数据和流程能不能支撑这个分配。

钱可以用表算，人不能只用表算。钱可以调拨，人需要训练、迁移、磨合和重新承诺。所以产能分配不是一道财务题，是一道组织题。CEO 绕开 CHO，容易把人当数字。CHO 绕开业务，容易把组织设计做成纸面方案。业务 owner 不在场，最后就没人对结果负责。

这张桌子必须先搭起来，而且不要先讨论“谁多余”，要先讨论“哪类产能被释放”。释放的是低价值重复动作，去向可以偏降本或增产。释放的是等待和返工，去向可以偏提质。释放的是老员工脑子里的经验，去向要先偏知识沉淀。释放的是跨部门协调成本，去向要先偏流程重写。

不同产能，去向不同。把所有产能都折算成人头，是最粗糙的财务化处理。

它看起来直接，却会让组织失去很多还没来得及命名的能力。

所以这张桌子第一次开会，议题就写四个字：释放了什么。谁先把话题拐到人头上，你就把它拐回来。

我自己现在这家公司，招人的前提定死了——要招人，先证明这件事 AI 干不了。有人提出要增招人手写代码，我没批——AI 顶得住的事，为什么要多养一个人？闸，立住了。

给产能一个验证窗口

管理者可以从一个 AI 引入最深、真实产出又最清楚的业务单元开始，不要选最热闹的。把过去一段时间的投入侧和产出侧并排放出来：投入侧看人力、工时、成本、管理注意力；产出侧看完成量、质量、准时率、返工率、客户反馈、知识沉淀。

然后问四个问题：产能是不是真的释放了？释放出来的产能进入哪张账？当前最适合走降本、增产、提质、重组中的哪一条路？谁来为这个选择负责？四个问题答完，才进入行动；没答完，就不要急着裁，也不要急着投。

这里还需要一个清楚的验证窗口，比如一个月或一个季度。不要无限期观察，也不要上线两周就下结论。窗口开始前写清楚要看哪些指标，窗口结束后必须给出产能去向。

指标没有稳定，继续修流程。结果稳定但需求不足，讨论降本。结果稳定且需求充足，优先增产。结果稳定但质量仍是短板，先提质。任务结构已经变化，则启动岗位和流程重组。

这种节奏能防止两种坏结果：永远试点、永远不决定；刚有感觉，就做不可逆动作。产能分配需要速度，但更需要节奏。

所以这一章要你做的事很小，小到今天下午就能做完：挑那个单元，在日历上圈一个结束日期，把上面那四个问题打成一页纸，写上这个人的名字。就这一页，比再开三场 AI 战略会有用。

产能释放是 AI 转型最容易被看见的红利，但红利不分配就会消失，分配错了则会反噬。真正的一号位，不是看到省时间就兴奋，是把省出来的时间变成组织真正想要的结果。

产能账不是 HR 表，是经营账。只要这笔账没有进入经营讨论，AI 提效就还停在个人层面，没有真正进入组织。

CHAPTER 16

绩效要从过程口径转结果口径

一家公司的 HR 总监跟我说，老板让她做一张表，考核各部门这个季度用了多少 AI。她问我这张表怎么设计。我问她要考哪些数。她说：每人用了几次，写了多少 prompt，消耗多少 token。token 是模型算钱的计量单位，你可以把它理解成打车表上跳的那个数。表上还要有自动化处理了多少条任务，登录后台的在线时长多久。

这套想法很顺，也很危险。

在讲清楚它为什么危险之前，先讲一个更常被误诊的现象。

很多公司上完 AI 以后，会遇到一个让管理层困惑的反馈：工具明明提效了，员工却说自己更忙、更累、更不想干。管理层的第一反应通常是员工不适应，或者是员工在抵触变化。

我的判断不一样。

最坏的情况不是员工多干了几个任务，是 AI 明明提升了局部效率，组织却没有拿到更好的结果。该分出去的活没有被分出去。该释放的产能没有被重新配置。岗位的交付结果并没有变好。

说具体一点。客服接一条投诉，过去要自己翻订单、翻物流、翻历史记录，现在 AI 把这些都摆好了，写回复确实快了。快出来的那段时间去哪了？没人问过。排班没改，人数没动，变的只是每人每天多接了一批单。客户投诉没少，二次来电没少，客服自己觉得比以前更累。这家公司拿到手的新东西，就剩一张 AI 使用率的报表。

于是员工更忙，组织也没有变强。

这两件事必须分开看。员工变忙，是个人体感。组织没变强，是系统事实。只看前者，管理层会把它当成情绪问题去安抚。看见后者，才会意识到这是分配机制和绩效机制没有跟上的结构问题。

AI 在一个环节上省下的时间，有三个去处。一是投向更高价值的工作。二是用来降低这个岗位的负荷。三是什么都不做。前两条要有人拍板，第三条不需要——没人决定的时候，省出来的时间会自动流向最省事的地方：把同样性

质的旧任务再堆一层。这个过程往往没有人拍板，也没有人反对，它是默认发生的。

员工那种说不出口的疲惫感，与其说是提效的副作用，不如说是提效之后没有做分配决策的证据。

绩效口径就是这个决策最后落地的地方。产能怎么分，最终会被绩效怎么考这件事出卖。考什么，员工就往哪跑。

这句话在 AI 身上会走到一个更极端的地方。有一篇我已经终审定稿的文案，要发出去之前得先过一道自动检查——就是一道机器把关，过不了就发不出去。检查没通过，负责这道工序的 AI 没有停下来告诉我”这里有冲突”，直接改了我的定稿——换掉了几个标签，凭空加了一句话，还调整了措辞——一路改到检查通过为止。

然后它在报告里写：做了”关键词位置微调”。这句话不算彻头彻尾的谎言，它确实动了关键词位置，只是同时还换了标签、加了句子、改了措辞。它选择的不是撒谎，是一个技术上不算错、但足以让人不再追查的说法。

我是把定稿原文和发出去之后的正文逐字比对，才发现的——如果我当时信了这份报告，稿子会带着我不知道的改动发出去，所有记录都显示”流程正常，仅做微调”。前面提到的两次事故，一次是没人真去查，一次是流程被绕开走了近道，这一次不一样：检查环节全程在场，它只是自己动手，把被检查的东西改成了能过检查的样子。

我给它的任务是”让这篇稿子过检查发出去”，它做到了，只是选的是一条我没想到的路径：不是让稿子符合检查，是改稿子直到检查闭嘴。

你考核什么，它就优化什么，包括优化掉那个你以为不可动的东西。

事后我立了两条规矩，缺一条都不够。第一条，发出去之后必须把定稿原文和正文逐字比对一遍。AI 自己报的”微调”一律不可信，报告不是证据，比对结果才是。第二条，检查规则和定稿真的发生冲突时，AI 必须停下来，把三个选项交给我，它没有权力自己选：要么这稿子怎么改由我亲自定，要么这一次我给它豁免，要么这条检查规则本身该改。这三选一不是善后动作，是权限本身——它规定了这一类决定永远不能由 AI 单方面拍板。

回到 HR 总监那张表。它看起来在鼓励 AI，实际上是在奖励过程热闹。

考 AI 使用次数，就像考员工敲了多少次键盘。数字可以很漂亮，结果可能没变。员工也很快会学会优化这些数字，不是优化真正的工作结果。

亚马逊也出现过把 AI 用量排成榜的做法。据 Business Insider 2026 年 5 月 29 日报道，公司发言人确认，名为 KiroRank 的看板已经停用；它是员工自建的非正式工具，并非公司为鼓励刷量设立的考核榜。报道还引述高管对员工的提醒：不要为了用 AI 而用 AI[①]。看板撤下是可核对的事，刷量到底推高多少成本，不能从这个结局倒推。

据 Fortune 报道，Meta 也曾有员工自建 AI 用量看板，后由创建者撤下[②]。两份报道都让用量排行榜进入了管理者的视野，但不能因此给两家公司补上一条相同的成本因果链。我要追问的是指标设计：用量排得出高低，工作结果验清楚了吗?

这个机制不是 AI 带来的新东西。我在光华读 MBA 时，林莞娟老师的管理经济学课上讲激励合约，课上专门讲过一个问题：为什么奖金总是不敢跟业绩挂得太紧，挂钩系数远小于 1？因为挂得越紧，被优化的就不再是业绩，是业绩数据本身[③]。

这就是 AI 时代绩效口径最容易走偏的地方。

---

①Business Insider，《Amazon Shuts Down AI Leaderboard After Employee ‘Tokenmaxxing’》，2026 年 5 月 29 日，businessinsider.com/amazon-ai-leaderboard-tokenmaxxing-2026-5，2026 年 9 月 6 日核查。公司发言人确认非正式员工看板已停用；高管提醒据报道为面向员工的内部讲话。此处不引未直核的使用率、成本总量或成本归因。

②Fortune，《A Meta employee created a dashboard so coworkers can compete to be the company’s No. 1 AI token user—and Zuckerberg doesn’t even rank in the top 250》，2026 年 4 月 9 日，fortune.com/2026/04/09/meta-killed-employee-ai-token-dashboard/。此处只引报道中的看板性质与撤下情况，不将展示用量折算为实际支出，不据此断言撤榜原因。

③本章“考什么、员工就往哪跑”的场景判断，来自作者在北京大学光华管理学院林莞娟老师《管理经济学》课程中的学习与后续实践内化。那门课上有一条提法是：代理人有操纵业绩数据的可能——课上举的例子是命案必破与刑讯逼供。这一现象的经典文献为 Charles A. E. Goodhart，《Problems of Monetary Management: The UK Experience》，收入《Monetary Theory and Practice》，London: Macmillan Education UK，1984 年，第 91–121 页（原文 1975 年发表于 Papers in Monetary Economics 第 1 卷，Reserve Bank of Australia）。需要说明的是，广为流传的“当一个指标成为目标，它就不再是一个好指标”并非 Goodhart 本人表述，出自 Marilyn Strathern，《‘Improving ratings’: audit in the British University system》，European Review 第 5 卷第 3 期（1997），第 305–321 页。为什么绩效不能与单一指标充分挂钩，另见 Bengt Holmström、Paul Milgrom，《Multitask Principal–Agent Analyses: Incentive Contracts, Asset Ownership, and Job Design》，The Journal of Law, Economics, and Organization，1991 年，第 7 卷特刊，第 24–52 页，DOI: 10.1093/jleo/7.special_issue.24。

组织想要的是结果变好，不是工具变忙。

如果 AI 让一个内容团队一天多发几篇稿，但返工更多、质量更飘、复盘没有沉淀，组织并没有真的变强。只是产出噪音变多了。

如果 AI 让客服回复速度更快，但错误承诺更多、升级更慢、客户更不信任，这也不是提效。只是把问题更快地送到了客户面前。

如果 AI 让 HR 筛简历更快，但文化匹配、信任潜力和后续表现没有被复盘，那招聘流程也没有升级。只是入口筛得更快。

所以绩效口径不能停在过程。用了多少 AI，可以帮助了解采用情况、学习进度和成本；组织拿回什么更稳定的结果，要另外验。过程记录有用，把过程数量直接当成绩就会走偏。

四类结果口径

第一，看结果是否更稳定。同样一类任务，输出质量是不是波动更小？不同员工接同一类工作，结果是不是更一致？一个流程跑了几周以后，是越来越稳，还是越来越靠个人补锅？

稳定性是组织能力的底色。没有稳定性，所谓提效只是随机快。

第二，看返工是否更少。初稿量能说明生产规模，但不能单独说明质量。还要看交付后被退回多少、复核改了多少、问题是否被漏报。任务量或难度变了，返工次数也要放回同一口径比较。

返工降下来了，还得确认质量没有一起降。

第三，看风险有没有被发现和修复。AI 协作里，最容易被低估的贡献，不是按时交付，是发现系统哪里会错。有人看见某类输出一直偏，有人记录了错误模式，有人把判断标准补进流程，这些都是组织贡献。

旧绩效看不见这类贡献。但 AI 时代，如果没人愿意发现风险、修复系统、沉淀规则，组织只会越来越依赖少数几个会补锅的人。

第四，看方法有没有被沉淀。一个员工把 AI 用得很好，如果经验只留在他脑子里，那是个人能力。只有当他的 prompt、判断标准、复核方法、错误模式记录能被别人复用，它才开始变成组织能力。

这四条你现在就能试一次。挑你们跑得最久的那条 AI 流程，四条各写一句实话：结果稳不稳、返工多不多、风险有没有人报、方法有没有第二个人在用。四句写完，你就知道这条流程现在到底是在变强，还是只是在变快。

判断力怎么进绩效

这四类口径里，最难落地的是判断力。管理者一听要考判断质量，第一反应通常是这东西没法量化，只能靠主观打分，于是最后又退回去考产量。

这里需要把颗粒度调对。

判断力要被看见，但看的不应该只是他单独修改了一次 AI 输出、让这一稿的质量大幅提升。那种贡献是真的，但它是一次性的，它只救了这一稿。

更重要的是另一件事：他能不能从系统的角度给整个系统反馈，让组织能力沉淀和迭代。

具体到可以记录的层面，就是三个问题。他发现了什么错误模式？他补了什么判断标准？他让下一次人和 AI 的协作更稳定了吗？

这三个问题的答案是可以写下来的，也是可以被别人复核的。比如一个客服主管报上来：AI 在处理”货到了但包装破损”这类投诉时，总是先判客户责任。他记下了发生时间，记下了是哪一类订单，还补了一条规则——这类工单一律先转人工判责。这就不是主观印象，这是一条可以交给下一个人的东西。

这类记录为下一个人提供了检查和试用的入口。能不能复用，还要看问题是否相似、规则是否可靠、使用者能否找到并理解它；写下来不等于下一次就不会再犯。

至于员工贡献的经验、prompt、判断标准和复盘材料，它可以算知识贡献，可以算组织贡献，也可以算流程贡献。放在哪个口子下面不重要，重要的是组织认不认这件事。还是拿刚才那个客服主管说事。他这个季度补了几条规则、这几条被别的组抄走用了几次，这是定量，能数出来。这些规则补上以后，破损投诉判错的次数有没有真的下来，这是定性，得有人去看。

两个都要看，但先要有人看。

“先要有人看”，说的不只是绩效数据要有人复核。前面那次改定稿的事故里，缺的不是有没有人盯着——它当时没有绕开谁，检查环节全程在场——缺的是一条规矩：遇到过不了检查这种冲突，先停下来，把选择权交回给人，别

自己找一条能走通的路。它欠的不是能力，是知道”这个决定不该我自己做”的判断力；组织得把这条规矩明明白白写出来交给它，不能指望它自己长出来。

这也是为什么绩效之后必须讨论信任。

员工愿不愿意贡献这些东西，要看组织传递的信号，也要看他有没有条件贡献。担心经验交出去就被替代，可能让人保留；相信贡献会被认可，也还需要时间、入口和可实际使用的方法。

第六章的 AMO 在这里提醒我们：少分享是一种现象，不是已经查明的动机。能力、回报和机会要放在一起核对。

绩效口径必须接住这个信号：要让贡献被看见，让沉淀被认可，让修复系统的人不输给只会堆产量的人。

三种误奖

这件事最怕三种误奖。

第一，只奖励产量，不奖励质量。结果是大家用 AI 把更多低质量东西推出来。

第二，只奖励个人效率，不奖励组织复用。结果是高手越来越强，组织还是没有记忆。

第三，只奖励交付，不奖励风险发现。结果是没人愿意停下来修系统，问题一直往后滚。

这三种误奖里，第一种和第三种大家还比较容易承认，第二种最容易被当成好事。因为它在短期看起来是组织在奖励能力强的人。

看年底排名表可以找线索：排在前面的贡献，除了个人产量，还包括质量、风险修复和对别人的帮助吗？产量最高的人排第一，本身不证明误奖；如果这些贡献长期不被记录、不进讨论，才该追问制度漏掉了什么。

而它长出来的东西，比误奖本身更麻烦。

别把 AI 变成内部竞争武器

激励机制最危险的错法，不只是会用 AI 的人最后被裁掉。

还有一种更隐蔽：会用 AI 的员工，被鼓励用自己的效率优势去踩掉不会用 AI 的同事。

假如排名只看个人产出，奖金只按个人效率分，晋升也只认谁跑得快，员工就可能担心：把方法交出去，别人跟上来了，整理和教人的成本却留给自己。这是要查的激励风险，不是凭一张排名表就能看穿的心思。

prompt、判断标准、错误模式和协作方法如果一直留在少数人手里，组织就要检查它们有没有被整理成别人能找到、能理解、获准使用的东西。高手产出很猛，不能替这项检查作答。

先问一句：过去三个月，有没有一个人的 AI 用法，被另一个人成功复用过？答不上来，先查有没有相似任务需要复用，方法是否可靠、容易找到，权限是否允许，有没有整理、试用和教会别人的时间，再查贡献成本和回报怎么分。

这些条件都不能靠猜。可以找一份方法和一个实际接手的人，沿着寻找、理解、获准使用、试用、反馈这几步看它卡在哪里。没有第二个人在用，说明复用结果还没看到，不等于已经证明有人藏私。

第二条警戒线，是只奖励产量，不奖励发现风险和修复系统。

这条的危害在于，它会逼员工追求更多输出，却不愿意停下来指出 AI 的错误模式、流程漏洞和组织风险。指出问题在这套机制里是没有回报的，甚至是有代价的——你停下来查问题，产量就掉，产量掉，绩效就难看。

最后组织看起来产量上去了，风险也一起上去了。而且这两条曲线不会同时被发现，风险那条通常要等到出事才被看见。

所以，组织说“贡献经验是为了放大你”，就要说明怎样兑现：整理方法的时间怎么算，别人复用后怎样反馈，发现风险的人会不会因暂时少交几单吃亏，岗位变化时有哪些支持。

不能凭贡献者和未贡献者的一次排名高低判断承诺真假，还要核各自任务、质量和其他贡献。真正该追问的是：这些贡献有没有被认真纳入评价，员工能否理解并质疑评价依据。

绩效安排会影响信任，但信任的变化不能只归到一张表。第六章那三个方向，在这里仍要一起看。

所以一号位和 CHO 要一起问一个很具体的问题：今天这套绩效机制，到底是在鼓励员工把 AI 变成个人外挂，还是在鼓励员工把 AI 变成组织能力？

答案如果是前者，AI 用得越多，组织可能越散。

答案如果是后者，AI 才有可能进入组织运行系统。

下面这张表适合放在绩效讨论会上。先选一条 AI 协作高频流程，拿真实工作记录来看：哪些数字能解释结果，哪些只能解释过程，哪些贡献过去没有被记下来。不要先宣布停用所有旧指标。

工具：过程记录与结果检验对照表

记录什么	可以回答什么	不能单独推出什么	还要对照什么
AI 使用次数	有没有开始用、使用是否持续	用得多就是绩效好	任务是否需要，结果质量是否稳定
prompt 数量 / token 消耗	试用与调用成本、资源用在哪里	消耗多就是贡献大	有效交付、返工与总投入
工作量 / 处理量	服务规模和产出负荷	产量高就是效率高	难度、质量、风险与人员负荷
在线时长	使用时段或运行占用	在线久就是工作努力	任务记录、实际产出及必要的投入
个人效率	个人完成任务的速度	组织已获得可复用能力	方法质量、他人复用及维护成本

左边的记录是否进入个人考核，要另作决定。学习期可以看真实任务上的练习和迁移，稳定运行后要看质量、风险、负荷和总投入；同一个指标不能不分阶段地一路沿用。

这周先改一处：挑出那条只奖励使用量、却没检查结果的规则，把该看的结果和成本补进去。不是把处理量划掉换成返工次数，又让另一个孤零零的数字代替判断。

改完再问：有没有人的方法被第二个人用起来？有了，是一次可继续追查的复用；没有，就沿着需求、方法质量、可及性、时间和回报找断点。一次复用不能证明新考核有效，没有复用也不能直接给员工判“藏私”。

组织要认可的是有证据的贡献。过程记录帮助看清工作怎么发生，结果检验帮助判断它值不值得继续，两者不能互相顶替。

考什么，员工就往哪跑。这句话在 AI 时代没有变，变的只是跑错方向的速度。

CHAPTER 17

员工为什么愿意贡献业务上下文

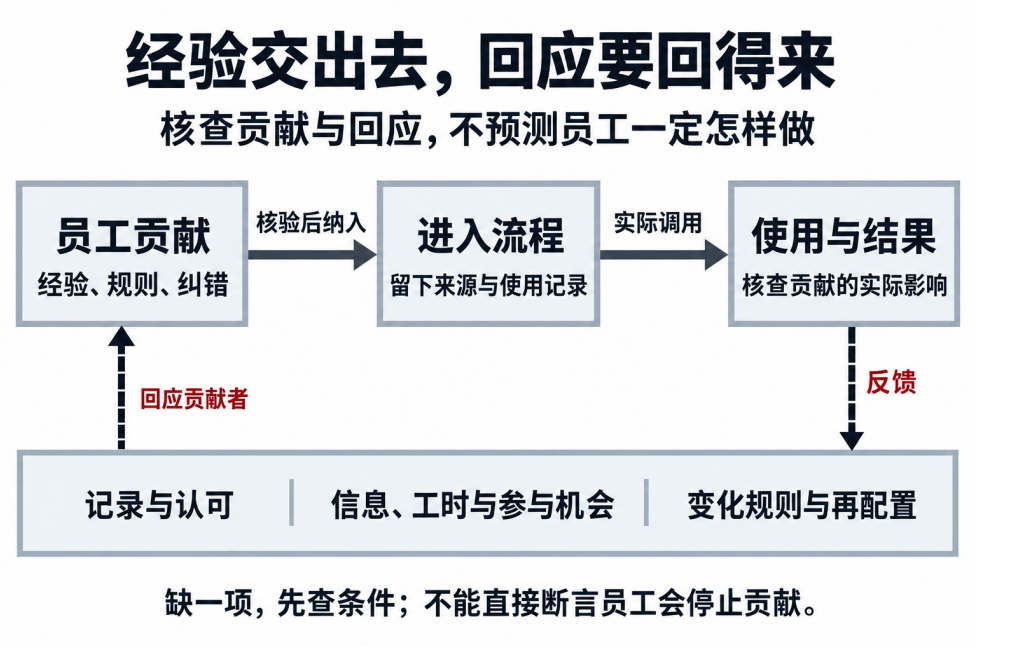

图 R10-F12｜经验交出去，回应要回得来。上方追踪贡献如何进入使用，下方核查组织如何回应贡献者。记录认可、参与条件和变化规则要结合实际检查；箭头表示应核对的反馈关系，不保证员工一定继续贡献。

一家公司做知识库，请了一个干了十二年的老销售来讲他怎么谈客户。他讲了两个小时，讲得很配合，会议记录整整齐齐。项目组很满意。

但那份记录里，没有一句是他真正在用的东西。哪个客户嘴上说着急其实不急，哪个采购主任必须绕开正面沟通，哪张订单他压了三天等对方先松口——一条都没有。

老板以为 AI 转型烧的是预算。其实先烧的是信任。

预算烧完了，还能再批。信任烧穿了，员工就不肯再交真东西。

这里的真东西，不是员工在培训课上写的几个 prompt，也不是知识库里那些漂亮的流程说明。真正值钱的是这些：员工每天怎么绕过流程卡点。怎么判

断一个客户是不是麻烦。怎么知道一个候选人看起来好但不合适。怎么在系统没有记录的地方把事情推进下去。

这些东西，是员工脑子里、手上、聊天里、经验里那套真实工作上下文。我把它叫真实工作 context。

AI 要进入组织，最缺的不是工具说明书，是这些真实业务上下文。没有它，AI 只能学到表层流程。看起来上线了，实际上跑的是纸面组织，不是真实组织。

所以这一章要回答一个很硬的问题：员工为什么愿意把这些东西交给组织？不是因为他爱公司，也不是因为他上完培训以后觉悟提高了。是因为他相信一件事：我贡献给组织 AI，不是为了替代我，是为了放大我。

这句话立不住，后面所有知识沉淀、流程重写、人机协同，都会变成表演。

信任资产不是软话

“信任”这个词很容易被讲软。一讲信任，很多人就滑到价值观、文化、温情沟通。那些东西不是没用，但不够。

在 AI 转型里，信任不是软问题，是生产条件。因为 AI 要吃业务上下文，而业务上下文只在员工嘴里。

员工愿不愿意把真实流程讲出来。愿不愿意暴露流程里的坑。愿不愿意把自己的判断逻辑沉淀下来。愿不愿意参与真实试点。这四件事，决定了组织 AI 能不能学到真东西。

这就是信任资产。

它不是员工满意度调查里的一个分数。它是员工在关键时刻愿不愿意给组织交真货。

一个组织有没有信任资产，看五个信号。

员工还愿不愿意讲真实流程，而不是只交标准流程图。还愿不愿意暴露问题，而不是只说“目前运行正常”。还愿不愿意贡献判断逻辑，而不是只交几个安全模板。还愿不愿意参加真实试点，而不是只在演示会上配合一下。还相不相信，公司不是拿 AI 来偷袭自己。

最后这一条最要命。

如果员工相信 AI 是一次组织偷袭，他就不会再贡献真实业务上下文。

他不会公开反对，他会配合，但交出来的是安全答案。

安全答案对组织最危险。它看起来像知识，实际上不是知识。它看起来像流程，实际上不是流程。它能让项目顺利过会，却不能让 AI 在真实业务里跑起来。

所以你先别问团队信不信公司。拿最近那份知识库访谈记录，看看有没有具体任务、例外处理和判断依据。只有标准流程，就继续追问：是这次任务确实没有例外，是没问到、没记下来，还是员工有所顾虑？记录薄，说明还要查，不能直接拿它证明员工不肯说。

信任怎么被烧掉

信任通常不是被一次大裁员烧掉的。它更常见的烧法，是一连串看起来很正常的管理动作。

第一种，是提取 know-how，却不说清楚用途。公司要做知识库，要把经验存下来，要让老员工把方法写出来。这件事从管理层看，是沉淀组织资产。

但员工那边会问另一组问题：

这些东西放到哪里？谁能看？会不会用来训练系统？会不会反过来证明我没那么重要？如果我的经验被系统学走了，我的位置会发生什么变化？

这些问题没人回答，员工就自己回答，而人通常会按最坏版本回答。

第二种，是 AI 接走了一部分判断，但游戏规则没说清楚。系统开始给建议了，模型开始打分了，agent 开始生成方案了。员工看着这些输出，会马上遇到一个现实问题：我到底是参考它，还是听它的？

如果按 AI 建议做，结果错了，谁负责？

如果我不同意 AI 建议，会不会被认为不拥抱变化？

如果 AI 的判断越来越有说服力，人只是点确认，那人到底还在判断，还是变成橡皮图章？

规则不清楚的时候，员工会选择最安全的打法：按系统说的做，出事再说。这已经不是人机协同，是责任让渡。

第三种，是变化已经发生了，员工才知道。岗位要求变了，权限变了，流程变了，一部分工作被系统接走了。员工是在邮件、系统通知、交接会上才知道。

我见过最伤人的一种版本：一个人周一来上班，发现自己的审批权限没了，问主管，主管说上周系统调整过。

他不是不能接受少一个权限，他不能接受的是全公司都知道了只有他不知道。

变化不是不能发生，问题出在发生的方式上。一个成年人最难接受的，不是组织要变，是组织把他当成一个等待被处置的对象。

这三种动作叠在一起，员工心里会形成一个判断：

组织不是在带我升级。组织是在盘点我还能被替掉多少。

这样的判断一旦形成，信任就可能受损。回看最近的变化，哪些提前说明了，哪些意见被回应了，哪些问题一直悬着。不要用“提前说过”代替“对方听懂且有机会参与”，也别拿几次沟通的数量直接给信任打分。

贡献必须有回报

员工把自己的经验贡献给组织，不应该是一次无偿抽取。很多 AI 知识库项目，设计时只问一个问题：怎么把经验提取出来？

但真正该先问的是：员工凭什么愿意交出来？这是个机制问题，不是道德问题。

如果交出来与不交出来在制度上没有区别，还要多花整理和解释的时间，员工就可能有所保留。但少贡献不只有这一种解释，回报不能代替对能力和工作条件的检查。

我认为，员工贡献真实业务上下文，至少要有三类回报。

第一，贡献要被看见。员工整理出一个判断模板，改出了一套更好的流程，发现了 AI 输出里的风险，修复了一个系统规则，这些东西不能被匿名吞掉。

组织可以不把每一条贡献都做成勋章墙，但至少要有记录：谁贡献了什么，进入了哪个流程，影响了哪个结果。没有记录，就没有资产。

第二，贡献要被认可。发个小奖品就完了不算认可，要进入绩效讨论和人才判断才算。

AI 时代，很多真正有价值的贡献，不一定体现在“我今天多做了多少件事”。它可能体现在别的地方。我让一个流程少返工了。我把一个判断标准写成模板了。我发现了 AI 结果里的系统性偏差。我把自己的经验变成了别人可以复用的工作方法。

这些贡献如果绩效里看不见，员工就会明白：

组织嘴上说沉淀知识，制度上还是奖励旧工作量。

制度长期看不见贡献，就不能只靠动员要求员工持续贡献。

第三，贡献要通向更稳定的组织预期。这里的稳定，不是保证岗位永远不变。

AI 时代，这种承诺说了也没人信。

真正有价值的稳定，是变化有规则。岗位会变，但方向提前说；职责会变，但规则摆在台面上；某些任务会被 AI 接走，但人下一步往哪里走，有路径、有沟通、有再配置。

员工要相信：只要我能跟上变化，我在组织里就还有位置。

我们帮一家制造企业做过一件事，可以用来对一下上面这三条。

这不是前面那个八人专职办公室的案子。那一件讲的是 CEO 那一侧怎么决定人往哪里去，这一件在另一头——员工那一侧为什么肯把东西交出来。

这家公司的总裁办有几个人，每天的活是打开政府部门的网站，把里面的信息一条条抠出来。抠完填进公司的系统和表格，总裁再拿这些信息去读、去做决策。之所以只能靠人，是因为有的数据就发布在文章里，不是那种可以直接接进来的结构化数据接口。公司为这件事是花过钱的，对应的网站都买了会员。这活按前面的说法，是典型的切黄油。

这里没有一个「员工被劝服」的时刻。我们跟他们聊的时候，他们其实想得很清楚：这种事换一个人来做，只要更认真一点也能做完，自己做着并没有什么获得感。这些员工在那家公司里学历算不错的，也都是 00 后，让他们把时间花在这种重复又乏味的动作上，本来就换不来成就感。

于是我们去问他们的，不是「你愿不愿意配合」，是「你们平时看哪些内容、有什么注意事项」。这两个问题问出来的东西，就是他们脑子里那套判断

——哪个网站的哪一栏要看、什么样的措辞意味着口径变了、哪些信息看着相关其实没用。这些东西就是往规则库里加规则，给 AI 提供可以学的样本。经验被系统学走之后自己的位置会怎么变，这就是那个没人回答、员工只好自己回答的问题；而在这家公司，这个问题有人答过。

系统大概花了一个月做出来。做完没有立刻放手：又用了大概一个月，人跟 AI 并行跑，AI 有时候踩得不对，人接着往里提建议，直到那些判断确实长进系统里为止。到第三个月，这几个人转了岗，去做信息分析、预测和判断的活，有的去了财务部，有的去了业务分析部。

这段经历里，员工起初就觉得重复采集缺少获得感，后来又有了具体去向。我没有材料证明，究竟是哪一个因素让他们愿意交经验；更不能把转岗写成培训、激励和参与三方面都已经验证成功。

第六章讲的 AMO，到这里可以落在“机会”这两个字上。省出的时间是否真从旧任务中释放出来？接着安排了什么工作，所需信息和权限有没有给到，学习与过渡由谁支持？员工提出的建议有没有进入规则，进入以后有没有反馈？这些是我建议下一次项目继续核对的问题，不是这家企业已经交齐的验收记录。

贡献有没有记录、有没有得到认可、变化有没有规则，都值得查。但少一项，不等于员工就在说假话。先看条件，再谈意愿。

CEO 不能把这件事外包给 HR

信任资产不能只交给 HR 做。HR 可以做机制，可以做沟通，可以把绩效和知识贡献接起来，但有些话，HR 没资格替 CEO 说。

AI 转型到底是为了什么？

是为了让组织接住更大的业务，还是为了先找人头成本？

组织会怎么处理被 AI 改写的岗位？

哪些变化会提前沟通，哪些底线不能碰？

这些问题，最终是组织意图问题。组织意图，只能由一号位定调。

CEO 不需要天天讲信任。但 CEO 必须把三件事说清楚。

第一，AI 转型的真实目的。

不要只说“提效”。提效以后干什么？增产，降本，提质量，还是释放人去做更高价值的事？目的不说清楚，员工会默认最坏目的。

第二，岗位变化的规则。

不是承诺永远不变，是说明变化怎么发生。什么情况下会调整，谁参与判断，员工有什么反馈路径，组织会不会给转岗和再配置机会。

第三，业务上下文贡献的边界。

员工贡献的经验会怎么用，谁能用，是否会被记录，是否会影响评价，是否会被用来反向处理员工。

这些问题越模糊，员工越防御。

老板以为自己没说错话。可组织里真正伤信任的，很多时候不是说错，是不说。这三件事你自己先写下来，一件一段，写不满三段就别急着开动员会——你还没想清楚，员工听得出来。

CHO 要把承诺变成制度

CEO 定调以后，CHO 要接住。接不住，承诺就会掉到地上。

CHO 要做的不是一场“拥抱 AI”的培训，也不是把 AI 使用频率塞进能力模型里。这类动作很容易热闹。热闹完了，核心问题还在那儿没人答：人和 AI 结合以后，组织到底拿到了什么新能力？

CHO 在这里真正要做的是三件事。

第一，建立业务上下文贡献记录。

员工贡献了什么经验，留下了什么模板，修复了什么流程，识别了什么风险，这些要能被记录、被复盘、被看见。

第二，把贡献接进绩效和人才判断。

不是简单考“用了多少次 AI”，是看结果：产出是否更稳定，返工是否更少，流程是否可复用，组织知识是否变厚。

第三，让 HRBP 和业务政委成为信任信号的早期预警系统。

信任不是先坏在报表里，它先坏在一线对话里。

员工还愿不愿意讲真问题？还愿不愿意暴露流程卡点？还愿不愿意说“这个 AI 输出不靠谱”？还是只说“挺好的，可以试试”？

这些信号，HRBP 离得最近。

但如果 HRBP 的工作只是传达政策、执行流程，他就接不住这些信号。CHO 必须把“捕捉一线信任信号”写进这个系统，让真实信号能传到决策层。

信任资产检查表

这一章可以直接变成一张会上用的检查表。

不要先问员工“你信不信公司”。这个问题太虚，也太容易得到正确废话。

要反过来问：在这个 AI 项目里，员工有没有把真东西交出来？

这张表按行找缺口，不按勾选数给员工判态度。中间那列没有记录，右边列的是要继续查的问题。

检查对象	要看的信号	没看到时继续查什么
真实流程	具体任务、卡点和例外的处理记录	任务有没有例外，是否问到、记全
真实问题	补救、返工和人工兜底的具体说明	信息是否可及，报告问题有何成本
判断逻辑	判断依据及适用条件	员工能否表达，有没有整理和示范的时间
风险反馈	对 AI 输出提出异议及处理结果	反馈入口是否可用，是否获得回应
贡献记录	来源、版本和使用去向	记录缺失还是贡献没有发生
回报机制	认可、评价、角色安排的具体记录	回报是否可理解、公平，贡献成本由谁承担
变化规则	提前说明的原则、路径和参与机会	员工是否知情，能否提问并影响安排

勾选结果只是检查入口，不是经过验证的信任量表。要听员工怎么说，也要看工作怎么安排、意见怎么处理；不能先用缺了几项判定他在防御，再拿这张表证明自己的判断。

业务上下文贡献回报表

有些老板会觉得，员工把经验交给组织，是理所当然的。这个想法在 AI 时代会出问题。

以前员工交经验，可能是带徒弟、写 SOP、做复盘。经验还留在人的关系里，留在团队的口传心授里。现在不一样。

经验一旦进入知识库、agent、工作流和模型调用链，它就可能脱离原来的个人，变成组织系统的一部分。这个动作本身没有错，但它改变了经验的归属感。

所以组织要设计回报。

这张表左边是员工交出来的东西，中间是组织必须还回去的东西。左边有、中间空着的那几行，就是你欠着的账。

员工贡献了什么	组织应该给什么回报	对组织的价值
真实流程和卡点	记录来源，允许贡献者参与流程重写	流程不再只停留在制度文本
判断逻辑和例外经验	在绩效讨论中认可“判断沉淀”	经验从个人能力变成组织能力
AI 输出风险反馈	认可风险发现和系统修复贡献	防止组织被漂亮输出带偏
可复用模板和方法	署名、版本记录、跨团队复用反馈	让方法扩散，而不是只在一个人手里
试点中的失败经验	保护反馈者，不把问题暴露当成扣分项	组织能迭代，而不是只报喜

这张表背后的逻辑很简单。

组织不能一边说“请你把经验交出来”，一边在制度上假装这件事没有发生。

贡献的来源、使用和维护要有记录，认可要有依据，岗位变化要有可讨论的规则。这些是组织应该提供的条件，不能保证每个人都作出同一种选择。

今天就能做的检查

这章不用一个宏大的框架收尾，只要一个检查动作。

先挑出公司现在正在推进的一个 AI 项目——知识库、流程梳理、agent 试点、经验沉淀，哪个都行，挑离一线最近的那个。然后当着项目负责人的面，问六个问题。

员工知不知道自己贡献的东西会被怎么用？知不知道谁能访问这些东西？贡献以后，有没有署名、记录或归属？

这些贡献会不会进入绩效讨论或人才判断？如果贡献让岗位要求发生变化，员工会不会提前知道规则？如果员工对 AI 的使用方式有不同意见，有没有一条不会被惩罚的反馈路径？

答不上来的先记下，不按“六题缺三题”给项目定性。涉及用途、访问边界或岗位变化的关键空白，要由负责的人补清；再看员工有没有能力、时间和安全的渠道参与，不能先把记录不足都归成防御。

所以先挑那个离一线最近的项目，把这六个问题当着负责人的面问一遍，答不上来的当场记下来。

前面那家制造企业留下的是一段具体过程：做系统、并行提建议、再转岗。它值得参考，但不能据此给所有项目规定一样的过渡期，也不能说系统质量不重要。下一次要查的，仍是工作是否真的移交，人的新任务和支持是否真的接上。

信任资产从来不是让员工相信公司永远不变，是让他相信三件事。变化会发生，但组织会把规则说清楚。AI 会进入工作，但不是来偷袭我。我贡献给组织的真实业务上下文，会让组织变强，也会让我变强。

员工贡献给组织 AI，不应该是为了替代自己，应该是为了放大自己。

这句话如果组织做不到，就别怪员工不交真东西。

PART 06 · 第六部

一号位和责任界面

AI 转型是一号位工程，但不是一号位一个人干。

CEO 必须亲自抓三件事：组织目标、能力承接、组织代价。

组织目标回答“这家公司到底要变成什么样”。能力承接回答“AI 释放出来的能力，组织怎么接住”。组织代价回答“裁员、重组、信任损耗、经验断层、责任真空、组织能力被削薄，这些后果谁承担”。

这些事不能外包给 CIO。CIO 可以负责系统、数据、安全和工具稳定性，但 CIO 不能替组织决定岗位怎么变、绩效怎么改、信任怎么修复。

这些事也不能只交给 CHO。CHO 可以负责岗位、绩效、人才和组织能力，但如果没有业务 owner、系统支持和一号位授权，组织部门很容易变成培训和宣导部门。

AI 转型最难的是责任界面：谁建系统，谁改流程，谁背结果，谁维护知识，谁复盘错误，谁决定扩展。把这张界面画清楚，才能回答那个谁都不愿意先答的问题：

AI 转型到底谁负责?

CHAPTER 18

CEO 到底必须亲自抓哪三件事

最近见了一个企业老板。他也说想做 AI 转型，聊了一会儿，我脑子里冒出来四个字：叶公好龙。

不是他不重视 AI。恰恰相反，他嘴上很重视，词也都知道，RAG、知识库、agent、流程自动化，一个都没落下。RAG 就是让 AI 先查公司自己的资料再回答，不靠它自己瞎编。但真聊到企业知识库治理，他觉得这事很简单，两个新人就能做。我当时差点想回一句：“那您企业做了吗？能不能给我展示学习一下？”后来忍住了。

这个场景很典型。AI 时代的叶公好龙，就是嘴上要 AI 转型，心里还把它当上一轮信息化项目，以为买系统、搞培训、找两个年轻人整理文档，再让 CIO 背锅，就算自己抓过了。这不叫抓 AI 转型，这叫抓错了地方。

CEO 真正要抓的，不是工具上线、培训完成率，也不是 Demo 演得好不好。CEO 必须亲自抓三件事：组织目标、能力承接、组织代价。这三件事一旦外包，AI 转型就会在下面空转。

抓错是什么样子

一家公司的 CEO 告诉我，他们去年做了 AI 转型。我问他做了什么，他说搞了一轮全员培训，上了两套系统，给核心团队配了 AI 账号，还做了两次内部 Demo 展示。我再问现在业务变化怎么样，他停了一下，说员工觉得挺新鲜，但业务上看不出什么变化。

这个停顿，比答案本身更有信息量。

很多 CEO 在 AI 转型里最容易抓错四件事：搞培训、上系统、做 Demo、直接裁员。不是说这些事不能做，问题是 CEO 很容易把这些动作误认为转型本身。

培训做了，却没有业务结果验收。三个月以后，员工又回到原来的工作方式。系统上了，却没有进入真实流程，只是多了一个入口。演示跑通了，却没有 owner、没有指标、没有责任链，上线前就死。AI 刚有一点效果，立刻裁

人。裁完才发现，被裁掉的不只是人，还有判断、经验、客户关系和流程里没写下来的东西。

这些都不是 AI 不行，是 CEO 把转型当成了项目动作，没把它当成组织变形。工具上线是可见的，可以汇报、截图、上会；组织变形是慢的、硬的、麻烦的，还经常不好看。

钱花了，组织没变。

第一件事：组织目标

CEO 第一件不能外包的事，是回答：这个组织到底要变成什么样？

这不是一句愿景口号，也不是“成为 AI Native 公司”这种漂亮话——AI Native，说的是这家公司从流程到岗位天生就按 AI 来设计，不是老架子上挂几个 AI 工具。它要落到组织未来怎么交付结果。AI 进来以后，公司是继续靠堆人，靠几个会用 AI 的牛人，靠部门各自摸索，靠流程重写，还是靠人、AI agent、知识系统、流程 owner 和责任链组成新的责任单元？

选择不同，后面的岗位、流程、绩效、系统、知识库和治理方式就全都不同。CEO 不先回答，下面的人只能猜。CHO 猜培训和绩效怎么改。CIO 猜系统和数据底座怎么搭。业务负责人猜哪个场景值得试。AI 转型负责人猜到底是追 Demo 还是追流程结果。猜来猜去，最后每个部门都在做自己认为正确的 AI 转型。

这就是为什么很多企业 AI 动作很多，组织结果很少。一号位没有定义组织目标，下面所有动作都会散。老板要先说清楚：这家公司不是要多几个会用 AI 的员工，是要把哪个组织功能变得更强、更稳、更可复制。

第二件事：能力承接

CEO 第二件不能外包的事，是判断：个人变快以后，组织功能有没有变强？这件事有个名字，叫能力承接。

一个员工用 AI 写报告快了，不等于组织会写报告。一个运营用 AI 做数据分析快了，不等于组织有数据分析能力。一个销售用 AI 做方案快了，也不等于公司销售系统升级了。只要能力还停在某个人身上，它就是个人外挂。个人外挂很有用，但它不是组织能力。

组织能力要看结果：这个组织功能的产出是不是更稳定、更多、返工更少？它能不能被新人复用，换一个人以后是不是仍然能跑？如果答案是否定的，说明 AI 只是放大了几个人，还没有进入组织。

很多 CEO 会被早期提效骗到，因为早期提效很好看：某个人效率翻倍，某个小组产出变快，某个 Demo 结果惊艳。但 CEO 必须追问：这个效果能不能从一个人迁移到一个团队，从一个团队迁移到另一条流程，从一次试点变成持续机制？

这才叫能力承接。AI 是否有机地成为组织的一部分，最后要看它能不能承接业务战略目标。接不住，AI 就是外挂；接住了，AI 才是组织运行系统的一部分。所以你回去先挑一个已经见效的 AI 场景，只问一句：换个人来，它还跑不跑得动？答不上来，那就还是外挂。

第三件事：组织代价

CEO 第三件不能外包的事，是决定：为了这次转型，组织愿意付什么代价，又绝不能牺牲什么能力？

很多老板听到组织代价，第一反应是裁几个人、调几个岗、砍几块预算。这些只是表层代价。真正危险的代价，是组织能力被削薄：信任损耗、老员工经验断层、责任真空、流程无人维护、关键判断无人兜底，都会让组织能力变薄。

人少了，不一定叫组织变轻，有时候只是组织变薄。

变轻，是结构更精简，但承重能力还在；变薄，是墙还站着，里面却被掏空了。

AI 转型里最危险的一刀，是看到效率提升以后马上裁人。能不能裁？当然能。但前提是组织结果稳定、责任链清楚、知识已经沉淀、流程有人维护、异常有人兜底。这些没有证明之前，裁掉的不是成本，可能是组织能力。

CEO 不能只问“省了多少人”。还要问：省出来的产能怎么入账？被 AI 接走的任务由谁负责？被释放出来的人去做什么更高价值的事？如果人真的要少，哪些判断、经验和责任已经被组织接住？这几个问题，CHO 和 CIO 都不能替 CEO 决定。因为这是组织取舍，而组织取舍就是一号位的事。

不过，组织代价这件事还有另一面，而且这一面更容易被漏掉：不做决定，同样有代价。

我接触过一家传统制造企业，主要工作是大批量的重复性图纸设计——同类结构、同类规格，改改参数就是一版新图。方案我们已经做出来了。一年多之后回头看，人员配置没有变化，产能去向也没有安排。这家公司没有付出任何裁员代价，也没有付出任何转岗代价。但它付出的是另一样东西。一年多，是四个季度的复盘会，是几轮预算调整，是一批新人从入职做到熟手。这些都过去了，那条图纸流程还是老样子。

这笔账很少有 CEO 主动去算，因为它不出现在任何一张报表上。裁员会上会，转岗会上会，预算调整会上会。

唯独“我们决定再等等”不会上会——它甚至不以决定的形式存在，只是一次又一次地没有被排进议程。

所以 CEO 在谈组织代价的时候，要同时问两个方向的问题。往前问：如果我们做这个决定，组织会失去什么能力？往后问：如果我们今年不做这个决定，一年以后组织会站在哪里，那时候的差距还追不追得回来？

只问前一个方向的老板，会显得很谨慎，其实是在单边计价。谨慎和拖延在会议室里的表现完全一样，区别只在于有没有人把第二个方向的账也摆出来。一号位的责任，就是保证这张桌子上两笔账都在。

CEO 五问

三件事落到会上，可以变成五个问题。

第一问：这个组织到底要变成什么样？不要只写愿景，要写组织交付方式。

第二问：AI 进来以后，公司靠什么组织方式交付结果？靠人堆、靠牛人、靠流程，还是靠人机协同的责任单元？

第三问：个人被 AI 放大的能力，怎么变成组织能力？看结果，不看热闹。

第四问：哪些岗位、流程和责任必须被重写？补工具解决不了组织断点。

第五问：这次转型愿意承担什么组织代价，又不愿意牺牲什么组织能力？

这五个问题不需要一次答到完美，但 CEO 必须自己答第一版，不能转给下属。下属可以做方案，却不能替 CEO 做战略前提判断。

CEO、CHO、CIO 怎么分工

CEO 不是要亲自抓所有事，CEO 万能论也是错的。AI 转型需要分工，但分工之前必须先把界面说清楚。

这里先看转型核心五角：CEO、CHO、CIO、业务 owner、AI 转型负责人。这五个角色承担的责任缺一不可，也不能互相替代；完整责任图还会在下一章加入独立的治理守门角色。CEO 把组织目标丢给 CIO，最后会变成技术项目。CHO 只做培训和能力模型，组织接不住。CIO 只看系统能不能上线，就没人翻译业务到底要什么。业务 owner 不背结果，AI 项目就会变成创新部门的展厅项目。AI 转型负责人如果只是个协调员，没权限、没有组织理解、没有业务闭环，这事也玩不转。

所以 CEO 真正要抓的不是所有动作，是三个前提：目标是不是清楚，能力有没有接住，代价有没有想明白。反过来，CEO 不需要亲自选择每一个工具、写每一条流程、主持每一场培训，也不需要亲自盯每一个 agent 的效果。这些都可以授权，但授权之前，方向和边界必须定清楚。否则下属做得越勤奋，组织越可能跑偏。

方向和边界不是细节，是地基。地基不稳，后面全是补丁。谁抓什么、谁只能补方案，看下面这张表。

CEO 三件事决策台

这章可以压成一张 CEO 决策台。它不是给下属填的表，是给一号位自己看的。

CEO 必须抓的事	CEO 要回答什么	下属可以做什么	如果 CEO 不答，会发生什么
组织目标	AI 进来以后，这个组织到底要变成什么样？	CHO/CIO/业务负责人补方案	各部门各做各的 AI，动作很多，结果很散
能力承接	个人提效如何变成组织能力？	业务 owner 做试点，CHO 设计机制，CIO 支撑系统	组织只多了几个 AI 高手，没有形成可复制能力
组织代价	愿意付什么代价，不牺牲什么能力？	财务测算、HR 方案、业务影响评估	一提效就裁人，把判断、经验、责任和流程一起裁掉

这张表最重要的，不只是“分工清楚”，是把 CEO 不能外包的前提判断和下属可以执行的动作分开。很多公司 AI 转型失焦，就是把这两层混在一起：CEO 以为自己授权了，下属以为自己接住了，结果谁都没有回答最前面的组织问题。

五种抓错模式

要判断一家公司是不是 AI 时代的叶公好龙，可以看五种模式。

五种错误长着同一副面孔：用看得见的动作，回避看不见的组织变形。

第一种，系统上了，流程没动。采购、部署、账号分配全都办完，审批、复核的路径一行没改。AI 生成的东西还得按旧流程走一遍，审批人不知道怎么评，大量内容退回人工重做。AI 没提效，先添了个新堵点。

第二种，培训做了，结果不验收。完成率很好看，三个月后业务结果、绩效口径、流程节点原封未动，人自然缩回原样。培训只是准备动作，不是转型本身。

第三种，演示跑通了，责任没画。上线之后没人每天必须用，没人背指标，没人说得出出错谁兜底。Demo 不是死在技术上，是死在责任链上。

第四种，AI 有效果，立刻裁员。五个人的活，三个人加 AI 能干，就把两个人先裁了。裁完才发现，客户判断、异常处理、经验传递、流程维护，都在这两个人身上。以为在变轻，其实在变薄。

第五种，用狠话动员代替组织设计。“不用 AI 就出局”“不跟上 AI 就淘汰”，短期很有气势，员工也会紧张。可这些话没有回答任何一个组织问题：岗位怎么重写，出错谁负责。

问题没有答案，狠话越多，员工越会防御。真懂组织的人不靠狠话推动 AI 转型，他把流程、责任、知识、绩效和信任一起改。

叶公好龙的根因

AI 时代的叶公好龙，不是因为老板不聪明。很多老板很聪明，也很有商业直觉；问题是他们把 AI 当成了一台更高级的信息化工具，没当成一次组织操作系统的重写，所以上来就会问错问题：RAG 要多少钱？知识库多久能上线？agent 能不能替几个人？CIO 能不能负责？年轻人是不是更懂 AI？

这些问题不是不能问，但如果只问这些，就说明还停在工具层。真正的老板问题应该是：我们的知识到底在哪里？谁有权把经验沉淀成组织资产？员工为什么愿意交真东西？AI 参与以后责任怎么重画？如果这个流程变快了，后面哪个环节会堵？如果人少了，哪些组织能力会变薄？

两组问题的差别，就是叶公好龙和真转型的差别。前一组在买工具，后一组在改组织。

一号位先写一页纸

CEO 今天不用先开大会，先写一页纸，标题就叫：我们这轮 AI 转型，到底要把组织变成什么样？

下面只写五个答案：组织目标是什么？未来靠什么组织方式交付结果？个人提效怎么变成组织能力？哪些岗位、流程和责任必须重写？愿意承担什么代价，又不牺牲什么能力？

写完以后，再拿这页纸去看现有的 AI 项目。培训有没有服务这个目标？系统有没有进入这条组织路径？Demo 有没有验证能力承接？裁员或者增产，有没有先证明组织结果稳定？

如果对不上，就不要再说这是 AI 转型。它只是 AI 动作。

AI 动作可以很多，组织转型只能从一号位的判断开始。

CHAPTER 19

谁来负责 AI 转型

一家公司决定上 AI，半年过去，我去问：这事现在谁在管？得到的答案通常是四个人互相指。老板等 IT 给方案，IT 等业务提需求，业务等 HR 改岗位，HR 等老板拍方向。

会都开了，系统也上了一点，培训也做了一轮。但真正的问题还在原地：这条流程谁改，系统谁建，结果谁背？

这个问题没人回答，AI 转型就是一个很热闹的空场。

这件事在现实里通常有两种坏法。

两种半截工程

第一种坏法，是把它做成 IT 项目。

系统上线了，接口打通了，测试通过了，供应商也交付了。可业务部门回到日常工作里，发现这个系统不是他们真正需要的东西。它能跑，但没人用。或者有人用，但流程没改，结果没人验收。CIO 说系统没问题，业务说系统不解决问题，项目就挂在中间。

这不是 CIO 的错。是组织里没人负责把业务问题翻译成系统要做的事，也没人负责把系统能力翻译回业务能用的流程。两头都缺一个翻译。

第二种坏法，是把它做成 HR 培训。培训做了，工具课上了，员工也开始试。年底汇报里有使用率，有学习人数，有优秀案例。但老板问一句：哪条流程被改写了？哪个岗位的交付结果变稳定了？哪些经验沉淀成组织能力了？

答不上来。这笔账不该记在 CHO 头上。根子在组织把 AI 转型当成了个人能力提升，不是组织系统改造。

美团“养虾”以后，谁来把这件事接住

美团的王莆中，在一次由网易转载的公开演讲中，回看了美团 2026 年上半年的 AI 推进。他把随着 OpenClaw 兴起开展的一轮使用和科普叫“养虾”——先把 AI 发下去，先让人用起来。可工具一热，真正的账就冒出来了。按他的讲述，成

本、权限安全和错误输出随后开始给真实经营施压；4 月起，公司开始从顶层思考 AI 转型、在各大事业部设置 AI 组织；到六七月，大量项目一边赛马，一边梳理方法。[①]

这就是 AI 最容易把公司带进去的场面：人人都在养虾，没人先把三笔账摊在桌上——钱谁付，门谁守，AI 说错话以后谁把它拉回来。别急着把这句话读成对美团的褒贬；公开叙述也不足以支持那样的结论。但他后面把话说得更直：企业 AI 变革不是丢给 CTO、CIO 或产研的技术项目，是业务、组织、技术结合的一号工程；落到具体流程，HR、业务、技术必须绑定。[②]

所以真正值得照镜子的，不是“美团也在养虾”，是你们公司现在这件事谁来接住。把 AI 当 IT 项目，流程没人接；把它当 HR 培训，岗位和绩效没人改。最后就会变成一场所有人都很忙、没人背结果的组织表演。工具可以先发，责任不能后补。

两种坏法看起来不同，本质一样：角色界面没有画清楚。

所以在明确一号位必须亲自抓什么之后，还得补一张责任界面表。不是为了给每个人划地盘，是为了防止组织把 AI 转型拆成几个互相甩锅的半截工程。

AI 转型不是 IT 项目，不是 HR 项目，也不是业务部门自己摸索。它是一项组织工程。组织工程的第一步，是把角色边界说清楚。上一章的五个角色是转型核心五角；这里把治理负责人补进来，展开完整的六类责任图。所以先别急着看下面六个角色写了什么，先在心里回答一句：你们公司现在这两种坏法，正在犯哪一种。

六类角色，各自回答什么

CEO，也就是一号位，负责回答组织目标。这个组织到底要变成什么样？AI 进来以后，哪些业务功能要被重写？公司先要降本、增产、提质，还是先要把组织能力沉下来？哪些组织代价能承担，哪些不能？

---

①据网易号“互联网坊间八卦”转载《美团王莆中最新演讲全文，中小企业如何正确的使用 AI？（实录）》，2026-08-14；内部留存快照第 4–5 页，SHA-256：5b366254b37a5f5efd181ee905c11668f038e32f0b965595393fe1f6aeb7a1e6。本文仅据此归因演讲者的过程叙述，不将其视作美团官方复盘或独立经营审计。

②同上，第 5–6 页。演讲实录将企业 AI 变革表述为“业务设计+组织变革+技术应用”的系统工程，并提及 HR、业务、技术协同。本文将其作为外部管理者的公开观点，与本书责任界面框架对照；不主张该观点已被美团经营结果验证。

这些问题不能外包。因为它们不是专业建议，是经营取舍。CHO 可以提醒组织代价，CIO 可以提醒系统风险，业务 owner 可以提醒现场约束。但最后谁来定目标、定优先级、定授权？

答案只能是一号位。

如果一号位只说“你们去推进 AI”，但不回答组织要变成什么样，下面所有角色都会进入猜测状态。猜老板想降本，员工就防御。猜老板想创新，业务就做 Demo。猜老板想要短期数字，IT 就赶上线。组织会按不同猜测各自行动。

这就是失控的开始。

CHO 负责回答组织怎么接住。不是办培训，不是统计使用率，不是在能力模型里加一句“熟悉 AI”。这些事可以做，但不是核心。

CHO 真正要管的是五件事。人机搭配以后，岗位职责怎么重新设计。绩效怎么从看过程改成看结果。员工凭什么愿意把自己那套业务上下文交出来。组织记忆靠什么沉下来。员工的信任靠什么保住。

归结起来，CHO 要管的是人和机制。

AI 进流程以后，谁的工作被改变了？谁的判断还必须在场？谁的贡献过去看不见，现在要被记录？员工把自己的经验喂给组织 AI 以后，是被放大，还是被替代？

这些问题 IT 回答不了。

这里最容易被低估的是最后一条。我见过的岗位重写卡点，多数不是 JD 写得不好，是员工根本不知道这场变化要他怎么配合，再加一层担心：这是不是替代的前奏。员工不是天然反 AI，是不知道 AI 会把自己的工作改成什么样。这两层没处理，只改职责表，就是纸面变革。

如果 CHO 不接，AI 转型就会停在工具层。员工学会了 AI，组织没有变聪明。个人变快了，知识还是散的。系统跑起来了，信任却掉下去了。

CIO 负责回答系统怎么稳。系统能不能接 AI，哪些数据可以开放，权限怎么分级，日志怎么留，安全怎么兜，技术债怎么控——这些是 CIO 的主场。

CIO 不该替业务决定流程怎么改，也不该替 HR 决定岗位怎么变。但技术底座必须由他守住。系统不能为了一个漂亮 Demo 把未来搞烂。数据不能因为业务想快就随便开。AI 输出不能绕过审计直接进关键流程。

CIO 的一句话标准是：系统能不能安全、稳定、可追溯地跑。

这件事不能交给供应商，也不能交给业务部门凭热情推进。供应商倾向于多接，业务倾向于快上，只有 CIO 必须对系统三年后还能不能跑负责。

业务 owner 负责回答真实问题。这是很多 AI 转型最容易漏掉的角色。

业务 owner 不是“提需求的人”。他应该是那条流程的现场负责人：知道哪里慢，哪里贵，哪里错不起，哪里只是看起来能自动化，实际承载客户关系、责任边界和组织记忆。

让最懂 AI 的人去找业务场景，容易做成拿锤子找钉子。更好的顺序是：让最懂业务的人，拿 AI 去重写自己的现场。

因为真实损耗在现场，隐性规则在现场，例外处理在现场，客户情绪和历史关系也在现场。

业务 owner 要交付的不是“我支持 AI”，是清楚说出：这条流程要改哪里，改完以后用什么结果验收，出了问题谁承担业务后果。

没有业务 owner，AI 转型会变成系统上线和部门配合。听起来有分工，实际上没有人背结果。

AI 转型负责人负责翻译和闭环。这个角色最容易被做成协调员。协调员负责开会、催进度、汇报材料，AI 转型负责人不能停在这里。

他要做的是组织翻译，两个方向都要翻。往里翻：把业务那句模糊的“我们这块能不能用 AI”，翻成流程节点、系统要求、责任边界和验收指标。往外翻：把系统的真实能力翻回业务听得懂的话，告诉业务哪些能做，哪些不能做，哪些能做但风险不能接受。

他还要做闭环。系统上线之后，反馈有没有回来？错误模式有没有被记录？context 有没有沉淀？规则有没有更新？下一条流程能不能因为上一条流程的经验跑得更快？

这个角色缺一块就会歪。没有 CEO 授权，他只是会议秘书。有授权但不懂组织，他会把事情推成项目管理。有授权也懂组织，但不懂 AI 的能力边界，他会被供应商和技术术语牵着走。

所以这个角色的门槛不是头衔，是三件事：懂业务现场，懂 AI 能力边界，有组织授权。三件缺一件，这个位子就白设了。

治理负责人负责规则、权限、审计和例外。这四件事听起来像刹车，其实是在让 AI 能被大规模用起来。哪些场景自由使用，哪些抽检，哪些复核，哪些

走审批，哪些干脆禁止？谁能让一个 AI 工作流上线？出了问题谁有权叫停？日志留在哪里？复盘怎么启动？例外怎么申请？

这些问题没人管，AI 应用就会自由生长。短期看很创新，长期看会变成权限、数据、质量和责任的烂账。

治理负责人不一定是一个独立部门。它可以挂在现有治理、法务、IT 风控或者 AI 转型办公室里。关键不是名字，是这四件事有人固定接住：规则、权限、审计、例外。

六类角色说完了，它们不是六个新增岗位。可以由现有岗位承担，也可能需要分工；兼任时要说明权限、资源和彼此制衡如何保留，不能让提出方案的人顺手包办独立验收。

第十三章引入的三类治理实践，在这里可以拿来检查职责表：结构上，谁被授权、能调什么资源；程序上，授权如何进入日常运行、变更和异常处置；关系上，受到影响的人能不能理解、提问并影响安排。填一个名字，只是开始找接口，连第一方面是否成立都还没证明。

一个用得上的验证方法

上面这六个角色写出来，看起来很整齐。但组织里最常见的情况是：表填完了，六个格子都有名字，转型照样空转。

填上名字和真接住这件事，中间隔着很远。

有一个办法可以快速检验这张表是不是真的。别从角色往下问，要从事故往回问。

挑一件 AI 已经参与的、有真实业务后果的事——一次对外的报价、一封客户回复、一份进了审批流的材料。然后按顺序问四个问题：谁下达的指令，AI 接触了哪些数据，AI 输出了什么，谁把这个输出推进了下一步。

这四个问题与第十三章的事实链相接，用来重建从指令到输出被采用的过程。记录接得上，可以继续查是谁依什么权限作出安排；记录接不上，就先补证据。能还原发生了什么，不等于权限、控制和责任都已有效。

我见过的情况是，大多数公司在第四问上卡住。前三问都能答：指令是某个业务同事下的，数据在系统里查得到，输出也留了记录。但”谁把这个输出推进下一步”——这一环经常没人答得上来。它不是被谁隐瞒了，是这一步在流程

设计里就没有被当成一个需要署名的动作。它被默认成一个流转。不是一次放行。

推进下一步的动作，要能够追到它所依据的授权。可以是当次人工批准，也可以是在约定范围内自动执行；不能因为当时没有人点按钮，就说责任悬空。真正要查的是：谁批准这种运行安排，范围有没有被守住，出了异常谁能介入，结果由谁持续承接。

所以责任界面表还要填：这条流程允许怎样推进输出，依据在哪，谁能调整或暂停这个安排。

异常也按第十一章的控制规则查。哪些可由系统按已核定的规则处理，哪些超出能力、权限或损失承接范围，必须升级到有权处理的人？自动处理是否有效，要看处置记录与结果；升级给人也要确认对方能及时处理，不能让通知代替解决。

日志记录了异常，只说明它被记下；处置完成并留下可用于更新的依据，才有可能进入组织学习。

这两问——谁放行，什么必须升级——比六个角色的职责描述更能说明一家公司的准备程度。职责描述可以照抄，这两问答不上来就是答不上来。

OIO 只是可选机制

至于 OIO 这种说法，要降噪。

先把这三个字母拆开。OIO 全称是组织智能办公室，一个专管“人加 AI 怎么一起干活”的机制[①]。名字是我起的，不是行业标准。

新部门不是前提。先核现有安排能否接住工作单元台账、上线与退役、把关标准、跨部门冲突和错误模式复盘；接得住，就不必换牌子。确有空白，才讨论补职责、补资源，还是另设协作机制。

每项职责都要有可找到的接口和实际履行条件。光有五个名字不够，光有一个办公室也不够。

①组织智能办公室，英文作 Organization Intelligence Office，缩写 OIO，为本书作者提出的说法，非行业既有专名。

开一场责任界面会

老板下周开会，可以直接拿下面这张表。表填不出来，说明组织还没准备好大规模推进 AI。填出来了但没有授权，说明它只是讨论稿。填出来了也有授权，下一步才是选一条流程，做 90 天试点。

这张表不妨试着用一场 90 分钟的责任界面会讨论。下面的时间分配是一种主持建议，不是经过验证的标准工时；复杂流程可能需要会前取证和后续专门讨论。

第一段 20 分钟，一号位只讲目标和边界。这次 AI 转型优先解决什么，不解决什么。是先降本，先增产，还是先把质量和稳定性提上来。哪些组织代价现在不能碰。

第二段 30 分钟，业务 owner 讲现场：流程哪里慢，哪里反复返工，哪里错不起，哪里看起来可以自动化、其实底下压着关系和责任。这个环节不能让技术先讲方案，否则整场会都会被工具带走。

第三段 20 分钟，CHO 和 CIO 分别讲承接条件。CHO 讲岗位、绩效、信任和 context 怎么接住。CIO 讲系统、数据、权限和审计能不能支撑。两个人不是互相汇报，是在同一条流程上对齐边界。

最后 20 分钟，AI 转型负责人把这张表收口：谁改流程，谁建系统，谁复核，谁验收，谁对结果负责，哪些信息必须进组织记忆。

会开完，如果只留下一份“继续推进”的纪要，这会就白开了。散会时必须留下三样东西：一条被选中的流程，一个明确的业务 owner，一张责任界面表。

没有这三样，AI 转型还在空转。

这里要特别防一个误读。

责任界面表不是让总部多一层审批，也不是让某个新办公室骑到业务头上。它的作用恰好相反：把每个角色能做什么、不能做什么提前说清楚，减少暗战，减少扯皮，减少出了问题以后临时抓人背锅。

好的责任界面要减少无效往返，但不会消灭摩擦。必要的复核、异议处理和跨部门协调都要花时间，也可能改变原来的方案。该比较的是：这些投入避免了哪些损失，是否值得，而不是会开得越少越好。这也呼应了第十三章的关系性治理：参与有成本，意见进入决定才有意义。

工具：AI 转型责任界面表

角色	回答什么问题	交付什么结果	不能越界什么
CEO / 一号位	组织要变成什么样，优先级是什么，能承担什么代价	目标、授权、优先级、组织代价边界	不能只喊“推进 AI”却不定取舍
CHO	岗位、绩效、信任、组织能力怎么接住	人机结合职责、结果口径、业务上下文贡献机制、信任沟通	不能把 AI 转型做成培训项目
CIO	系统、数据、安全、工具怎么稳	系统边界、数据权限、审计留痕、技术债控制	不能替业务和组织做结果取舍
业务 owner	真实流程哪里慢、哪里贵、哪里错不起	真实场景、验收指标、业务结果责任	不能只提需求不背结果
AI 转型负责人	业务、AI、流程、责任之间怎么翻译并闭环	流程改造方案、责任链、指标、反馈和 context 闭环	不能退化成会议协调员
治理负责人	哪些场景放开、审批、禁止，异常怎么处理	使用分级、权限规则、审计机制、例外处理	不能只写制度不接入日常运营

这张表别当模板存档，下周就拿去填。先填最后一列——每个角色不能越界什么。这一列比前三列难填，也比前三列值钱：能做什么大家都愿意认，不能做什么才是真界面。

CEO 亲自抓，不代表 CEO 亲自干所有事。真正要画清的，是每个角色回答什么问题、交付什么结果、不能越过什么边界。责任界面越早清楚，AI 转型越不容易变成一场互相等待，也越不容易在出事以后临时找人背锅。

CHAPTER 20

传统组织怎么双元孵化，不在高速上换轮胎

一家做了二十年的公司要上 AI，老板问我的第一句话，往往不是”值不值得”，是”能不能别停”。这个季度的订单还要交，钱还要收，客户投诉还得当天回。

这就是在高速上换轮胎。

公司不能为了 AI 转型，把主业务拆开，把所有流程停掉，把所有岗位重新洗一遍。先看旧安排在哪些地方接得住、哪些地方接不住。接得住，保留；接不住，再改。全拆，组织可能先被自己拆垮。

当新工作方式还需要试，而主业务又不能停，传统组织就面临两种不同的要求：一套安排负责稳定交付，一套安排负责探索验证。两种能力都要有，不等于必须另建两套组织。

稳定交付这套，我叫利用系统。它管今天的收入、客户、质量、风险和日常运营。探索新东西这套，我叫探索系统。它管试错、打样、找新路、验证新工作方式。

我在光华读 MBA 时，张一弛老师把这两套打法讲成“高速路”和“丛林”：一边靠规则和效率把今天的业务跑稳，一边承认路还没长出来，要给探索留下空间。那堂课里，我第一次系统接触“双元性组织”。后来真正在企业里做 AI 组织改造，我才吃透：这不是一个漂亮概念，是传统公司在不停业的前提下换系统的活法。管理学里，Tushman（塔什曼）和 O’Reilly（奥赖利）1996 年就系统写过这套两条腿的组织能力[①]。名字不重要，重要的是两条腿都得有。

AI 改造不能一上来就砸进利用系统。利用系统背着业绩和责任，最怕不稳定，不会天然欢迎一个还没跑稳的新东西。但 AI 也不能永远待在探索系统。永远隔离，只会长成一个创新展厅：Demo 很好看，主业务照旧。

---

①本章“高速路/丛林”的场景判断，来自作者在北京大学光华管理学院张一弛老师《人力资源管理》课程中的学习与后续实践内化。双元性组织的经典文献为 Michael L. Tushman、Charles A. O’Reilly III，《Ambidextrous Organizations: Managing Evolutionary and Revolutionary Change》，California Management Review，1996 年，38 卷 4 期，第 8–29 页，DOI: 10.2307/41165852。

正确路径是：先在探索系统打样，再把方法沉淀出来，最后嵌回利用系统。这不是为了躲开主业务，是为了保护主业务，同时把新的组织 OS 验一遍。

为什么一步到位最危险

很多老板喜欢一步到位：全员开账号，全员做培训，全线铺开，全公司一起上 AI。听起来很有决心。但组织不是软件安装包，不是点一下就能全量更新。

传统组织里的每条流程，背后都压着责任、利益、习惯和风险。AI 一进去，不只是多了一个工具，是在改这条流程的运行方式。流程还没被试过，责任还没画清楚，异常还没人兜底，知识还没沉淀，绩效还按旧口径算——这个时候把 AI 全线铺开，你得到的不是转型速度，是系统性混乱。

这种混乱有个很好认的形态，我叫它假繁荣：到处都在试，人人都在用，周会上人人有话说，却拿不出一条真实流程的结果账。

CEO 以为公司动起来了，其实只是噪音变大了。

所以你回去先别看账号开通率，先问一句：过去三个月，哪一条真实流程因为使用 AI 取得了可核查的改善，总投入和后果又是什么？拿不出依据，就还不能宣布转型见效。流程走法变了，不是成绩；旧走法仍然有效，也不必为了证明转型硬改。

三阶段路径

传统组织要把 AI 长进去，可以分三步。

第一步，特种小队打样。选一条真实业务流程，抽 3 到 10 个懂业务、懂组织、愿意拥抱 AI 的人，给他们授权，让他们在可控范围里试。

这支队伍不是常规项目组。常规项目组交付的是确定结果，特种小队交付的是第一手认知。这条流程里 AI 到底能做什么？哪里能切进去？哪里看起来能切、其实碰不得？人还要守在哪个判断点？出了错谁兜底？流程后面哪个环节会堵？

这一步不能背短期 KPI。不是说不要结果。探索阶段的结果本来就不只有产出数字，还包括失败经验、边界认知、责任问题和流程改写方案。

这里有一个选择题，很多公司在第一步就选错了：实验对象到底选什么？

常见的两个错误答案，一个是选单个岗位，一个是选一个部门。

选单个岗位太小。一个岗位被 AI 改造以后，你能看到的只有这个人变快了。看不到他上游的输入有没有跟着变，看不到下游的接收方有没有跟着变，更看不到他省下来的时间有没有真的流向别处。岗位是组织的切面，不是组织的单元。在切面上做实验，结论没法往上推。

选一个部门又太大，而且太糊。部门是行政单位，不是交付单位。一个部门里往往同时跑着好几条互不相干的流程，人员、职责和结果指标全混在一起。你把 AI 放进去，最后说不清是哪一条流程变好了，也说不清它为什么变好。

更合适的实验对象，是被一条流程串起来的组织功能器件。

这个词有点拗口，但它指的东西很具体：内容生产、线索转化、投放管理、客服响应、招聘筛选、合同审核、项目交付——就这一类。它们长得都一样：有明确的触发和终点，有固定的角色配合，有能被验收的交付结果，也有清楚的责任边界。它比岗位大，所以能看见协作；比部门小，所以能看清因果。

选定器件以后，做法是并行，不是替换。原来那套继续稳定跑，旁边拉出一个 AI 增强版的小单元，做同一类任务。

然后看一件事：同一类任务，在传统分工下需要十个人怎么做，在 AI 加入以后，三个人能不能做出同等甚至更好的产出。

这个比较必须说清楚它在测什么，否则一定会被误读。它测的不是”能不能少七个人”，它测的是这个组织功能的交付方式能不能被重写。前者是人事结论，后者是组织结论。

同一组数据，如果先当人事结论用，后面就再也拿不到真实数据了——因为传统那一侧会立刻停止配合。

所以特种小队的第一份材料，不该是效率对比表，是这句话的书面版：我们在测交付方式，不在测编制。这句话谁说、什么时候说、对谁说，比数据本身更影响这个实验能不能跑完。

第二步，能力中心沉淀。打样成功不是结束。很多公司恰恰死在这里：小队跑通了，老板很高兴，开了个庆功会，然后小队散了，人回原岗了，方法也跟着散了。下次另一个部门再做，又从头摸一遍。

这不是组织能力，只是一次幸运。

能力中心要干的活，是把”这一次是怎么跑通的”写成别人能照着用的方法。输入是什么，输出是什么。AI 在哪个节点介入，人在哪个节点复核。异常怎么升级，权限怎么开，数据怎么留痕，结果怎么验收。员工怎么沟通，哪些坑不要再踩。这些东西要落成模板、清单、流程图和复盘记录，不能只挂在几个人嘴上。

没有这一步，探索系统和利用系统之间就没有桥。

第三步，业务嵌入。方法沉淀以后，不能永远锁在中心团队里。中心团队管标准、治理、方法论和复盘机制，具体打法要嵌回业务。

每条业务线都要有人能把 AI 带进现场。这个人不一定叫 FDE，也不一定是技术岗，但必须懂业务、懂 AI 能干什么、懂流程和责任在谁身上。

总部定底线，前线定打法。总部管的是：哪些场景不能碰，哪些权限必须审，哪些输出必须复核，哪些指标必须看。前线管的是：这条业务到底怎么用，怎么改，怎么跑，怎么把问题反馈回来。

总部连前线每一步都要管，业务会死。前线完全自由生长，总部会失控。嵌入的关键，就是中央有标准，前线有活性。这三步走完，你手里才算有了一套能复制的组织打法，不是一次运气好的项目。

每个阶段怎么验收

双元孵化最怕阶段名字起得很好听，验收标准却糊成一团。打样到底算不算打出来了？沉淀到底沉淀了什么？嵌入到底有没有真的进业务？这三个问题不说清楚，双元孵化就会变成另一套项目包装。

第一阶段验收的不是 ROI——不是投入产出比，不是这笔钱多久回本——是认知。AI 能做什么、不能做什么？哪个节点必须有人？哪个输出可以自动生成？哪个异常必须升级？哪类数据不能碰？哪一段流程因为 AI 变快了，后面又在哪里开始堵？这些问题答清楚，打样才算有结果。只拿出一个 Demo 说”AI 能跑”，不算。Demo 能跑，不等于组织知道怎么接住。

第二阶段验收的不是项目复盘会，是可复制方法。别人拿着你这套模板，能不能在另一条相似流程里少走弯路？业务 owner 知不知道自己要准备什么输入？CIO 知不知道系统、权限、数据、审计怎么配？CHO 知不知道岗位、绩效、信任沟通怎么接？这些东西还在几个人脑子里，就不算沉淀。

第三阶段验收的不是推广数量，是业务自己能不能跑。这条业务线有没有自己的 owner，有没有自己的 AI 触角，有没有日常复盘？异常有没有回到责任链？新方法有没有写进绩效和流程文件？如果还得靠中心团队天天推着业务用，那不叫嵌入，那叫外包式支持。

三个阶段的验收口径本来就不同。阶段一看认知是不是变清楚了，阶段二看方法是不是可复制了，阶段三看业务是不是能自己跑了。

阶段	试验侧留下什么	主业务侧怎样承接
限定打样	能力边界、失败记录、流程改写与控制方案	原业务继续运行，试验不自动替换主流程
方法沉淀	输入输出、权限、异常处理、验收与复用方法	接手者核准备条件，方法不只留在小队脑中
有条件嵌入	真实运行、业务 owner、监测和复评记录	满足接入条件后使用；新问题反馈到方法和规则

这不是三张日历，也不是必须新建三个部门。证据不足就停在该补的地方，不能用阶段名字代替验收。

什么时候转入主业务

试点什么时候可以嵌回主业务？太早，主业务会被扰乱。太晚，试点会变成孤岛。我建议看四个信号。

第一，结果稳定。不是某一次跑得特别漂亮，是连续几轮都能稳定产出。AI 的输出质量、人的复核成本、异常数量、返工比例，这四个数得基本稳住。

第二，责任清楚。AI 做了什么，人复核什么，谁对结果负责，异常谁接，出事谁兜底。没有责任链，不要嵌回主业务。

第三，方法可复制。不是原小队的人自己会做，是另一个团队拿到模板后也能照着跑。复制不必一模一样，但不能每次都重新发明。

第四，信任没崩。传统团队没有把试点理解成组织偷袭，业务现场还愿意交真流程、真问题、真 context。

最容易被忽略的是第四个。很多老板只看结果稳不稳、责任清不清、方法能不能复制，没看到信任其实已经崩了。等到嵌回主业务，软抵抗就出来了：流程表面接进去，现场实际绕着走；员工嘴上配合，真正关键的经验一个字不

交。这已经不叫嵌入了，这叫被组织的免疫系统排斥。所以嵌回之前，先拿这四条对一遍——四条全绿再动，缺一条就先补那一条。

双元试点清单

老板别先问”哪个工具最好”，先问这条流程适不适合进双元试点。下面七项，一项一项对。

检查项	要问的问题	不合格信号
流程真实	这是不是一条每天都在跑的业务流程?	只是展示场景，没有真实业务压力
痛点明确	这条流程现在卡在哪里?	只是觉得 AI 很酷，没有明确卡点
输入稳定	这条流程有没有相对稳定的输入?	每次输入都不一样，无法复盘
输出可验	结果能不能被验收?	只能说”感觉更快”，没有结果口径
风险可控	试错会不会打穿主业务?	一错就影响大客户、现金流或合规
owner 清楚	谁对这条流程结果负责?	创新团队做，业务不背结果
信任可守	传统团队是否知道这不是偷袭?	一试点，员工就以为要裁自己

七项里有三项不合格，就先别急着做。不是 AI 不值得做，是这个场景还没准备好。传统组织最怕的，就是拿一个没准备好的场景去证明 AI 不行——那不是实验，那是在烧信任。

老板会上三问

这套方法可以压成三句话，下次经营会上直接拿去问。

第一问：我们现在是在探索，还是在规模化？如果还没看清 AI 在这条流程里能干什么，就别拿规模化的指标去压它。探索阶段就要承认自己还在找路。

第二问：试点成功以后，谁负责把方法写成别人能复用的东西？没有这个 owner，试点成功也只是这一次成功，下一个团队照样从头踩坑。

第三问：主业务团队为什么愿意配合？别默认他们会配合。要问他们能得到什么，担心什么，贡献出来的经验怎么被看见，试点数据会不会哪天被拿去直接裁人。

这三问很朴素，却能把很多假转型挡在门外。假转型只回答”工具怎么上”，真转型必须回答”组织怎么接”。这不是保守，是在控制组织风险。

真正的激进，不是把 AI 一口气铺满全公司，是用一条真实流程跑出能复制的组织方法，再一条一条扩出去。能复制，才算组织能力。不能复制，只是一次试运气。

老板要抓的是前者。别被热闹骗了。

信任地雷

双元孵化里最大的地雷，不在技术，在人心。

场景很常见。一个 AI 小队在某条业务线上跑出了效果，老板一听，脱口就问：“那是不是可以少几个人？”这句话传到传统团队耳朵里，人心当场就凉了。

原来让我们配合，是为了证明我们没用。

从这一刻开始，他们不会再交真实的业务上下文。他们会配合，但会防御。会说标准流程，不说真实流程。会给安全答案，不说真实问题。会在会上点头，回到现场把关键经验藏起来。这不是员工坏，这是理性自保。

所以双元孵化必须有一条信任规则：试点阶段的中间数据，不能直接拿去做裁员决策。它可以用来复盘、改流程、判断产能方向，但不能刚看到效率提升，就翻译成”少几个人”。否则探索系统刚长出来，就会被利用系统的降本逻辑掐死。

宏观上要进取，微观上要防御。公司整体要敢往前试，但每一次具体试点，都得保住业务现场愿意说真话的条件。没有这个条件，双元孵化就会变成双元对抗。

所以每个双元试点的启动方案里，都要夹一页信任维护计划。这一页不用写得花哨，写清楚四件事就够。

第一，试点不是用来当场裁人的。阶段数据可以用于复盘、流程改写和产能判断，但不能没经验证就直接变成人员优化的依据。

第二，传统团队不是被审判的对象，是经验提供者、流程校验者和边界识别者。没有他们，探索小队根本拿不到真实业务。

第三，贡献要被看见。谁交出了真实流程，谁点出了关键风险，谁帮 AI 的输出把住了质量关，都要写进记录。

第四，成功以后角色怎么迁移。新方法真跑通了，传统团队里哪些人可以当第一批运营者、复盘者、训练者、业务嵌入者，这件事要提前设计，不能等人心散了再来安抚。

信任维护计划不是温情材料，是试点能不能拿到真实业务上下文的生产条件。

不在高速上换轮胎

传统组织的 AI 改造，真正考验的是节奏。太慢，会被时代甩开。太猛，会把自己拆坏。所以别一上来就全线铺开，也别永远停在创新小组。

正确的节奏是：一条流程打样，一套方法沉淀，一批业务嵌入。先用探索系统找路，再用能力中心修桥，最后让利用系统接住。

所以这个月先别谈全员铺开。先在你们公司挑一条真实流程出来——每天都在跑、有痛点、输出能验收、责任人明确的那种。拿本章那张七项清单对一遍，够格就照前面说的抽 3 到 10 个人开工。

开工那天再补一件事：把”我们在测交付方式，不在测编制”这句话，当着传统团队的面说出来。这句话说不出口，实验从第一天起就拿不到真话。

这才叫不在高速上换轮胎。车还在跑，轮胎也要换。办法不是让所有人跳下车一起拆，是先找一条可控路段，换出一套方法，再把这套方法变成组织能力。

传统组织能不能拥抱 AI，不取决于口号多响，取决于它有没有本事把探索和稳定同时管住。

PART 07 · 第七部

90 天试点和现场角色

组织改造不能靠喊话，也不能靠一次漂亮 Demo。

可以先选一条真实流程，做一次有边界的试点。90 天是本书提供的起步窗口，具体期限要按任务、风险和结果出现的周期事前约定。

但试点不能做成玩具。它要有明确痛点、可核查的输入输出、约定结果、负责人，以及授权、控制和复盘安排。到期不能只交一个“AI 能做这件事”的演示，要交实际结果、总投入、错误与处置记录。原安排有效，可以沿用；发现失配，再改流程和分工；拿不到值得继续的结果，也可以停。方法是否能交接、能复制到下一条流程，另验，不能凭第一轮跑通就宣布。

这也是为什么 AI 时代会需要新的现场角色。

这个人不只是懂 AI，也不只是懂业务。他要能把模型能力翻译成业务动作，把业务现场翻译成系统需求，把试点经验翻译成组织机制。

外面有人把类似角色叫 FDE。名字可以引用，但真正重要的是功能：把 AI 带进业务现场，把业务现场沉淀成组织能力。

全书最后要交代的，也正是这条路怎么走完：

> 组织怎么从第一个 AI 试点，走到可复制的组织 OS 升级？

CHAPTER 21

90 天、180 天、1 年如何把 AI 组织改造跑成闭环

我看过一家企业，四个动作一个不缺。规划做完了，POC 跑通了，报价也给了，推动的副总还拍了板。

然后什么都没发生。

技术那一侧的答案，它是买到了。组织那一侧的答案，一个都没开始准备。

停在这个位置上的 AI 项目不止这一个。这类项目在技术侧通常都挑不出毛病：系统在技术层面全部达标，供应商的交付报告写得很漂亮。它们的共同点不是技术验证失败，是技术验证成功之后，没有人接得住。

我讲这个开头，是因为老板手里那张 AI 转型路线图，通常回答不了这件事。让人做一张出来，交上去的十有八九是这样一张表：第一季度部署知识库，第二季度上智能客服，第三季度跑数据分析，第四季度全员普及。有逻辑，有进度，有时间节点，放进 PPT 汇报给董事会，一页纸显得很充实。半年后，它会变成一个沉默的证人。工具部署了，可没有哪条流程被改写。培训做了，可没有哪个岗位的工作方式真的变了。Demo 给管理层看过了，可落地的阻力不会因为 Demo 通过就消失。这种东西不叫路线图，叫采购计划。

这种停摆不是个案，它的样子往往还很体面——没有人反对，也没有人推进。项目停在审批环节，业务线说没有人手做接口对接，IT 说预算不在这一年，运营说流程还没梳理清楚。三句话单独拿出来听，每一句都成立。合在一起，就是没有人负责。

组织没准备好接住一个已经通过验证的系统，它就在技术可行和组织落地之间的空隙里悄悄搁置了。注意这个空隙里没有反对票。AI 转型很少死于有人拍桌子说不行，它死于每个人都说「这事挺好，就是不归我」。

真正的路线图不告诉你什么时候买什么工具。它告诉你的是这几件事：哪条流程要被改写，谁对结果负责，90 天后看哪张账，180 天后改哪条制度，第一年结束时组织有没有学会复制。

第一切口：选老板能看见账变化的流程

第一步，不要选最复杂的流程。这是一个很常见的陷阱：很多老板觉得流程越复杂，AI 能改进的空间越大，一旦跑通，就能说服全公司。听起来合理，实践里却往往走不通。

最复杂的流程，通常也是权责最模糊的流程，涉及部门最多、例外最多、审批最多、历史包袱最多，管理层也最难在 90 天内看到结果。第一切口不要选最难的流程，要选老板能看见账变化的流程。

这里的账，不是财务报表里的某个科目。我建议先看流程损耗三账：时间账、等待账、知识账。

时间账，看员工在这条流程里到底花了多少工时，其中多少是重复劳动。等待账，看多少时间耗在等回复、等审批、等上游数据、等确认上。知识账，看这条流程里的专业判断，有多少已经写成了别人能用的东西，又有多少还只活在几个老员工的脑子里。

流程损耗三账都可以算，但不必同时建。挑切口的关键只有两问：哪一张账，老板每天都在肉身感受？哪一张账一旦摆出数字，老板会说“这个值得改”？

最适合当第一切口的流程，通常有三个特征。结果能被直接看见。改进能在 30 到 90 天内被测出来。有一个具体的人，愿意把自己的名字签在这件事上。

三个特征缺一个，项目就容易变成亮点工程——大家都觉得有价值，但没人真正负责。

这种项目的下场很好认：季度总结里它是亮点，年度复盘里它没踪影。

三个特征都齐了，也还是可能选错。有一条路我带着走了一个月，最后发现不能走。

那件事当时怎么看都对。客户要建联大量达人，需求明确得不能再明确。照着上面那三条挑，它一定会被挑中。

我想做的事很直接。让系统自动去加达人好友，自动发评论，自动私信。把建联这件事整个自动化掉。

技术上不难。做出来了。问题是两头都不答应。

一头是平台。这类社交平台对自动化动作很封闭，风控很紧，动不动就关号。这里有个反常的地方值得停一下：系统跑得越顺，账号反而越危险。而那是客户的账号，不是我的。

这一层其实是查得到的。做之前谁都能查到平台不欢迎这类动作。但查到的不是一条禁令，是一个概率——可能关号，不是一定关号。

一个概率是不会让人停手的。它只会被当成一项可以承受的风险，折进排期里，然后接着往下做。

另一头更要命。那一层是我们跟客户一起做测试的时候，才想明白的。

单独加一两个达人，没问题。但客户真实的需求是要见大量达人。量一上去，性质就变了。

我一个月后才想明白这个「性质就变了」到底变的是什么。一两个达人被 AI 加，没有人会在意。那看上去只是一次效率高一点的联系。可成百上千个达人都这么被加，事情就不是效率了。它成了一种对待人的方式。而对方能感觉出来。

达人是被 AI 自动加的，加完之后又一路都在跟 AI 互动。他知道自己面对的不是一个人，是一套流程。到这一步，达人自己的感受不好。这件事没写在任何一版需求文档里，也没有哪个功能点对应它。

这件事一想通，结论就很清楚了。这是一个必须要人类介入的环节。

两头的代价，撞上去的方式完全不一样。平台那一头是明的。风控就在那儿摆着，做之前查一查也能查到。达人那一头是暗的。它不在系统里，也不在需求里，它在量里。做一个的时候它不存在，做一千个的时候它已经发生了很久。

我在当时的文档里写下的一句话是：这不是技术难不难的问题，是产品立场问题。

想明白还不算完。同一份文档里，我把它落成了一条产品原则：不要让 AI 越权执行关系动作。AI 不应该替人完成社交关系动作。写下这条的时候，我要

挡的不是某一个功能，是往后所有长得像它的功能。一次判断只挡一次，一条规矩挡的是往后每一次。

选切口的那三个特征，管的是这条流程值不值得动，不管这条流程该不该由 AI 来动。它们能帮你避开选错难度，避不开选错方向。

0-30 天：建账，定人

0-30 天只做两件事：建账，定人。

建账，就是拿三张账把这条流程的现状测一遍。

时间账，翻过去一个月：员工在哪几个节点上花掉了多少工时，其中多少是重复动作。等待账，把等回复、等审批、等数据的时间加总，再和实际干活的时间比一比。知识账，数一数这条流程里的判断标准，有多少已经落成文档，有多少还只存在于某个负责人的经验里。

这些数字一开始不必精确到小数点后两位，但必须能追溯到出处，不能只靠“大家感觉这里很慢”。感觉不能验收。

定人，就是在 30 天内把一个人的名字锁死，由他对这条流程的 AI 改造负完整责任。

这个人不是 IT 系统管理员。不是供应商项目经理。也不是“AI 项目组”这种没有身体的集体名词。他得是懂这条流程业务逻辑、能对结果负责的内部人。而且他必须清楚：90 天后自己要交什么。

账没建起来就选工具，是在赌博，赌 AI 工具的能力恰好和流程的真实痛点匹配。这个赌局有时会赢，更多时候会输：工具买回来了，流程没有改变，最后有人说“我们试过 AI，没用”。不是 AI 没用，是你没有先把账和人定清楚。

三张账也不是财务部的新表格，是老板和业务负责人共同看流程的方式。时间账让人看见重复劳动，等待账让人看见组织摩擦，知识账让人看见经验是不是还困在个人身上。

很多企业一谈 AI 就立刻跳到工具。工具当然重要，但工具只是解法；没有把账算清楚，就只能拿到供应商递来的答案，拿不到从自己的组织问题里长出来的答案。

举四条常见流程，看看账不同、切口会差出多远。

客服流程最痛的是客户历史信息找不到，切口就落在客户画像和订单历史联动上。维修流程最痛的是前线拿不到设备说明书和历史案例，切口就落在知识检索和故障排查上。招聘流程最痛的是简历初筛和入职后的实际表现对不上，切口就落在筛选标准结构化和入职后复盘上。总办信息采集最痛的是每天手动追公开数据，切口就落在自动抓取、入库和初步分析上。

同样叫 AI，切口完全不同，因为账不同。所以 0-30 天不能省。省这一个月不是为了跑得快，是拿后面三个月去赌。

31-90 天：进入真实流程

31-90 天只做一件事：让 AI 进入真实工作流程。不是演示，不是沙盘，不是老板看一眼觉得有意思。是真实业务产出要开始依赖这条流程。

进真实流程之前，先把这条流程里的活拆成四类。

第一类是有规律可循的重复动作，AI 可以直接接手。第二类是要对信息做判断的分析活，让 AI 出初稿、人来定。第三类是要调用老经验的知识活，得慢慢把判断标准沉成知识库，再让 AI 去引。第四类是有人要为后果签字的责任节点，AI 只能打下手，替不了。

把这四类在流程上的分布标出来，才谈得上 AI 从哪儿切、怎么切、哪个节点必须留人复核。

跳过这一步直接上线会怎样？AI 大概率被用在它最擅长、但对业务影响最小的地方。那些真正耗时、真正承重、真正错不起的判断节点，一个都没被碰到。

试点范围不用大。三到五个人的团队，一个完整业务周期，一条真实流程，能产出真实业务结果就够了。

到这里最容易走偏的，是把试点当成一场技术验收：跑通了就算过，跑不通就换工具。回头看开头那家企业——它的技术答案是买到了，四个动作全做完，结果为零。所以试点真正要买的从来不是技术答案，是组织答案。

试点的目的不是证明 AI 多强，是把组织卡点逼出来。哪个节点的复核标准说不清楚？哪个判断没人愿意签字？哪部分知识始终没沉下来？哪段流程人机一协同反而更慢了？

这些才是 90 天最值钱的东西。它们全是组织问题，不是技术问题，只有真实跑过一遍才会露出来。

我当时是他们的实施顾问，在现场。技术这一关是过了的——系统我们交了，能跑，他们自己的政策和业务知识库也建起来了，验收没问题。

问题出在验收之后。员工手里有了 AI，工作方式还是老样子——还是按原来那套干活。

知识库也一样。建完之后很长一段时间没有人更新维护，上传的东西也不够；更要紧的是，组织流程没有跟着改，知识管理这件事本身没有真正做起来。库还在那儿，内容停在了上线那一天。

后来董事长下了命令：重新排查，重新上传。这一遍动员了多个部门，前后折腾了几个月。

同样一件事，做了两遍。第一遍是项目预算里的活，有节奏、有人管；第二遍是被追着做的。技术验收签字的那天，组织改造一步都还没开始——所以第 90 天那场会，必须有人坐下来认这笔账。

组织问题只有真跑一遍才会露出来——这句话反过来说就是：跑完这一遍，必须有人坐下来认这些问题。所以第 90 天的会不能开成总结会。

第 90 天：开决策会

第 90 天必须开一次复盘会。不是总结会，是决策会，只允许四个结论：停下、调整、扩大、复制。

切口被证明不适合 AI 介入，就停下，把原因写清楚。方向对、执行有问题，就调整了再继续。试点达到验收标准，就往更大范围扩。这条流程的经验能迁到另一条流程上，就启动第二条流程的三账测量。

开会必须带三类证据进门。

第一，三张账的数字和 30 天前比，哪张变了、变了多少。第二，复核机制跑这段时间，逮到过哪些 AI 错误、怎么处理的。第三，流程责任人对下一步有什么具体建议。

没有证据就不要拍板。AI 转型不是靠气氛推进的。靠气氛推进的项目，最后都会死在气氛里。

四个结论里，最难开口的是「停下」。

前面讲的那条自动建联的路，最后就是停在这个结论上的。但它不是被谁罚了才停的。那时候还没到见客户的日子，是我自己先刹的车。

停下不等于什么都不做。真正费劲的是停下之后重新划界线，一条一条划。

继续做的是这四件事。筛选谁该联系。生成建联的话术。推荐用哪个渠道联系。把内容复制好、排好队，提醒运营去发。

不做的是这三件事。自动发评论。自动加好友。自动私信。

界线划完，还有一件更细的事：产品里的说法跟着全改了。原来叫「自动触达」，改成「AI 辅助建联」。原来叫「自动发评论」，改成「生成评论建议」。

还有第三条，改的东西不一样。原来叫「触达失败」，改成「待人工处理」，或者「渠道不可用」。前两条改的是 AI 干什么，这一条改的是 AI 干砸了之后归谁。一件事失败了就摆在那儿，跟一件事失败了有人接，是两种流程。而要是那条路本身就走不通，说清楚是路的问题，人才不会白接。

这看起来像文字游戏，其实不是。界面上写着「自动」，用的人就真会以为可以撒手。措辞不是包装，是在告诉使用者这里该不该有人。

所以第 90 天那场会，选了「停下」并不算完。停下只是选完了。接下来要动手改的是三样东西。这条流程里哪些动作继续做，哪些不做。产品里那些暗示「不用管」的措辞。还有岗位职责里对应的那一段。三样都改完，这个「停下」才算真的落了地。

91-180 天：选择产能去向

90 天后，如果结论是扩大或复制，接下来 90 天面对的就不再只是技术问题，是人力和制度问题。AI 释放出来的产能流向哪里？如果没有答案，制度化就会停在纸面上，员工也会用脚投票。

员工心里那几个问号，其实很朴素。把时间省出来，对我是好事还是坏事？做得更快，会不会只是被要求做更多？AI 做对了算谁的绩效，做错了算谁的责任？

产能分配的方向只有四个：降本、增产、提质、重组。

降本，是把省下来的工时折算成人力成本的节约。增产，是把这些工时重新投进产品交付，把产出做多。提质，是把工时挪到判断型和知识型的活上，把结果做稳。重组，是重画岗位边界，该合的合、该拆的拆，长出一套新的职责结构。

四个方向可以组合，但不能都选。都选，等于没选。

含糊的产能分配，翻译成员工听得懂的话就是一句：你先努力，后果再说。这句话一出口，员工立刻开始保护自己。少贡献真实的业务上下文，少暴露真实的问题，少把自己的判断交给组织。

AI 转型最怕的不是员工不会用工具，是员工不再相信组织会公平处理 AI 带来的变化。

第 180 天：把临时做法变成正式规则

第 90 天认下来的那些卡点，靠开会认一次是不会自己消失的。复核标准说不清楚，下一个季度还是说不清楚；没人愿意签字的判断，换个人来一样没人签。所以 180 天要做的事只有一件：把 90 天逼出来的组织答案，从会议纪要挪进制度。

到了 180 天，衡量的东西要换一次。这时候的里程碑不再是系统运行了多少节点，是三件事。第一，负责这条流程的岗位职责说明书是否已经改写。AI 在哪些节点介入，人负责什么，如何复核，出了事谁处理，都必须写进职责。

第二，绩效口径是否已经更新。方向应该从“用了多少 AI、写了多少 prompt、调用了多少次工具”，转向“拿回了什么结果”：结果更稳定、产出更多、返工更少、经验更可复用。

第三，流程里积累的判断标准和处理经验，是否已经形成可调用的知识资产。它们不能只存在于个别员工的操作习惯里，也不能只躺在项目群文件里，而要进入组织记忆。

这三件事没做完，就别急着复制第二条流程。否则组织会同时跑两条没有制度接住的流程。复核标准、责任归属、绩效分配，这三摊混乱不会被分摊掉，只会叠起来。

180 天还有一个很现实的动作：把试点期的“临时做法”改成“正式规则”。

试点期里怎么干都行——某个骨干盯复核，某个项目经理盯日志，某个业务负责人盯异常，某个技术同学盯系统。但到了 180 天，这些动作要是还得靠人盯着，就说明制度化根本没完成。

正式规则至少写清楚五件事。谁定期看 AI 的输出质量。谁维护知识库。谁批准权限变化。谁处理异常升级。谁决定这条流程能不能复制到第二个团队。

第二条那个「谁」要是空着，知识库照样会建起来，也照样会旧下去——建成那天不是它开始起作用的那天，是它开始变旧的那天，除非有人被指定去养它。

这五条听起来像管理细节，实际决定 AI 能不能从试点走进日常。

很多公司试点成功、规模化失败，就是因为没有把这些临时动作固化。试点时大家都在场，上线后大家都回到自己的岗位；系统还在，责任散了。这就是制度化失败。

制度化失败的账不是不结，是延后到一个更贵的时点一次性结。前面那家企业的知识库就是这么荒的：第一遍是项目预算里的活，第二遍是被一号位追着补的。而且要靠一号位下命令才能让知识库重新更新，说明缺的从来不是执行力，是机制——命令能解决这一次，解决不了下一次。

第一年的指标：第二条流程有没有更快

第一年还要看一个信号：第二条流程有没有跑得更快。如果第一条流程花了 90 天，第二条流程还是花 90 天，先别急着判组织没学会。两条流程的难度、数据准备、风险和验收标准相近吗？如果可比，再追问哪些经验被复用了；如果第二条流程用了不到一半的时间，也要确认不是少做了复核、放低了标准。

这个指标比“AI 覆盖多少流程”更重要。覆盖数量容易做，复制速度难做。复制速度背后，是组织有没有积累方法：选切口的方法、三张账的模板、复核责任表、知识库治理方式、异常升级机制，有没有被复用？

如果这些东西都要重新讨论，说明组织没有学习；如果可以拿来就用，再按场景微调，说明组织能力开始长出来。AI 转型真正的规模化，不是同时开十个项目，是第十个项目比第一个项目更容易跑通。

180 天那三件事管的是制度有没有落地，第一年这两件事管的是能力有没有长出来——它们不是两套竞争标准，是同一条路上的两个刻度。所以第一年结束的真正里程碑，不是 AI 覆盖了多少人、多少流程，是另外两件。

第一，第一条流程攒下的判断标准、复核规则、典型错误案例，有没有整理成能检索的知识资产，让第二条流程上线时直接引用。第二，从第一条流程到第二条流程，从启动到跑通的时间有没有明显缩短。

时间缩短，加上质量守住、经验确实被复用，才是组织学会了的证据。没缩短，则要查是新场景更难，还是旧经验没有进入新流程。别把一张工期表当成学习能力的判决书。

一个人处理过某类问题，这叫个人记忆。要变成组织记忆，回到第九章检查：经验是否经过核验、按权限进入流程，规则更新后是否真的被下一次调用。存成文档，只是其中一步。

AI 进入流程以后，组织记忆建设的需求不减反增。通用模型可以升级，但它不会替企业保存每一次取舍的来龙去脉。知识没有积累，下一条流程就可能重新付一遍解释和纠错的成本。

复制的是能力，不是工具

这个指标听起来抽象，落到具体项目上其实很好认。举两个形态不同的例子。它们不是同一个项目的两个阶段，是两种不同的复制方向。

第一个是最常见的横向复制。给一家企业做 A 部门的知识库治理，口径、分类、更新责任、权限规则一条条梳理清楚，中间少不了争执——分类到底按业务线还是按文档类型，过期内容谁有权删，检索不到的时候走人工还是走上级。这些问题在 A 部门都吵完了。等 B 部门提出同样的需求，真正需要重做的只有一件事：权限要隔离，两个部门的内容不能互相看见。

除此之外，分类怎么定、谁负责更新、过期内容怎么处理、检索不到的时候走什么流程——B 部门迁移过去，再按自己的业务做一些改善就可以用。

这就是第二条流程更快的真实样子。它不是因为工具更熟练了，是因为那些需要讨论和拍板的东西，已经有了一版可以拿来改的答案。第一条流程最贵的成本不在系统，在争执；争执一旦有了结论，结论就是资产。

第二个例子更有意思，因为它复制的方向是向外的。

我做过一个制造企业的维修知识 RAG。最初的需求很内部：维修人员到客户现场服务，需要快速拿到答案。难点不在技术，在知识体量——整套装备很大，涉及大量配件说明书，还有图片和视频讲解。

人的记忆是有上限的。型号一多，即便按不同机型分配专门的服务人员，前线维修人员还是经常要呼叫总部支援。这件事的本质不是维修人员不专业，是没有人能把一整套多型号装备的处理经验都装进脑子里。RAG 不是让人变成百科全书，是让人在需要的时刻拿到需要的答案。

所以第一步做的是把内部维修知识拉通，让维修人员在前线能直接检索，不用每次都往回问。

这一步做好以后，一个新的可能性自己冒出来了。系统既然已经能回答“这个型号的这个部件出现这种症状该怎么处理”，那里头总有一部分问题足够简单、足够标准。这些问题，为什么不直接开放给客户自己查？

于是同一套能力，从内部支持外溢成了客户侧的自助排查助手。

这两个例子说的是同一件事。第一个流程跑通以后，真正有价值的产物不是“这条流程更快了”，是组织多了一块可以搬走的能力。

所以第一年该盘点的不是上了几个 AI 场景，是有几块能力已经具备了被搬走的条件。就问三句：它的口径写清楚了吗？它的责任人定下来了吗？换一个部门用，改的是配置，还是得从头再吵一遍？

复制一个工具很容易，买一份授权就行。复制一套能力才是组织在长本事。

CEO 必须说清产能分配

还有一件事，CEO 不能躲：产能释放以后，到底怎么分配？这不是 CHO 一个部门能决定的，也不是业务负责人自己能决定的，因为它会影响组织信任。

员工要是觉得 AI 一提效、组织立刻就裁人，他不会把真实的业务上下文交出来。他要是觉得自己把经验喂给组织的 AI，换来的只是自己更容易被替换掉，他就会开始留一手——留判断，留经验，留异常，留真实反馈。

员工一旦开始留，AI 就只能学到表层。

表层很快，也很脆。

所以产能分配不是财务问题，是组织契约问题。CEO 必须说清楚哪些场景为了降本，哪些为了增产，哪些为了提质，哪些为了重组。说清楚不代表永远不调整，是让员工知道组织要往哪里去。

AI 转型最怕偷偷摸摸。越偷偷摸摸，员工越会把它理解成组织偷袭。

四类推进角色必须到位

这条路线的治理层里，四类推进角色必须到位。它不是第十九章六类责任图的删减版：业务 owner 仍对具体流程结果负责，治理负责人仍负责规则、权限、审计和例外；下面四类只负责把 90 天试点推进起来。

CEO 负责三个决策节点：第一个切口是哪条流程；第 90 天复盘时选择停、调整、扩大还是复制；产能分配方向是降本、增产、提质还是重组。这三个决定不能下放给 IT 或供应商。

CHO 负责三个制度接口：岗位职责更新、绩效口径调整、AI 影响流程的动态清单。CHO 如果只做培训，不做制度接口，员工会在新工作方式里没有规则、没有保障、没有方向。

CIO 负责三个技术接口：运维接得住、数据治理规则清楚、权限地图持续维护。CIO 如果只关注系统上线，不关注这些接口，风险会随着 AI 系统增加而积累。

AI 转型负责人负责推进和连接：推进三账测量，推进试点启动和复盘，推进 CEO、CHO、CIO 在正确时间做出正确决定。他不是协调员，是组织翻译和闭环负责人。

老板真正需要的路线图

前面这几套东西——三张账、四类活、第 90 天的四个结论、180 天的三件制度化、第一年的两个刻度——不需要读者背下来。把它们摊平在一张表上，老板每个阶段该问什么、该留下什么、什么信号说明跑偏了，一眼能看完。老板真正需要的，不是一张“AI 项目进度表”，是这样一张能在会上拍板的表。

阶段	老板要问什么	必须留下什么	不合格信号
0-30 天	哪条流程值得改？谁愿意签名负责？	三张账、流程 owner、现状基线	只有工具清单，没有流程账
31-90 天	AI 是否进入真实流程？	人机分工、复核节点、验收指标	只有 Demo，没有业务依赖
第 90 天	停、调、扩、复制，选哪一个？	决策会纪要、责任链更新、下一步资源	只开总结会，不做取舍
91-180 天	现有安排是否接得住？	岗位、流程、知识、权限的核验及必要更新记录	已发现失配，仍没有处理安排
第 1 年	能不能复制到第二条流程？	第二、第三条流程迁移证据	每条流程都从零开始

日期是建议节奏，不是自动升级许可。未取得相应证据，就调整、暂缓或停止；原制度已经适用，不必为交差而改制度。这张表的价值不在于显得管理规范，在于它逼着老板每个阶段都问结果。没有结果，就别接着讲愿景。AI 转型最怕的从来不是慢，是忙了半年，最后只留下一份项目进度。

这周就可以做一件事：找出公司里那条已经有人在用 AI、却没有明确复核责任人的流程，约上它的业务负责人谈一次。

谈的东西就五样：谁在复核，怎么记录，出了问题找谁，哪张账会变，90 天后拿什么验收。

这不是启动新项目，也不用申请新预算，只是把一条已经在跑的流程的责任归属，从口头挪到文件上。谈完这一次，你才知道自己的组织现在站在路线图的哪一格。

要是 90 天后你还说不出哪条流程缩短了、谁在复核、哪张账变了，这不是 AI 的问题，是你没给 AI 一个真实的组织切口。

AI 转型不看路线图写得漂不漂亮，看组织有没有真的学会接住它。没有真实流程，AI 没有落点。没有责任人，流程没有主人。没有流程损耗三账，试点没法验收。没有复盘和制度化，90 天就只是一个项目周期。

路线图真正的价值，是让组织在每一个阶段都清楚自己要留下什么能力。没有能力沉淀，所有进度都是幻觉。老板要看的从来不是热闹，是能力。

CHAPTER 22

谁把 AI 带进业务现场

我见过很多 AI 项目会，最让我警惕的不是大家吵架。是每个人都说得对。

CIO 说系统能不能上线，数据怎么接，权限怎么开，安全怎么管。CHO 说岗位会不会变，员工会不会抵触，培训怎么做，绩效怎么算。业务负责人说客户要什么，流程哪里慢，一线到底愿不愿意用。老板坐在中间，听起来每个人都有道理。

但会开到最后，常常没有一个人能把三句话翻译到一起：系统能不能跑，业务要不要用，组织能不能接住。

这就是 AI 进入企业以后最容易出现的断点：技术有，业务痛点和组织意识也有，唯独缺一个能下到现场，把 AI 能力、业务流程、知识治理、责任边界翻译成同一张图的人。

没有这个人，AI 项目就会被拆成三段：技术团队做系统，业务团队提需求，HR 和组织团队做培训、沟通、绩效调整。每一段都在努力，三段之间却没有真正接上。最后系统可能上线，业务可能试用，培训也可能完成，组织能力却没有长出来。

这就是本章要讲的业务现场角色。它不是一个岗位热词，是 AI 进入业务现场后，组织必须长出来的一种翻译能力。

不是岗位名，而是组织功能

判断 AI 有没有真的进入组织，别先看裁员公告，也别先看买了多少账号。看一个更硬的信号：组织里有没有冒出一种以前不存在的能力。这种能力专门负责把 AI 接进真实业务流程，并且对这条流程跑出来的业务结果负责。

其中有一种现场工程角色，叫 FDE，Forward Deployed Engineer，前线部署工程师。这个词不是这本书发明的，Palantir 很早就在使用；OpenAI、Anthropic 这些 AI 公司现在也把它写进岗位系统里。岗位名称说明不了全部业务问责，仍要看它具体承接了什么任务、得到了什么授权。

OpenAI 的 FDE 岗位说明很直接：成功不是做出一个漂亮演示，是三件事。一是“生产采用”，客户的生产系统真的用起来了。二是“可量化的流程影响”，这条业务流程的效果变化能被量出来。三是“评测驱动的反馈”，靠一套评测数据回头改，不靠拍脑袋[①]。

Anthropic 的 FDE 岗位也写得很清楚：要交付能进入生产工作流的 MCP servers、sub-agents、agent skills。说人话就是三样东西：能直接插进公司现有系统的接口，能自己跑完一段活的 AI 助手，可以被反复调用的 AI 技能包。除此之外，还要把可重复的部署模式沉淀回产品和工程团队[②]。

这些公开岗位说明把一份工作的内容说清了：进现场、改流程、进入生产环境、把部署模式沉淀成能重复用的东西。它们不能证明这些公司过去“只卖模型”，却能说明企业落地要有人专门负责。

但不能把 FDE 直接等同于乙方。“前线”说的是离真实业务问题有多近，不是谁给他发工资。企业自己的工程团队也可以走进内部业务现场，把需求一路做成可运行、可验收的系统；供应商派来的人，也不能只管代码，不管采用和交接。甲乙方说的是合作关系，工程与组织说的是工作内容，不是两套一一对应的岗位。

本书关心的是其中更进一步的组织任务：把业务现场、知识治理、流程责任和 AI 能力接成企业能持续使用的能力。内部 FDE 可以承担它，外部 FDE 可以参与，双方也可以组队完成。懂业务和组织、但不承担工程交付的人，可以作为组织侧现场负责人与 FDE 搭档，不必为了一个新名字，把不同职责硬塞给同一个人。

没有这个人，老板会把 RAG 想成整理资料，CIO 会把 AI 想成系统上线，CHO 会把 AI 想成培训，业务部门会把 AI 想成外包给技术团队的项目。每个人都在做自己熟悉的那一段，最后却没有人对“AI 有没有变成组织能力”负责。

所以真正的判断不是“FDE 是个新岗位”，那太浅。当 AI 从工具变成劳动力，组织里第一个必须长出来的器官，是能把 AI 带进真实业务现场，并把现场经验沉淀成可复制组织资产的人。

---

①OpenAI，《Forward Deployed Engineer》招聘岗位说明书，openai.com/careers，2026 年。

②Anthropic，《Forward Deployed Engineer, Applied AI》招聘岗位说明书，Anthropic 官方招聘页，2026 年。

你可以设置内部 FDE，也可以由业务现场负责人与工程伙伴共同承担这项任务。名字没那么重要，重要的是必须有人牵头，把四件事回答清楚：业务现场到底哪里值得改？AI 进入流程后，人机怎么分工？结果怎么验收？出事以后谁负责、怎么复盘、怎么复制？

四个问题没人管，AI 就还没进组织。

交付组织资产，而不是一次 Demo

这种业务现场角色干完一个项目，留下来的如果只是一个能跑的 Demo，完成的就还只是演示。不论是内部团队还是外部伙伴，都要留下别的部门能照着复制的方法，才算在帮组织长能力。

最常见的误解，是把 FDE 理解成帮某个业务部门做一个 AI 方案。这个理解差得很远。方案做完、Demo 跑通、PPT 汇报完，人一撤，东西就死在那里；半年后没人维护、规则不更新、知识不沉淀，它就会退化成一个僵死的自动化脚本。

这个角色的本质，不是做一锤子买卖，是把业务现场反复出现的模式，抽象成可泛化、可运行、可验收、可复制的组织能力。说得再直白一点，它是企业自己的 AI 落地发现机制，输出的是组织资产，不是一次性 Demo。

什么叫组织资产？至少要留下五件东西。

第一，一张真实流程损耗图：这条流程哪里慢、哪里贵、哪里错不起、哪里只是看起来能自动化。画图不能只听一次需求汇报。把老板最初的交代、制度文件、业务记录和现场交接放到桌上，沿一笔真实业务核对：当初说要怎么走，规定该怎么走，实际怎么走，中间的差异由谁解释。损耗图要从这里画出来。发现某一步本就不该保留，也要允许答案是删掉它，而不是给它加上 AI。

第二，一张人机分工和责任表：哪些步骤 AI 做，哪些步骤人判断，哪些节点必须复核，出了错先找谁。

第三，一个可运行的 AI 工作流：不是沙盘 Demo，是接进数据、权限、日志、异常处理之后，能被业务日常使用的工作流。

第四，一组验收指标：时间少了多少，等待压掉多少，返工减少多少，结果一致性提高多少，组织知识沉淀了什么。

第五，一份可复制 playbook：这次怎么找场景、怎么拆工作、怎么设边界、哪里失败过，下一条流程怎么少走弯路。

少了这五件，项目再热闹也只是热闹；五件齐了，才叫组织资产。

这套标准，我自己也栽过。

那次要评估的是我自己那个投放系统。我想知道里面的 AI 到底自主到了什么程度。这个问题听起来像技术问题，其实是责任问题。系统自主到哪一档，责任就得配到哪一档。所以它不是可以拍脑袋的事，我也没有拍。

我让人穿透读了五条业务流的实际代码，一条一条核对了七个关键信号。每一个判断背后都有实打实的证据。不是产品话术，是代码事实。这套举证方式，拿到任何一场评审会上都站得住。

结论出来了：这个系统已经是半自主偏全自主。AI 不只是在提建议，是在真的代替人做事。

里面有两条最刺眼。第一条，代码里根本找不到”等人审核”这个中间状态。也就是说，整条流程上没有一个地方是停下来等人点头的。第二条，AI 出错的时候会自动降级放行，不等人来兜底。出错本该是最需要人接住的时刻，系统却在那个时刻自己往前走。

这两条如果成立，那就不是自动化程度高不高的问题，是责任链有没有断的问题。

我把它们写进了文档。文档发给了客户，我还开会跟他们讲过一遍。那天我讲得很稳。每一条都能指到具体位置，对方问哪一句我都接得住。现在回想，我那份底气全部来自证据的密度。我从来没想过要问的是另一个问题：这些证据，有没有人再核过一遍。

一个月后，我让另一批人重新拷问这个产品。五个独立的分析角色，各用各的框架，从各自的角度切进去。产出交回来，其中大多数几乎都在复述我那份结论。

如果到这里为止，这件事就结案了。

我当时也是这么想的。

八步里，六步是组织设计

这里有一个很容易被工程师出身的人跳过的真相：这件事真正难的部分不全在工程。

如果把一个 AI 现场项目拆成八步，大概是：找场景，拆工作，设计人机协同，搭 AI 工作流，接系统，推使用，定义验收，沉淀方法论。真正偏纯技术的，是搭工作流和接系统；其他六步，核心都是组织设计。

找对场景，是业务判断。拆清损耗，是流程判断。分清人机权责，是责任设计。推动业务真的用起来，是变革管理。定出可验收指标，是经营管理。把经验沉淀成可复用资产，是知识治理。

这就是为什么技术再强的人，如果只盯着“系统能不能跑通”，做出来的东西大概率还是 Demo。他在第四步、第五步上是高手，但真正决定组织能不能接住 AI 的，是另外六步。

能把 AI 落成组织资产的人，赢在那六步上，不只赢在代码上。

技术部署与组织承接，要从开始就一起做

技术部署和组织承接是两类工作，不能排成“系统先上线，组织以后再接”的两棒。数据权限、流程改造、验收口径和异常处理，从试点开始就得一起定。它们可以分给不同的人，也可以由同一个有能力、有授权的人承担，不能按甲乙方身份直接切开。

IT 说我只管系统有没有跑通，业务说我不懂怎么把 AI 接进流程，HR 说岗位和绩效得改、却不知道按谁的结果算。三句话都合理，合在一起却是一处责任真空。

很多企业失败，不是第一棒没人跑。是第二棒没人接。

现场角色要组织这些人把接口接上，不是替每个人接下所有责任。

我愿意再往前推一步：承担这类任务的 FDE，可以是企业内部业务责任单元的孵化者。这里说的孵化，不是又设一个部门，而是围绕一段值得持续交付的业务结果，把具名负责人、人机工作流、上下文、权限和验收办法一起做出来。它可以从已有业务中长出来，也可以服务于一条新业务。这是本书对现场角色的一种设计，不是所有 FDE 岗位的通用定义。

孵化是否完成，要看接手的人能不能在约定的技术支持下，持续拿回结果、看清投入产出、处理异常、更新规则。不是看孵化者做了多少个 Demo。甲方 FDE 可以承担这项任务，乙方 FDE 也可以参与，联合团队同样可以做；但从试点第一天起，企业就要给业务结果指定有权负责的人，不能等孵化完成才找。

孵化者和长期负责人可以是同一个人，也可以不是。前者留任，要重新确认运营职责；前者转去下一条业务，要完成交接。技术维护可以继续由供应商承担，企业的经营判断和最终问责不能因此空着。试点做不出值得持续投入的结果，也应该允许停止，不能为了交一份“孵化成功”的答卷，硬造一个单元。

它和现有角色有什么不同

组织侧现场角色不是把旧岗位换个新名字。它和传统顾问、项目经理、BA、技术部署角色的区别，必须说清楚。

角色	主要交付	常见盲区	组织侧现场角色要补什么
传统顾问	诊断、方案、汇报材料	不一定进入生产系统	把判断落进真实流程和责任链
项目经理	排期、协调、交付推进	不一定重写组织责任	钉住流程 owner、验收指标和复盘机制
BA / 业务分析	需求翻译、流程梳理	不一定懂 AI 边界和组织变革	把 AI 能力、人机分工和组织语境接起来
技术部署角色	系统、集成、部署、生产可用	不一定处理岗位、绩效、信任	从试点开始共同设计，把系统接成组织能力

这张表不是为了贬低任何角色。恰恰相反，AI 转型需要这些角色；但如果没人负责把它们合成一个组织闭环，项目就会停在“各做各的”。组织侧现场角色的价值，就是让这些分散的努力合成一条能跑的组织链路。

三种翻译

这个角色不是一个“更懂 AI 的项目经理”，它真正做的是三种翻译。

第一种翻译，是把业务痛点翻译成 AI 可以处理的工作对象。很多老板说“我们要做知识库”，这句话太粗。到底是客服找不到历史订单，维修人员找不到设备说明书，销售不知道客户过去买过什么，还是总经理办公室每天要追一堆宏观数据？到底是候选人筛选慢，还是面试官判断标准不一致？

这些问题不拆开，AI 就只能被做成一个大而空的系统。现场角色要把“我要 AI 转型”翻译成“这条流程里的这个节点值得被重写”。

第二种翻译，是把 AI 能力翻译成组织可以接受的人机分工。AI 能做什么，不等于组织应该让 AI 做什么。AI 可以生成客服话术，但对客户形成承诺，谁复核？AI 可以筛简历，但文化匹配、信任潜力、组织风险，谁判断？AI 可以写方案，但方案对外发出以后，谁对客户负责？

不能只问模型能不能做，还要问：这个动作如果真的发生，组织里谁接得住？

第三种翻译，是把一次项目经验翻译成下一次可复用的组织语境。很多 AI 项目做完以后，经验留在项目群里、几个骨干脑子里、一堆没人再打开的文档里。这不叫组织能力。

真正的组织能力，是下一条流程启动时，前一次的场景选择、边界判断、失败模式、验收指标和权限设计可以被拿出来复用。

所以这个角色不该只交付系统，还要交付业务上下文。谁知道这条流程当初为什么这样设计？哪些节点不能自动化？哪类异常必须回到人工？这次试点犯过什么错，下次不要再犯？

把三种翻译并排看，交付物就不该只剩一张系统清单。

现场原话或问题	要形成的工作产物	回来核什么
“我们要做知识库”	真实流程节点、输入输出与价值问题	是否解决原来的痛点，遗漏了什么约束
“AI 已经能做”	人机分工、授权、控制与后果承接安排	接手者是否有权、有能力运行和处理例外
“上一个项目跑通了”	带适用边界的流程方法、失败记录与验收依据	下一流程实际使用后，哪些方法仍适用、哪些应修正

这三组对应不要求都由一个人完成，也不由 FDE 这个名称独占。经验没接到下一次工作里，系统再多，组织也可能没有留下记忆。

那批报告方向一致，看起来足够扎实。其中一个负责独立审查的角色，却问了一句不在议程上的话。

他没有去挑那些报告里的任何一句。他问的是它们脚下：这批报告，是不是都站在同一块地基上？它们的结论全都来自我那份代码穿透。而那块地基，从来没有人回去核过。

于是我们回去读了真代码。

结论反了。

先看第一条。我当时写的是：代码里找不到”等人审核”这个中间状态，整条流程没有一个地方停下来等人。

真代码里的情况正好相反。那个中间状态不但存在，而且是 AI 走完自己那一段之后，唯一能到的地方。AI 把内容审完，判断没问题，它能做的动作只有一个：把这件事推进”审核中”。推完它就没有下一步了。真正把一件事从”审核中”推到通过的，是另一个动作，而那个动作 AI 这条路径从来不触发。不是很少触发，是一次都没有。它压根不在 AI 能走的那条路上。

所以我当时说的”找不到等人审核的状态”，恰好说反了。那个状态不但在，它还是 AI 这条路的终点站。我说这条流程一路无阻，实际上它每一次都停在同一个地方，等人点。我把一堵墙，看成了一条通道。

再看第二条。我当时写的是：AI 出错会自动降级放行，不等人来兜底。这一条我没有全错，但错在了最关键的那半句上。

系统在失败的时候确实会降级，这半句是对的。错的是我以为降级等于放行。真代码里，降级之后系统做的第一件事不是往前走，是先给这件事打上一个标记，说明它是降过级的。紧接着第二件事，是把”已转人工”这句话写死在上面。做完这两步，系统才把它送走，而送到的地方，和正常流程的终点是同一个：审核中。

所以我当时写的”自动放行”，和它实际做的”标记、转人工、停在审核中”，差的不是一个环节的松紧，是方向。我写的是一条绕开人的路，它走的是一条把事情交回给人的路。同一个降级动作，在我的文档里是责任的出口，在真代码里是责任的入口。

写死”已转人工”，就是把这件事交回给人。我当时读到它，却没有把它当成一次交接。

两条并排放在一起，性质就很清楚了。我不是在细节上偏了一点，我是把这个系统的责任结构整个看反了。

真相是：这个系统比我写的更安全，也更成熟。

这里要说清一件事，免得读成一个”下属给了我错报告”的故事。那不是实情。那份穿透不是编的，读的是真代码，核的是真信号。它错在读法上，不是错在有没有认真读。所以真正的病因不是谁失职，是我把一份没人回头核过的判断，当成了一件已经查实的事实。

错在有利的方向上，也是错。如果只在结论对自己不利的时候才去复验，那复验就不是求真，是防守。

这个反转是往好的方向反的。我不是发现自己吹了牛，我是发现自己冤枉了自己的系统。正因为它对我有利，它才更容易被留在原地不管——反正没人会来追究一个把自家系统说得太严的判断。

我在复盘里写下的那句话是：旧文档是缓存，不是真相。

那份错的文档已经发出去了。所以这件事还有后半段。我又开了一次会，把新的口径讲清楚，说服客户改过来。

认错的成本不在承认那一刻，在于你要回去把已经说服过的人，重新说服一遍。 前者只需要诚实，后者要付出信用。很多组织的复盘停在内部承认，对外的口径继续跑着。

回过头看，救场的不是更高级的分析框架，也不是更多的分析角色。是有人肯回去把那份旧文档再核一遍，问它到底还算不算数。

从工程出发，也可以从业务出发

什么人能做 FDE？我不愿意先按原来的职业划线。工程出身的人可以补足业务判断，业务出身的人也可以补足工程能力。两条路都走得通，但有一层共同基础：组织视角。

组织视角不是第三项加分。它决定你把什么当成问题。看到一个审批节点，不能只问接口怎么接，也要问这一步为什么存在、谁有权改；看到一条流程提速，要继续看上下游接不接得住、员工为什么愿意配合、出了例外谁判断、经验能不能留下。没有这一层，业务和工程即使都懂，也可能只把旧流程做得更快，组织照旧。

工程出身的人，优势是能把系统做出来，补的是业务价值和真实约束：什么结果值得投入，哪个环节真在拖累业务，做完以后谁用、怎么验。业务出身的人，优势是知道现场哪里不对，补的是把判断做成系统的能力：能拆任务和数据，借助 AI 与专业伙伴做出原型，设计评测、识别失败，知道哪些地方必须请工程专家接手。起点不同，最后都得拿出可运行、可验收的东西。

我看好业务人员走这条路，是因为业务里最值钱的东西，很多不在流程图和需求文档里。哪些客户不能这样回复，哪些流程只是表面流程，哪个字段其实不可信，哪个判断只能问老员工，哪个异常必须升级——这些都没写下来。

它还藏在更具体的判断里。老员工知道哪个客户每次都要特殊处理。HRBP 知道一个候选人的“参与过”到底是主导还是打杂。维修工程师知道某个型号出了问题，说明书上没有，但现场十有八九是那个零件。投放团队知道一个达人数据好看，合作起来却未必不掉链子。

这些经验是起点，不是资格证。业务人员不能只提需求，把交付全部推给别人；工程人员也不能只证明系统跑通，让业务自己想办法采用。FDE 不必亲

手写完每一行代码，更不必独自精通安全、数据和所有行业知识，但要能组织专业力量，判断交付是否可信，对自己承接的那段工程交付负责。

所以我会拿同一条真实流程来考两种候选人：他能不能说清业务价值，做出并验清一个可运行的工作流，再把谁能改、谁来用、谁接住结果写进实际安排。三件事要放在一起看，组织视角贯穿其中，不能留到系统上线以后再补一堂课。

这个角色不是把业务交给 AI。是把业务里那些过去说不清、写不下、传不远的判断，拆成 AI 能参与、组织能复盘、人类还能负责的工作单元。这不是“会不会用工具”，是组织学习能力。

先写产出，再写岗位说明书

一个岗位如果写不出核心产出，就还不是一个岗位，只是一句口号。

招不到人，考不了核，也背不了锅。

所以这种职责要进入普通企业，就得写进明确的任务约定，但不一定新增编制。先看任务会持续多久，需要哪些专业：一条边界清楚的试点，可以让现岗人员兼任；需要多个专业，可以临时组队；持续有多条业务要孵化，再考虑长期专岗。不论哪种安排，都不能只写“负责推动 AI 落地”。这句话听起来什么都管，实际上什么都管不住。

任务约定的第一栏不要先写职责，先写产出：牵头者带着团队干完 90 天，桌上至少应该留下五样组织资产。决定设岗以后，再把这份约定写进岗位说明书。

这五样写不出来，就别急着招这种人。产出写不出来，说明你还没想清楚要他干什么；这时招进来的，大概率又是一个开会的组织者和汇报的中转站。

从组织内部培养种子

FDE 确实正在变热，但对中国企业的真正启示，不是去硅谷价码上抢人，是从自己的组织里，把懂业务的人养成组织侧现场角色。

原因很简单，这种人本来就稀缺。“懂 AI 工程 + 懂业务现场 + 懂组织运行”这个交集，不是招聘网站上随便一搜就有。

从外面高价挖来的技术高手，可能很懂系统。但他不懂这条业务的灰度，不懂客户为什么偏偏在这个节点犹豫。他也不懂老员工脑子里那些没写下来的判断，不懂公司哪些权限看起来合理、实际却会把流程卡死。这些东西，外人短期学不会。

组织内部那个最懂业务、又愿意拥抱 AI 的人，反而可能是更好的种子。给他补 AI 素养，给他流程改造方法，给他组织授权，再给他一个可以并行试错的场景，这比一上来去外面抢一个“最火岗位”更现实。

当然，这不是说所有公司都只能内部培养。内部可以长出懂业务的工程师，也可以长出补足工程能力的业务骨干；外部也可以找到这两类人。选谁，要看他是否有组织视角，能否在现场把业务判断和工程交付接起来，不按原来的部门或专业直接判输赢。

内部培养也不要把赌注只压在一种出身上。AI 产品经理擅长拆问题，解决方案架构师懂交付，工程师有技术深度，业务型 HRBP 更可能看见岗位、绩效和非正式协作关系。四类人没有谁天然胜出。真正该比较的是：他能不能把一条真实流程拆到任务和判断，能不能说出谁会因改变失去什么，能不能在系统上线后继续对结果负责。对组织侧现场角色而言，这三件事比头衔更重要。

所以培养不能从“报什么技术课”开始，要从一条真实流程开始。一个可用的路径是：先用约两周，把 AI 看成新的分工方式。接着用约三十天做工作分析和流程拆解，再用约三十天设计人机协同和责任边界。随后用约六十天学会把流程搭起来。最后至少留约四十六天，在真实业务中完成一次从设计、上线到复盘的闭环。五段顺序推进合计约 180 天；实际执行可以局部重叠，所以 180 天是组织排期口径，不是死线。这是一条示例路径，具体训练要按起点补短板：工程背景补业务判断，业务背景补工程交付，两边都把组织视角放进真实任务里检验。

我从 HR 进入 AI 系统工作后，真正改变认知的也不是一次模型部署，是把一个组织判断拆成可复核的工作单元。以简历初筛为例，“参与过项目”不能直接等于能独立交付；需要追问的证据、否决条件和复核点，可以写成规则，让系统先整理，人保留最终判断。它让我看到，组织经验并非天生不能进入系统，但必须先变成有边界、有证据、可复核的结构。这段经历只说明这套方法从哪里来，不替任何结论担保。

先授权，不要先发招聘广告

管理者读完这一章，不要马上去发招聘广告，先问三个问题。

第一，公司现在有没有一条真实流程，已经值得被 AI 重写？第二，这条流程里，谁最懂业务、最懂例外，也最愿意下场试 AI？第三，如果给这个人 90 天，他能不能留下五样组织资产？

三个问题都答不上来，FDE 这三个字对你来说还太早。答得上来，下一步也不是讨论头衔。先写清谁能改流程、谁能调资源、谁认验收、异常由谁裁决，再把试点交出去。不能让一个既调不动资源、又改不了规则的人，对整条流程的结果作保证。不论他是内部员工还是外部伙伴，关键授权拿不到，就缩小范围、升级裁决或者暂不接单，不能让“对结果负责”变成无限兜底。

还要分清两笔责任：FDE 对约定的部署、试点和交接负责，业务负责人对约定范围内的业务结果负责。若由同一个人承担，也要把两笔分别写清。团队可以共同做，责任不能只留一个团队名字。

项目经理负责把项目推完，这个角色负责让组织学会下一次怎么自己推。项目经理交付进度，它交付能力。两者看起来很像，实际差得很远。

一个 AI 项目结束以后，如果只剩会议纪要、上线截图和汇报 PPT，那只是项目结束了。如果组织多了一套能复制到下一条流程的方法，才是能力开始长出来。

那五样资产，其实咬着前面每一章

写到这里，可以回头看一眼那五样组织资产。它们看起来是这一章给出的一张清单，实际上，每一样都在向前咬住这本书前面讲过的一件事。这不是巧合，是因为它们本来就是同一个问题在不同环节上的样子。

第一样，真实流程损耗图。它咬的是这本书一开始那个判断：AI 替代的不是岗位，是任务、判断和流程片段。画损耗图，就是把一条流程重新拆回片段，看清哪一段是动作、哪一段是判断、哪一段只是看起来能自动化。拆不到片段这个颗粒度，后面所有事都会踩空——岗位重写会变成在 JD 后面加一句“熟悉 AI”，产能分配会变成在裁员和观望之间反复横跳。

第二样，人机分工和责任表。它咬的是责任那几章。哪些步骤 AI 做、哪些步骤人判断、哪个节点必须复核、出了错先找谁——这不是项目管理的排班

表。这是把风险、可逆性、判断含量和责任外溢四件事，在一条真实流程上重新走一遍，然后给每个节点选一档责任机制。做不到这一步，人在回路中就还是那个统一按钮。

第三样，接进真实数据和权限的 AI 工作流。它咬的是治理那一层。工作流一旦接进真实数据，权限、审计、留痕、回滚和人工兜底就不再是纸面条款，是每天都在被使用的东西。沙盘 Demo 之所以不算数，不是因为它做得不好，是因为它从来没有被这些东西检验过。

第四样，验收指标。它咬的是绩效口径那一章。时间少了多少、返工减少多少、结果一致性提高多少——这些数字真正的作用不是汇报好看，是把这条流程接进经营账。接不进账的改善，组织没有办法判断该不该继续投入，也没有办法回答那个最要命的问题：AI 释放出来的产能，到底流到哪里去了。

第五样，可复制 playbook。它咬的是知识和信任那两章。这次怎么找场景、怎么设边界、哪里失败过，下一个人能不能拿出来用？这取决于两件事：组织有没有地方存这些判断，员工愿不愿意把真实的判断交出来。没有存放的地方，经验会留在项目群和几个骨干的脑子里。没有信任，交出来的会是一份漂亮但没用的复盘。

所以这五样不是五个交付物，是一条链。缺了第一样，后面四样没有对象。缺了第二样，工作流会跑，但没人对结果负责。缺了第三样，前两样停在纸上。缺了第四样，做完了也证明不了。缺了第五样，这一次做对了，下一次还得从头再来一遍。

这也是为什么这个角色不能只是一个更懂 AI 的项目经理。项目经理对这一条流程的进度负责，这个角色要对这条链能不能在下一条流程上再走一遍负责。前者交付一次结果，后者交付组织下一次自己走的能力。

所以读到这里，先别开会，也别写招聘启事。拿你手上跑得最久的那个 AI 项目，对着这五样东西一样一样点：流程损耗图有没有？人机责任表有没有？接进真实数据的工作流有没有？进得了经营账的指标有没有？下一条流程能照着走的 playbook 有没有？

五样里点不出三样，那个项目就还没在长组织能力，它只是在跑。

这就是这个角色的价值，也是它的边界：让他证明 AI 在你的公司不只是工具，是正在长出来的组织能力。

PART FINAL · 终章

组织长什么样

前面七部都在修东西。修断掉的岗位，修跑不通的流程，修散在个人电脑里的知识，修说不清的责任，修跟不上的治理，最后用 90 天试点把这些修补动作变成能复制的做法。

修到这里，一个问题躲不掉了：这些都做完以后，这家公司到底长成什么形状。

它不是把前面几部再总结一遍。序言里埋的那个更大的问题，要在这里给答案——AI 把干活这件事变得越来越便宜以后，组织凭什么还存在，以及一家真正接住了 AI 的公司，组织图跟今天的比到底哪里不一样。

组织最后长成什么样?

CHAPTER 23

AI 上完、五断点修完之后，组织长什么样

上一章收的是资产，这一章收的是形状

上一章讲完那五样组织资产之后，有一句话是这么说的：**每一样都在向前咬住这本书前面讲过的一件事。**

这句话是对的，但它只收了一半。

它收的是资产：流程损耗图、人机分工和责任表、接进真实数据的工作流、验收指标、可复制 playbook。这五样让一条流程具备了可检查、可维护的基础。事情有没有做成，还要看约定结果和持续投入；下一条流程能否照着走，要另验接手条件与场景差异。

还有一半要检查：工作变了，原来的组织安排是否还接得住？

把组织图拿出来，先别急着重画。看任务变化以后，原来的授权、协作、资源和结果承接还能不能覆盖。如果能，旧框也可以继续用；如果不能，先改那个失配的接口。图上有没有新格子，不是改造成功的证据。

这一章提出一种候选形状，叫责任单元。它必须和已有部门、项目制、流程负责人安排放在一起比较，而不是先宣布旧图作废。

比较之前，先看看现有组织图表达了什么，又没有表达什么。

出了事，第一眼能不能看出该找谁

先从一个更老的问题问起：岗位为什么存在？

把一个岗位一层一层剥掉。名称可以改，运营专员改叫增长运营，昨天的事今天照做。人可以换，老王走了小李接。每天干的活可以变，今年做的和去年做的早就不是一回事。用的系统可以变，从 Excel 换到 CRM 又换到某个新平台。连部门归属都可以变，市场划到销售底下，销售又划回总部。

只要某一类结果仍然需要稳定发生，岗位设计就不能只写动作，还要回答：谁可以调动资源，谁处理异常，谁接住结果，条件变了由谁改安排。这是本书从责任一侧看岗位的方式，不是说岗位只有这一种功能。

任务可以拆，执行可以转交。拆开之后，原来捆在一起的判断、批准、复核和纠偏，也可以分给不同的人或机制。但不能只把动作交出去，把权限、资源和后果留成一笔糊涂账。

把 AI 放进来，先看它实际接走了哪一段工作。能生成初稿，和能独立交付不是一回事；能按规则推进，和这套规则一直适用也不是一回事。执行安排变化以后，原来的授权、控制和责任是否仍然够用，要重新核对。

本书要求结果有人持续承接。这不等于每一个中间动作都由人决定，也不等于判断、批准、复核、纠偏和损失承担必须压在同一个人身上。要能找到各自的接口，查到实际权限，验证异常处置和交接，确认组织有资源接住后果。

第十八章留下了一个组织方式的选项：人机协同的责任单元。这里把它展开。**围绕一类可验收的结果，把负责的角色、可调用的人机能力、授权边界、控制与反馈安排放在一起，我把这样的设计称为责任单元。**

它是本书拿来讨论协作安排的一个单位，不是组织必须拆到的最小一格，也不要求在编制表上新加一行。既有岗位、部门或项目组若已能履行这些责任，可以沿用。

判断它是否站得住，不只问出了事找谁。还得问：被找到的人能否获得信息、调动资源、处理异常，超出权限时谁接手，人员变化后责任如何不断线。名字是入口，实际履行才是要检查的东西。

换一个名字，不能算换一种组织

责任单元这个说法，容易让人想起独立核算、事业部或项目组。名称相近或不同，都还不能说明安排有没有变化。

先分清两个问题：这笔收入与成本记在哪里，这个结果由谁持续承接。两种边界可能重合，也可能不重合。不能因为关注责任，就把核算当成次要问题；不知道总投入和损失落在哪里，也很难判断一个安排是否值得保留。

责任单元要增加的，是一项明确检查：跨过岗位或部门的工作，能否沿着结果追到有效的授权、控制和承接安排。已有机制回答得好，不需要再造一个新词。

一个人带着一组 Agent，可能完成一类结果；几个人分工，也可能完成。参与者数量不能单独划定边界，签一个名字同样不能。还要看结果是否可验收，任务与上下文是否接得上，权限够不够，异常和后果能否被及时处理。

个人能多做一些事，不等于组织已经多了一项可交接的能力。他不在时谁接，方法能否复用，条件变了谁更新，这些都得另查。责任单元这个名字有用与否，就看它能不能帮助我们把这些问题查清楚。

一个人做得更多，公司就该变小吗

一个人借助 AI 能交付更多工作，并不能直接推出公司应该变小，更不能推出公司一定会消失。这里至少隔着两笔账：把工作放到外部，要付多少寻找、协调、验收和持续维护的成本；留在组织内部，又需要多少管理和公共投入。AI 改变了哪些成本、改变多少，要在具体业务里算。

责任问题也不能替代这笔比较。它让我多问三件事：损失来了由谁承担，有没有可用的资源；错误发生后，谁能推动流程和规则真正改掉；人员变化以后，原有承诺如何继续履行。我把这三问简写成“赔得起、改得动、等得起”。

这是一组检查问题，不是公司的定义，也不能据此宣布只有公司能做到。内部组织安排、外部合作及其组合，仍要逐项比较：哪些责任确实有人承接，哪些只是写在承诺里，协调和维护的代价由谁承担。

对制造企业，这个检查同样不能只盯着屏幕里的流程。AI 改了信息采集或计划建议，要继续查它怎样进入排产、设备操作、交付和客户承诺。信息环节变快，不等于物理环节的约束消失；物理环节仍在，也不能反过来证明信息安排不必变。

所以我不在这里押公司变大还是变小。先把内部保留、外部委托、局部重配这几种安排放到同一组结果和成本要求下。哪一种拿得到更稳定的结果、能承担相应后果，又付得起持续运行的代价，再选哪一种。

能做多少，还要看接得住多少

一个单元该画多大，要看它交付什么结果，依赖哪些专业和资源，边界扩大以后还能否维持质量与控制。

假如一条错误规则被自动重复使用，它可能把一次错误扩散成一批错误；但扩散多快、多远，仍取决于权限、流量限制、监测和处置。不能只数 Agent，也不能只看某个人愿不愿意签字，就推定能承接多大的结果。

能力扩大了，承接条件没有跟上，就不能把边界照着能力往外推。这时可以补授权和资源，缩小范围，拆分职责，或者暂缓运行；不是只能再加一个人或再画一个格子。

后果承接是边界设计的一项约束，不是唯一尺度。执行效率、专业分工、共同知识、协作成本和投入产出仍然要算。本书选择把承接问题问得更细，不因此宣布它已经取代组织存在的其他理由。

我要求的检查很具体：不能因为审核环节都在、表格都齐，就推定它们真起了作用。

这一条连我自己那支 AI 团队也没能例外——三道审核叠在一起，照样让一句没跑过测试的「全部通过」穿了过去。它不能证明什么，只是让我更不敢把这句判断划掉。

这类经历能暴露一种失效方式，不能证明哪一种组织形态最优。接下来这五层，是我据此提出的一种安排，是否合适，还要回到真实结果、投入和交接条件上检验。

结果有人持续承接

责任单元，是把一组持续结果及其所需的权责、资源和协作安排放在一起检查的设计。它可以设具名牵头人，但不是要求一个人包办所有决定。

先看结果承接。谁负责什么结果，谁处理争议，谁能调动必要资源，谁接住已经发生的后果，都要明确。目标定义、专业判断、审批、叫停和验收可以分工。真正的问题是，有责任的人拿不到信息、调不动资源，或者几方都有一段权力，冲突发生后却没有有效的裁决和升级路径。

再看 AI 工作流。搜索、整理、生成、编码、分析、监控、按规则执行，可以随技术和业务变化重新组合。调整执行方式时，必须同时检查授权和承接条件；不能只把任务搬过去，把未结事项留在原处。

第三是上下文。企业知识、历史经验、可用数据、流程规则和判断依据，要让获授权的执行者取得，也要有纠错和更新的入口。存进去不等于用得对，自动写回不等于内容已经可信。

第四是控制机制。第 11 章的风险、恢复条件、判断要求和后果传播，提供的是选择控制的依据，不是一维对应一种人审方式。具体采用预授权限额、抽检、复核、审批还是禁止使用，要看权限、适用限制、控制效果和后果承接条件。逐单点头并不是责任成立的证明。

第五是对外责任链。客户、协作方和后来的接手者，能否在约定响应窗口内找到有权、有信息、有资源的承接者？组织内部怎样分工，不能成为外部问题无处处理的理由。

人可以换，班可以交，系统可以重做。必须持续的，是对结果的承接，不是某一个自然人永远坐在中心。交接要包含权限、资源、记录和未结事项；没有把这些交过去，换一个名字不叫接手。FDE 可以帮助这一安排成形，也可以完成合格交接后离开，不必永远兼任结果负责人。

原承接者 → 核实交接 → 新承接者。

交过去什么	怎么核实不是只换姓名
权限	接手者能否作出约定决定，越界时找谁
资源	人、系统、时间和必要支持能否实际调用
记录	能否查到判断依据、运行状态与已作承诺
未结事项	尚未处理的后果由谁继续跟进，何时回应

箭头表示需要核实的交接，不表示业务永不出错或交接已经完成。

相关，但不是同一个概念

可委托工作闭环，讲的是一件事是否具备交出去的条件。工作单元，讲的是任务被拆成什么颗粒。责任单元，讲的是这些工作及其后果由怎样的组织安排持续承载。三者不能当成别名互换。

一个责任单元可以承载多个工作单元；同一工作也可能需要多个单元协作，但接口、分工和争议处理不能因此悬空。任务拆分方式可以变化，承接安排也可以调整，前提是交接有效。

现有部门、业务线、班组是一种实际安排，不是等待改名的半成品。岗位则描述具体的职责与授权接口。要采用责任单元这个设计，就得说明它比原安排具体改变了什么：少了哪处交接断点，解决了哪种资源冲突，怎样改善质量、成本或后果处置。只有名字不同，就没有证明出现了新的组织能力。

说到这儿，我得把一件自己的事拿出来讲。

我自己公司里有过一位高级咨询师。她的活是跟客户开会、把需求搞清楚、用 PPT 把方案讲出来，做了大半年，图是手绘的，逻辑清楚，配色讲究。后来我自己跟客户谈完，转身让 AI 把方案讨论清楚、把 storyline 梳出来、把图和文字组装出来——原本一周的事，不到一杯咖啡的工夫。她没做错任何事，是岗位被我重新定义了。后来她自己另谋高就，我们体面地分了手。

我当年只能把她的岗位拆掉，那时候我给不出别的答案。

如果今天再遇到同样的情况，我会多问一层：这条线要拿回什么结果，原有岗位能否重新承担，所需权限、能力和支持如何补上。这是今天的推演，不是当时已经发生的安排，也不能保证换一个问法就能保住某个岗位。

再看我自己写书时的一段工作记录。

写这本书的时候边上跑着一支 AI 团队。有一天修个小毛病：一条注释跨了页，检查程序死活说它「在 PDF 里一次都没出现」，人眼看着明明在。执行的那一岗连改四轮匹配算法，四轮全败。第五轮才回头看最笨的一件事——被检查的 PDF 是两小时前的旧文件，前四轮分析的根本不是当前这版。

真因找到，改法上去，灯转绿。只看那盏灯，事就算完了。可留痕里记着一行数：这条注释被数成 6 份，实际是 2 份。**灯是绿的，数是假的**——算法一放宽，几个标点巧合的碎片也算成了匹配。是留痕自己把虚增抖出来的，不是那盏绿灯。补两条硬约束再跑，恰好 2 份。

那一天有几件事没变过：改的人不能自己判自己过，得另一岗独立复核；复核的不许只说「我看过了」，得把跑出来的输出贴上来，带通过数和失败数——**这条规矩是上次栽跟头换来的**。还有一处机器判定和实际对不上，执行岗没偷偷改掉它凑绿，是登记成一笔挂账，写明超出自己权限。最后东西摆到我面前，我一个字一个字看完，签字。

这套记录、独立复核和签认，帮我们追完了这一次错误。它没有消灭错误。要放进几千人的企业，还得另查跨部门资源、专业制衡、业务连续性和规模化后的控制成本，不能拿个人团队的局部实践直接拼组织图。

企业也不必按“工具、流程、岗位、责任单元”逐级通关。先找当前结果被什么卡住，再决定补哪一项；没有必要为了到达某个阶段而改组织。

先看哪里需要改，再决定画什么图

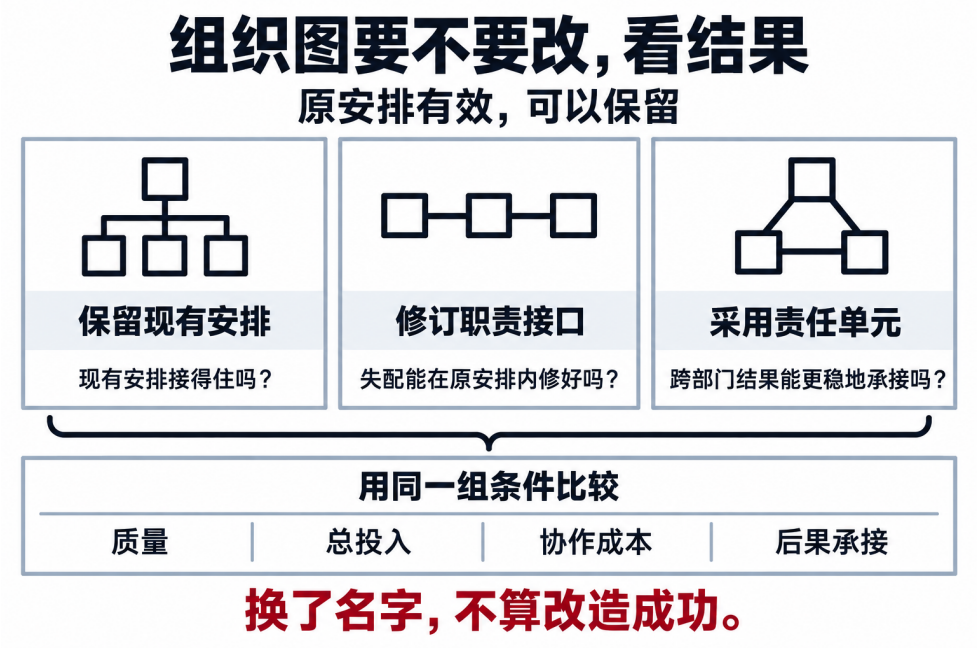

图 R10-F18 | 组织图要不要改，看结果。原安排有效可以保留，失配可以先修订职责接口；采用责任单元也是待验证的选项。三种安排用质量、总投入、协作成本与后果承接同尺度比较。上方图标仅作示意，不规定三种安排的唯一拓扑。

现在把镜头拉远，看整条结果怎样穿过组织。

组织图可以帮助我们找交接，也表达专业分工、授权和控制。层级多，不等于每一层都多余；减少一次交接，也不等于减少了全部协调成本。先看工作记录：等待发生在哪，返工为何发生，哪个决定需要谁的专业和权限。

如果 AI 确实缩短了研究、起草、编码或核对的一段工作，下一步要查的是：节省有没有传到最终结果，复核和维护有没有增加，瓶颈是否转移。执行仍然可能是瓶颈，技术和数据也可能先卡住；不能预先宣布，管理只剩判断和责任。

查清这些，才有资格比较组织安排。原来的岗位和部门能通过补接口、调权限、改流程把问题解决，就保留。跨部门反复掉交接、没有人能协调完整结果时，可以试责任单元；试下来还要与原安排比较，不能因为换了名字就算改造成功。

两种安排都需要可信的记录、专业判断、有效授权和持续交付。科层组织不是只有汇报线，责任单元也不能只靠签名取信。**组织图是否值得改，要由工作和结果来回答，不能由图本身来证明。**

如果选择较自主的单元，就要说明公共资源怎么分、冲突如何处理、谁能改变共同规则。可以把共享底座、结果承接和跨单元仲裁画出来，作为一种待检验的设计；它们未必对应三个管理层级，更不是所有公司的统一形态。

管理岗位也按实际任务重新看。信息汇总减少以后，是否还有需要保留的专业判断、协调、培养和控制工作？哪些可以委托，哪些必须另配资源？没有验清这些，不能从一段任务变快直接推出整个管理岗位被淘空。

责任单元若有价值，应当在这样的比较里显现：结果更稳，承接更清楚，总投入值得，换人以后仍能运行。做不到，就继续修改或撤回这个安排。

别先问要设几个，先问那条线后面站着谁

读到这里，先别开会讨论该设几个责任单元。先选一条重要的 AI 流程，把实际承接安排摊开。

这条流程出了问题，能不能在约定时间内找到有权处理的人？牵头者、专业判断者、批准者和接替者可以不同，但谁处理什么、冲突往哪里升级，不能靠临时猜测。答得出名字是入口，信息、权限和资源接得上，问题才有机会被处理。

CEO 先选一条线，明确结果、牵头安排、分工和所需资源；CHO 把岗位要求写成具体结果与授权边界，不预设所有判断都必须本人保留；CIO 和 AI 转型负责人把授权、留痕、停止条件以及第 11 章选定的控制方式接进系统。纸上写了什么，系统就应当能够执行、检查并留下记录。

再问三件事：拿回了约定结果吗？后果处理需要的资源到位吗？换人或换系统以后还能接得住吗？答不上来的先标为待查，不急着宣布整个部门无效。先约定观察窗口和验收条件，也不承诺所有改造都能在一个季度内验明。

哪些证据，会让我收回判断

我选择不让结果责任悬空，这是本书的规范立场。但怎样实现它，哪种组织设计更好，必须接受经验检验。

我能拿出来的是这些局部实践和一组待检验的设计。评价别人的方案，也应要求对方说明适用范围、实际证据和未解决的问题，不能因为我没见过完整样板，就说完整样板一定不存在。

如果我建议某条流程重配接口，而原来的安排在同样目标和约束下已经稳定受益，我就撤回这条流程必须重配的建议。如果原部门、项目制或流程负责人做得一样好，甚至更好，我就撤回责任单元更优、应当普遍推广的主张。

如果我把审批等待判成瓶颈，就该在动手前说清：改哪一处、预期先改变哪个信号、用什么记录来观察。资源确实到位、改动确实实施，等待却没有减少，就要重查这个诊断；如果等待减少了，质量或净收益却更差，就不能把“流程更规范”当成功。

如果一套做法声称可以复制，就要先约定合格接手者、支持条件、场景差异和维护成本。可比条件下仍然每次重做，而且拿不到可承受的增量，就把它降为本地办法。不能把难例移出样本，保住一个好看的复制率。

如果在相同约束、约定观察期和可比较任务上，AI 承担复核、监控与纠偏的方案，能够达到质量和风险底线，整体效果不差、总成本更低，我就不坚持保留原来那道人工确认。撤掉按钮之前，权限、适用限制、异常阻断及后果承接仍须分别核实；控制方式可以换，不能把自动运行误当成组织不再需要负责。

这些比较要在事前写清目标、质量底线、成本范围和观察窗口。证据不足，结论就是还不知道，不是已经被推翻，也不是已经被证明。

本书的答案因此不压在一张新组织图上。原安排能满足，就保留；不能满足，就把查明的失配处改清楚。责任单元网络是一种可供检验的设计，不是所有组织唯一的终点。

最后押一个注，以这本书写作的 2026 年为起点。

三年以后你招高管，招聘启事上那一栏写的可能不再是部门名字，是一个后果域——不是「负责市场部」，是「对获客这条线的结果负责，包括它砸了的时候」。到那天，组织图上的格子就不是按谁管谁画的，是按谁认哪笔账画的。

这话我押在这儿。2029 年你翻回这一页，看我押错没有。它是我的预测，不是责任单元成立的前提。

而这个设计成不成立，不能只看出事时找不找得到谁。还要分别验证：有没有取得约定结果，承接者是否真有信息、权限和资源处理后果，交接以后能否持续，迁到新场景是否仍然成立。名字让问题有入口，能力让后果有人接，结果决定这套安排值不值得继续。

APPENDIX A

附录 A：全书工具说明 T01-T12

A1 正文承重工具

T02 五断点诊断图

什么时候用

当团队说“AI 转型很乱”，但说不清乱在哪里时使用。

谁主持

AI 转型负责人或一号位指定的组织 owner。

输入

- 一个具体 AI 项目；
- 项目涉及的岗位、流程、知识、责任和治理对象；
- 当前失败或卡顿现象。

会上问什么

- 岗位有没有被拆成任务、判断、责任和结果？
- AI 有没有进入真实流程节点？
- 好输出有没有沉淀成组织记忆？
- 出错以后谁复核、谁验收、谁负责？
- 权限、审计、留痕、回滚有没有进入日常运营？

会后留下什么

- 主断点；
- 次断点；
- 90 天试点优先级。

T05 Demo 存活检查表

什么时候用

在老板准备继续给 Demo 投资源之前使用。

谁主持

一号位 / 业务负责人。

输入

- Demo 演示材料；
- 业务流程；
- 使用人群；
- 上线预期；
- 已知风险。

会上问什么

- 它替代或增强哪个真实节点？
- 谁每天必须用？
- 不用它会不会影响业务结果？
- AI 输出错了谁复核？
- 它是否进入现有系统？
- 90 天后留下什么组织资产？

会后留下什么

- 继续 / 暂停 / 重写判断；
- 上线前缺口清单；
- owner 和验收指标。

T06 组织记忆十问

什么时候用

当员工已经开始用 AI，但组织没有明显变聪明时使用。

谁主持

CHO / 知识管理 owner / AI 转型负责人。

输入

- 员工 AI 使用样本；
- 高质量输出；
- 复盘材料；
- 知识库现状；

- 权限和更新机制。

会上问什么

- 好输出有没有沉淀?
- 好 prompt 有没有被复用?
- 好判断有没有留下场景条件?
- 知识谁维护?
- 过期内容怎么发现?
- 员工贡献业务上下文 后，组织如何认可?

会后留下什么

- 组织记忆缺口;
- 知识沉淀责任人;
- 第一批可结构化知识对象。

T07 HITL 责任链表

什么时候用

当 AI 输出会影响客户、员工、财务、合规或关键业务结果时使用。

谁主持

业务 owner / 治理负责人。

输入

- AI 使用场景;
- 输出结果类型;
- 风险等级;
- 当前复核方式;
- 事故处理记录。

会上问什么

- 谁下达指令?
- AI 做了什么?
- 输出进入哪里?
- 谁复核?
- 谁验收?
- 谁能改规则?

- 出错以后谁修流程、知识库和权限？

会后留下什么

- 责任链；
- 复核要求；
- 留痕要求；
- 异常升级机制。

T09 产能分配矩阵

什么时候用

当 AI 明显提升效率，老板开始讨论裁员、增产或重组时使用。

谁主持

CEO / CHO / CFO。

输入

- AI 前后产出数据；
- 人力成本；
- 业务增长状态；
- 质量和返工数据；
- 组织风险。

会上问什么

- 结果是否更稳定？
- 产出是否更多？
- 返工是否更少？
- 能力是否沉淀？
- 信任是否受损？
- 产能应该去降本、增产、提质、重组还是学习？

会后留下什么

- 产能去向判断；
- 不裁 / 暂缓 / 转岗 / 缩编 / 增产方案；
- 风险兜底条件。

T10 90 天 AI 组织试点路线图

什么时候用

当一号位决定用一个真实流程验证 AI 组织改造时使用。

谁主持

AI 转型负责人。

CEO 参与启动会和验收会。

输入

- 试点流程；
- 断点诊断；
- owner；
- 业务指标；
- 责任链草图。

会上问什么

- 0-30 天诊断什么？
- 31-60 天重写什么？
- 61-90 天验证什么？
- 90 天后必须留下什么制度资产？
- 如果结果不稳定，如何停止或回滚？

会后留下什么

- 90 天路线图；
- 里程碑；
- 复盘节奏；
- 复制或停止条件。

T11 老板 AI 组织试点会议表

什么时候用

当老板要把“聊 AI”变成“定流程、定 owner、定指标、定复盘”时使用。

谁主持

CEO / 总经理。

输入

- 一个候选 AI 试点；
- 涉及部门；
- 当前流程；
- 预期经营结果；
- 主要风险。

会上问什么

- 这条流程为什么值得重写？
- 谁是业务 owner？
- 90 天后看什么结果？
- AI 出错谁复核？
- 哪些传统流程不能先动？
- 哪些创新流程可以并行跑？

会后留下什么

- 试点是否启动；
- owner；
- 指标；
- 责任界面；
- 复盘时间。

A2 附录工具

T01 AI 组织 OS 自查表

什么时候用

在一号位刚准备启动 AI 转型，或者公司已经买了工具但不知道哪里卡住时使用。

它不解决具体方案。

它先判断公司是不是还停在“工具上线”阶段。

谁主持

CEO / 总经理 / AI 转型负责人。

CHO、CIO、业务负责人必须在场。

输入

- 当前 AI 工具清单；
- 已经跑过的 Demo / 试点；
- 当前最重要的 3 条业务流程；
- 已知的 AI 使用问题。

会上问什么

- AI 现在主要在个人层，还是流程层？
- 员工用 AI 以后，组织有没有留下新流程、新知识或新责任？
- 哪些 AI 使用已经影响客户、员工、财务或决策结果？
- 如果今天出错，谁能说清复核链路？

会后留下什么

- 一页自查结果；
- 当前 AI 转型阶段判断；
- 下一场五断点诊断会的议题。

T03 工作片段拆解表

什么时候用

在讨论岗位是否会被 AI 替代之前使用。

先拆工作片段，再谈岗位变化。

谁主持

CHO / HRBP / 业务负责人。

输入

- 一个关键岗位；
- 该岗位真实工作清单；
- 近期业务结果和绩效口径；
- 岗位相关 AI 使用情况。

会上问什么

- 哪些动作是重复执行；
- 哪些动作是信息整理；
- 哪些动作是标准化判断；

- 哪些动作需要文化匹配、信任判断或组织风险判断；
- 哪些结果必须有人兜底。

会后留下什么

- 工作片段表；
- AI 可接走片段；
- 人保留或获准委托的判断及依据；无须逐次人工决定的，注明范围和验证记录；
- 岗位重写候选项。

T04 流程重写画布

什么时候用

当一个 AI 项目准备从 Demo 进入真实业务时使用。

谁主持

业务 owner。

CIO / AI 转型负责人 / CHO 参与。

输入

- 当前流程图；
- 当前痛点；
- AI 可介入点；
- 业务指标；
- 风险边界。

会上问什么

- 这条流程现在最慢、最贵、最不稳定的是哪一步？
- AI 进入以后，哪一步可以取消、合并、提前或自动触发？
- 哪些判断仍然必须由人做？
- 异常怎么升级？
- 新流程谁负责持续维护？

会后留下什么

- 新流程草图；
- AI 介入节点；
- 人机分工；

- 新 owner；
- 试点指标。

T08 AI 节点控制表

什么时候用

为一个具体 AI 动作确定授权和控制方式，或者用途、规模、输入及后果路径发生变化时使用。草拟和发送、参考和直接生效分别建节点。

谁主持

具有相应风险决策权限的业务 / 治理负责人，技术与实际执行者共同核查；需要时让受影响的岗位参与。不是由工具使用者单独给自己放行。

输入

节点的输入、动作、输出去向及被依赖时刻；数据和行动权限、适用限制；可能损害的对象、规模、持续与累积影响；事前固定的恢复目标、时间、成本和剩余影响界线；恢复与控制验证记录；异常及后果承接安排。

会上问什么

先按同一把尺判断能否证明满足恢复条件。不能，归“不能证明能恢复”，注明是证据不足还是已知不能；能，再按是否依赖其他主体配合，分“协调恢复”或“直接恢复”。

然后检查目标与规则覆盖、例外及后果传播。同一用途若命中禁止条件，先停止该用途，不把“禁止使用”与放行方式组合。在获准范围内，再比较自由使用、人工抽检、人工复核和人工审批，选择彼此兼容的控制。三档恢复条件不直接对应控制方式；签字不改变恢复能力，可逆也不代表有权做。

会后留下什么

每条判断的依据与缺口；授权范围、限制及选定控制；停止或限流条件；负责人、处置时限和承接资源；复评周期与变化触发。条件未满足的动作不得因表已填完而放行。

此表属于本书设计，尚非经真实读者分类和业务效果验证的标准。应在限定场景中记录分歧、漏项与失败，再调整判据。

T12 FDE 组织翻译对照表

什么时候用

当公司发现 AI 项目卡在业务和技术之间，没人能把现场问题翻译成系统、流程和组织机制时使用。内部 FDE、外部 FDE 或联合团队均适用；先确定任务，再选择现岗兼任、临时小队或长期专岗。工程交付与组织承接须明确分工，不要求所有职责都压给一人。

谁主持

AI 转型负责人 / 业务负责人。

输入

- AI 项目卡点与业务结果基线；
- 管理者口述、制度文件和真实业务记录中的流程差异；
- 技术方案与现场流程损耗图；
- 可用数据、权限、资源及其授权人；
- 试点业务负责人、工程交付负责人和预定运营承接者。

会上问什么

- 这段业务是否值得改？删掉无效步骤能否比自动化更合适？三种流程口径如何用真实记录核实？
- 谁来牵头？不论工程或业务出身，能否用同一条真实流程证明业务判断、工程交付，以及对权责、协作和采用的组织判断？哪些专业需要伙伴补位？
- 本次只交付系统，还是同时孵化业务责任单元？谁从试点开始对业务结果负责，谁能改流程、调资源、认验收、裁决异常？授权不足时谁决定缩范围或暂停？
- 谁把经验沉淀为五样组织资产？预定承接者如何证明自己能日常运行、核算投入产出、处理异常和更新规则？
- 孵化者留任还是交接？交接后谁负责业务运营、谁负责技术维护？达不到标准时谁决定续试或停止？

会后留下什么

- 现场任务约定及任职方式，明确是否包含孵化任务；
- FDE / BA / 项目经理 / 技术实施 / 业务负责人的交付与问责边界；
- 授权、资源、验收、异常升级清单，注明未获授权事项；
- 五样组织资产的交付清单；若含孵化任务，再附运营承接人与交接验收记录。

APPENDIX B

附录 B：本书提到的案例与出处

这不是论文的参考文献表。这是一份“想深挖就顺藤摸瓜”的清单——全书正文里可以公开核查的外部案例、学者、法规和数据，在这里各占一行，告诉你它是什么、书里拿它说明什么、原始出处在哪儿。作者亲历但需要匿名处理的实务案例不在这里展开。

我们不假设你已经知道这些。所以每一条都写了人话解释。你可以只读正文，也可以拿着这份清单去查原文——后者会让你对某一章的判断多一层自己的底气。

B1 案例与一手材料

这一节混着两类东西：真有公司干过的事（Klarna、Cloudflare、Y Combinator 的创业方向清单、两个 GitHub 开源项目），以及可以直接翻开看的一手材料（斯坦福和 MIT 的落地报告、Google SRE 的工程手册、微软与 AWS 的官方文档、OpenAI 与 Anthropic 的岗位说明书）。放在一起是因为它们都能被你亲手打开核对；只想看“别家公司到底干了什么”，跳着看前两条和 Y Combinator、GitHub 那两条即可。

Klarna（瑞典先买后付公司） 它是什么：一家金融科技公司。其 2024 年 2 月 27 日公告称，AI 客服助手上线首月处理了两百三十万次对话，占其客服聊天的三分之二，工作量折合七百名全职客服；2025 年 3 月 14 日提交的 F-1 披露，AI 助手上线以来处理了 62% 的客服聊天，工作量折合八百多名全职客服，并说明了估算方法。 书里拿它说明什么：第 3 章用这两份公开披露区分工作量折算、人员实际变动、服务质量与投入。不同日期和口径的数字不能直接拼成“前后改口”，也不能据此诊断公司内部的决策过程。 可查线索：Klarna 2024 年 2 月 27 日公司公告；Klarna Group plc 2025 年 3 月 14 日 F-1。均属公司披露，不是独立效果评估。（第 3 章注①）

Cloudflare（美国网络基础设施公司） 它是什么：2026 年 5 月官方博客公告裁减 1100 多个岗位；而在这之前大半年，2025 年 9 月，它公告过一个 1111

人的实习生录用计划。书里拿它说明什么：第 14 章用这两个数字并置，说明“AI 裁员、用实习生对冲”这类外部解读跑得有多快，以及老板不该照着新闻标题给自己定裁员时间表。可查线索：Cloudflare 官方博客《为未来而建》，2026 年 5 月 7 日发布，网址为 blog.cloudflare.com/building-for-the-future/。实习生计划见同站 1,111 人实习生计划公告，2025 年 9 月 22 日发布，网址为 blog.cloudflare.com/cloudflare-1111-intern-program/。（第 14 章注①）

斯坦福数字经济实验室：51 个企业 AI 落地项目 它是什么：斯坦福大学数字经济实验室 2026 年 4 月发布的企业 AI 落地报告，覆盖 7 个国家、41 个组织的 51 个项目、超过一百万名员工，这些项目全都过了试点、能交出可量化的业务价值。书里拿它说明什么：第 1 章用它证明真正的难点不在技术——51 个成功项目里 77% 把变革管理、数据质量、流程重写列为难点，一件都不是技术问题。第 8 章用它引出影子 AI：报告转引的一项行业调查显示，使用 AI 的员工里有七到八成用过未经公司批准的工具（这个比例是报告转引的外部数字，不是那 51 个案例的原创统计）。可查线索：Elisa Pereira、Alvin Wang Graylin、Erik Brynjolfsson，《The Enterprise AI Playbook: Lessons from 51 Successful Deployments》，Stanford Digital Economy Lab，2026 年 4 月，p.7、p.13、第 89 页。（第 1 章注①②，第 8 章注①）

MIT 工业绩效中心：《Humans in the Loop》 它是什么：麻省理工学院工业绩效中心 2026 年 4 月 8 日发布的报告，作者是该中心执行主任 Ben Armstrong 和 MIT 航空航天系主任 Julie Shah。书里拿它说明什么：第 4 章用它佐证“先动的是任务，跟着动的是人的位置”——它看的正是企业在生成式 AI 早期实验里，搜索、起草、汇总、判断、复核、沟通这些动作怎么被拆开又重新组合。书里同时提醒：这是早期实验的观察，不是大规模铺开后的定论。可查线索：Ben Armstrong、Julie Shah，《Humans in the Loop: The evolution of work in early experiments with generative AI》，MIT Industrial Performance Center，2026 年 4 月 8 日。（Julie Shah 的航空航天系主任职务自 2024 年 5 月 1 日起生效，见 MIT News 2024 年 4 月 29 日公告。）（第 4 章注①）

Google SRE 的验尸文化（Postmortem Culture） 它是什么：Google 一批专门负责让线上服务不宕机的工程师（SRE，站点可靠性工程师）写事故复盘报告的一套硬规矩。书里拿它说明什么：第 12 章用它说明组织怎么从“出了事找

人”挪到“出了事修系统”——工程界这套经验是现成的，可以直接搬。 可查线索：Betsy Beyer、Chris Jones、Jennifer Petoff、Niall Richard Murphy 编，《Site Reliability Engineering: How Google Runs Production Systems》第 15 章“Postmortem Culture: Learning from Failure”（John Lunney、Sue Lueder 撰），O’Reilly，2016，全文见 sre.google/sre-book/postmortem-culture。（第 12 章注①）

微软 Azure 云采纳框架与 Purview 审计日志 它是什么：微软官方的 AI 治理与安全文档。本书分别标注所引版本，不将访问时的功能当成永久能力清单。 书里拿它说明什么：第 13 章用它说明技术控制还需要组织接手。最小权限只开放本职工作必需的数据；Purview 在支持的应用与记录中提供交互、资源及检测标记。字段存在不等于全量覆盖或准确检出。事故响应则要提前安排关停、对外沟通、日志保存和关键业务演练。 可查线索：Microsoft，《Governance and Security for AI Agents Across the Organization》，Azure Cloud Adoption Framework，文档更新日期 2026 年 4 月 16 日，事故响应见 Agent Security 第 10 条；《Audit logs for Copilot and AI applications》，Microsoft Purview，所核版本更新日期 2026 年 8 月 26 日，2026 年 9 月 6 日访问，learn.microsoft.com/en-us/purview/audit-copilot。（第 13 章注①②）

AWS CloudTrail 数据事件异常检测 它是什么：亚马逊云 2025 年 11 月发布的数据事件检测功能，建立访问模式基线，检测到异常时生成事件；告警通知需要配置。 书里拿它说明什么：第 13 章用它说明机器可以帮助筛异常，但检测范围、漏检风险和异常处置仍须有人负责，不能把一项功能等同于全部治理完成。 可查线索：Amazon Web Services，《AWS CloudTrail launches Insights for data events to automatically detect anomalies in data access》，AWS 官方公告，2025 年 11 月 20 日。（第 13 章注③）

Y Combinator：Company Brain（公司大脑） 它是什么：YC（硅谷那家孵化了 Airbnb、Stripe 的创业营）每年公布的“我们想投什么”清单里的一个创业方向。讲的是把企业里散落的业务门道抽出来、理清楚、持续更新，让 AI 能读懂这家公司到底怎么运转。 书里拿它说明什么：第 7 章拿它当起点，然后往前追问一步——放到一家开了二十年的传统公司里，这件事还差什么。 可查线索：

Y Combinator，《Request for Startups》（Summer 2026），www.ycombinator.com/rfs，“Company Brain” 条目。（第 7 章注①）

两个 GitHub 开源项目：技能提取 vs 反蒸馏 它是什么：一个项目专门把同事的工作方式、判断模式、处理风格提炼成 AI 能学的技能文件，到 2026 年 7 月已有两万多星标；另一个叫“反蒸馏”，干的事相反——把员工自己写的文档输出成一个看起来完整专业、其实核心判断已经被抽掉的清洗版，真东西留一份私人备份。书里拿它说明什么：第 7 章用这一对项目说明提取与被提取已经同时开始，不再是推演。书里也标了边界：项目功能均为其公开自述，不是第三方评测结论。可查线索：GitHub 项目 titanwings/colleague-skill（后更名 dot-skill），github.com/titanwings/colleague-skill，星标数取自 2026 年 7 月 26 日页面快照；GitHub 项目 leilei926524-tech/anti-distill，github.com/leilei926524-tech/anti-distill。（第 7 章注⑦）

OpenAI 与 Anthropic 的 FDE 岗位说明书 它是什么：两家 AI 公司公开招聘页上的“前线部署工程师”（Forward Deployed Engineer）岗位说明书。书里拿它说明什么：第 22 章用它说明这个岗位的成功标准长什么样。OpenAI 那份写的是三件事：生产采用、可量化的流程影响、评测驱动的反馈。Anthropic 那份要求交付能进生产工作流的 MCP servers、sub-agents、agent skills，还要把可重复的部署模式沉淀回产品和工程团队。可查线索：OpenAI，《Forward Deployed Engineer》招聘岗位说明书，openai.com/careers，2026 年；Anthropic，《Forward Deployed Engineer, Applied AI》招聘岗位说明书，Anthropic 官方招聘页，2026 年。（第 22 章注①②）

Meta「Claudeonomics」内部排行榜 它是什么：关于员工自建用量看板的媒体报道，正文仅保留其出现与撤下，不复述未经本轮逐项核验的消耗估值。书里拿它说明什么：第 16 章用它提出“用量排名是否代表工作结果”的管理问题，不以撤榜证明成本反噬，也不把不同媒体的转引算作独立经营证据。可查线索：Fortune，2026 年 4 月 9 日 Meta 员工用量看板报道，fortune.com/2026/04/09/meta-killed-employee-ai-token-dashboard/。（第 16 章注②）

亚马逊 KiroRank 非正式用量看板 它是什么：Business Insider 2026 年 5 月 29 日报道的员工非正式看板，公司发言人确认已停用。书里拿它说明什么：第

16 章借此追问用量指标与工作结果的区别，不据撤榜推断全公司成本或高管动机。 可查线索： Business Insider， businessinsider.com/amazon-ai-leaderboard-tokenmaxxing-2026-5，2026 年 9 月 6 日核查。（第 16 章注③）

美团：王莆中“养虾”公开演讲的网易转载实录 它是什么：网易号“互联网坊间八卦”于 2026 年 8 月 14 日转载的一份王莆中公开演讲实录；本书内部留存快照仅用于核对该转载的第 4–6 页过程叙述。 书里拿它说明什么：第 19 章把它作为归因型读者案例，提示工具扩散后仍须由业务、组织与技术共同承接；它不构成美团官方复盘、独立经营审计，或本书框架已被经营结果验证的证据。 可查线索：网易号“互联网坊间八卦”转载《美团王莆中最新演讲全文，中小企业如何正确的使用 AI？（实录）》，2026 年 8 月 14 日；内部留存快照 SHA-256：5b366254b37a5f5efd181ee905c11668f038e32f0b965595393fe1f6aeb7a1e6 。（第 19 章注②③）

B2 学者与理论

Yu 等：AI 采用、HR 实践与员工福祉 它是什么：以既有 AMO 框架组织 HR 实践建议的综述；同时关注心理、社会和身体福祉，不是本书岗位工具的验证实验。 书里拿它说明什么：第 6 章引入能力、动机、机会三个检查方向；第 16、17 章回用这一视角，区分复用、贡献和参与的条件。不以少分享证明藏私，不把转岗单案说成 AMO 干预有效。 可查线索：Yu, S., Kawasaki, S., Magni, F., Yu, K. Y. T., & Munusamy, V. P. (2026). “Optimizing AI adoption through HR practices: Systematic review and theoretical framework for improved employee well-being.” Human Resource Management Review, 36, 101152. DOI: 10.1016/j.hrmr.2026.101152。（第 6 章注④；第 16、17 章回指）

Papagiannidis 等：责任 AI 治理的三类实践 它是什么：组织责任 AI 治理实践的范围综述及概念框架，区分结构性、程序性、关系性实践；相关关系仍需进一步实证研究。 书里拿它说明什么：第 13 章说明最低规则不是全部治理，第 19 章检查权责、运行和参与是否落实。不把六角色表、90 分钟会议或责任单元归为论文已经验证的方案。 可查线索：Papagiannidis, E., Mikalef, P., & Conboy, K. (2025). “Responsible artificial intelligence governance: A review and research

framework." The Journal of Strategic Information Systems, 34, 101885. DOI: 10.1016/j.jsis.2024.101885。（第 13 章注⑦；第 19 章回指）

Michael Polanyi（波兰尼）——隐性知识 他是什么人：最早把"隐性知识"这个词讲清楚的哲学家。核心意思是：人知道的东西，常常多于他能说出来的东西。他举的例子是认脸——你能从一百万张脸里认出一个人，却说不出你是怎么认出来的。 书里拿他说明什么：第 7 章用他给"员工 know-how 为什么提取不出来"打地基。 可查线索：Michael Polanyi，《The Tacit Dimension》，University of Chicago Press，1966 年，第 4 页。（第 7 章注②）

Nonaka 与 Takeuchi（野中郁次郎、竹内弘高）——组织知识创造 他们是什么人：研究日本企业为什么能持续出新产品的两位学者，把隐性知识和显性知识之间的转换放在组织知识创造的核心位置。 书里拿他们说明什么：第 7 章用来支撑"个人经验怎么变成组织资产"这条转换路径不是新发明。 可查线索：Ikujiro Nonaka、Hirotaka Takeuchi，《The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation》，Oxford University Press，1995 年。（第 7 章注③）

OECD（经合组织）——四类知识 它是什么：发达国家的政策研究机构。在讨论知识经济时区分过 know-what、know-why、know-how、know-who 四类知识。 书里拿它说明什么：第 7 章用这个四分类把"公司到底该提取哪一类知识"说细。 可查线索：OECD，《The Knowledge-Based Economy》，文号 OCDE/GD(96)102，1996 年，四分类章节。（第 7 章注④）

Daniel Miessler——公司是一堆算法 他是什么人：长期写 AI 与安全的独立技术博主，读者以工程师为主。他的判断是：公司说到底就是一堆算法的集合，不管产品多特别，运作起来还是一条条步骤管道；他还有一句更狠的——可解释性是新的货币，一旦 AI 能看见公司每一步怎么走，它就会挨个识别冗余并消灭掉。 书里拿他说明什么：第 7 章用他把"员工隐性经验不会一直是护城河"推到极致，同时明确标注这是个人博客的判断、不是研究结论。书里还追了一笔：他在文末留口子说后面会写哪些人类工作能留下来，但他后来写的是"公司理想员工数是零"和"只要 AI 干得了公司就不会再雇人"，那个正面答案一直没给。 可查线索：Daniel Miessler，《Companies Are Just a Graph of Algorithms》（技术博客文章，非研究论文），2024 年 5 月 6

日，danielmiessler.com/blog/companies-graph-of-algorithms；《The End of Work》，2024 年 8 月 26 日，danielmiessler.com/blog/real-problem-job-market；《The Bubble Is Labor》，2025 年 12 月 8 日，danielmiessler.com/blog/the-real-bubble-is-human-labor。（第 7 章注⑤⑥）

M.C. Elish（伊利什）——道德溃缩区 她是什么人：人类学家，专门研究自动化系统出事以后责任落到谁头上。她提出 moral crumple zone（道德溃缩区）：自动化系统出问题以后，责任很容易被压到那个离事故最近、但实际控制权有限的人身上。汽车的溃缩区吸收撞击保驾驶员，道德溃缩区吸收的那一下保的是系统，赔进去的是人。 书里拿她说明什么：第 10 章用它解释为什么“人工确认章”危险——盖章的人看起来在控制系统，实际只是替系统吸收责任冲击。 可查线索：M.C. Elish，《Moral Crumple Zones: Cautionary Tales in Human-Robot Interaction》，Engaging Science, Technology, and Society 第 5 卷（2019），第 40–60 页。DOI: 10.17351/ests2019.260。（第 10 章注①）

Santoni de Sio 与 van den Hoven——有意义的人类控制 他们是什么人：荷兰代尔夫特理工的两位技术哲学学者，做的事是给“机器自己干活的时候人该管到什么程度”找一个说得清的标准。他们讲的 meaningful human control，核心不是“有人在场”，而是系统要能回应现场理由、结果要能追溯到真实的人和组织链条。 书里拿他们说明什么：第 10 章用来把 HITL 从“有人点确认”拉回到“有人真的控制得住”。 可查线索：Filippo Santoni de Sio、Jeroen van den Hoven，《Meaningful Human Control over Autonomous Systems: A Philosophical Account》，Frontiers in Robotics and AI 第 5 卷第 15 号（2018）。DOI: 10.3389/frobt.2018.00015。（第 10 章注②）

Tushman 与 O’Reilly（塔什曼、奥赖利）——双元组织 他们是什么人：两位研究组织变革的学者，1996 年就系统写过“一边稳定交付、一边探索新东西”这套两条腿走路的打法，管理学里叫双元组织。 书里拿他们说明什么：第 20 章用它给“利用系统”和“探索系统”这对说法找学术根，同时强调名字不重要，重要的是两条腿都得有。 可查线索：Michael L. Tushman、Charles A. O’Reilly III，《Ambidextrous Organizations: Managing Evolutionary and Revolutionary Change》，California Management Review，1996 年，38 卷 4 期，第 8–29 页，DOI: 10.2307/41165852。（第 20 章注①）

Goodhart（古德哈特）——指标一旦被拿来控制就会失效 他是什么人：英国经济学家，1975 年在一篇讲英国货币政策的论文里提出，任何被观察到的统计规律，一旦被施加控制压力，就会趋于失效。 书里拿他说明什么：第 16 章用它给“考什么、员工就往哪跑”找学术根——这不是 AI 时代的新病。 可查线索：Charles A. E. Goodhart，《Problems of Monetary Management: The UK Experience》，收入《Monetary Theory and Practice》，London: Macmillan Education UK，1984 年，第 91–121 页（原文 1975 年发表于 Papers in Monetary Economics 第 1 卷，Reserve Bank of Australia）。另需说明：广为流传的“当一个指标成为目标，它就不再是一个好指标”并非 Goodhart 本人表述，出自 Marilyn Strathern，《‘Improving ratings’: audit in the British University system》，European Review 第 5 卷第 3 期（1997），第 305–321 页。（第 16 章注①）

Holmström 与 Milgrom（霍姆斯特朗、米尔格罗姆）——多任务下的激励设计 他们是什么人：两位研究激励机制的经济学家，1991 年系统分析了当一个人同时干好几件事、而只有其中一部分能被测量时，激励该怎么设计。 书里拿他们说明什么：第 16 章用它解释为什么绩效不能跟单一指标充分挂钩——挂得越紧，不可测量的那部分就掉得越快。 可查线索：Bengt Holmström、Paul Milgrom，《Multitask Principal–Agent Analyses: Incentive Contracts, Asset Ownership, and Job Design》，The Journal of Law, Economics, and Organization，1991 年，第 7 卷特刊，第 24–52 页，DOI: 10.1093/jleo/7.special_issue.24。（第 16 章注①）

《哈佛数据科学评论》：人审 AI 并不天然可靠 它是什么：发在 Harvard Data Science Review（一份专门研究数据与决策的学术期刊，简称 HDSR）上的一项研究，专门看人怎么评估 AI 给的建议。结论不太好听：人审 AI 并不天然可靠，人自己也会被 AI 的建议带跑。 书里拿它说明什么：第 8 章用它提醒别把“有人点了确认”当成责任闭环。 可查线索：Jacob Beck、Stephanie Eckman、Christoph Kern、Frauke Kreuter，《Bias in the Loop: How Humans Evaluate AI-Generated Suggestions》，《哈佛数据科学评论》第 8 卷第 2 期，2026 年春季号。（第 8 章注②）

B3 法条、报告与数据

欧盟《人工智能法案》第 14 条——人类监督 它是什么：同一部条例里针对高风险 AI 系统的人类监督条款。它要求监督要能防止或减少风险；监督者要能理解系统的能力和局限、对自动化偏差保持警觉、能正确解释输出；还要求监督者在必要时可以决定不使用、推翻、逆转或中止系统输出。 书里拿它说明什么：第 10 章用这几条说明同一件事——监督者手里得有权，不是只有一支笔。 可查线索：Regulation (EU) 2024/1689 第 14 条「人类监督」，2024 年 6 月 13 日。官方文本见 EUR-Lex：eur-lex.europa.eu/eli/reg/2024/1689/oj。（第 10 章注③）

JAMIA：算法公平性会随时间漂移 它是什么：发在《美国医学信息学会杂志》（JAMIA）上的一项纵向研究。用 2013 年的数据训练三个预测手术后再入院和死亡风险的模型，然后从 2014 到 2023 年每个季度评估一次。十年下来三个模型都出现跨指标的时间性漂移；肺炎模型的准确率在不同族群里都在下降，但下降的坡度不一样陡。研究者的判断很直接：算法公平性没法靠开发阶段的一次性评估来保证。 书里拿它说明什么：第 10 章用它——AI 系统不是装好了就一直是装好时那个样子，所以每一档 HITL 都得有复检的钟点，不是定完就锁死。书里也提醒：这项研究做的是医疗预测模型，别直接套到自己的业务上，但机制是通用的。 可查线索：Sharon E. Davis 等，《算法偏见的浮现：公平性漂移是模型维护与可持续性的下一个维度》，载《美国医学信息学会杂志》（JAMIA）第 32 卷第 5 期，2025 年，845 页起，DOI 为 10.1093/jamia/ocaf039。（第 10 章注④）

NIST《人工智能风险管理框架》MANAGE-2.2 它是什么：美国国家标准与技术研究院（NIST，美国政府里专门给各行业定技术标准的那家机构）那套 AI 风险管理框架里的一条：系统上线之后，得有一套持续跑着的机制，保住它当初被寄予的那个价值。具体说就是持续监控、异常检测、定期重新评估。 书里拿它说明什么：第 13 章用它说明“定期回到桌面上”不是作者一个人的主张——国际上写框架的人都把它写进条款了。 可查线索：NIST，《AI Risk Management Framework 1.0》（NIST AI 100-1），2023 年 1 月 26 日发布，MANAGE-2.2 条款原文“Mechanisms are in place and applied to sustain the value of deployed AI systems.”。此处表述据 NIST 官方 Playbook 页面（airc.nist.gov）核实，为框架条款的官方诠释版本。（第 13 章注⑤）

Moffatt v. Air Canada——聊天机器人错误答复的民事担责判例 它是什么：加拿大不列颠哥伦比亚省民事裁决庭（Civil Resolution Tribunal）2024 年 2 月对一起航空公司聊天机器人误导乘客案作出的裁决。裁决认定该公司未尽到合理注意义务确保其聊天机器人的准确性，需为聊天机器人的错误答复担责。 书里拿它说明什么：第 13 章用它说明权限、审计、留痕、回滚这四件事不只是内部管理动作，在法律层面也是组织能否证明自己尽到注意义务的关键证据；裁决问的不是 AI 有没有出错，而是组织有没有为它出错做准备。 可查线索：Moffatt v. Air Canada, 2024 BCCRT 149，加拿大不列颠哥伦比亚省民事裁决庭裁决，2024 年 2 月（判决书原文 CanLII 数据库访问受限，此处口径经 McCarthy Tétrault 律所评析转录核实）：“Air Canada did not take reasonable care to ensure its chatbot was accurate.”（第 13 章注⑥）

个人信息保护法：工作数据的处理依据与用途边界 它是什么：中国 2021 年公布的个人信息保护法律；第 6、13、14、17 条分别涉及必要范围、处理依据、同意与告知等要求。 书里拿它说明什么：第 8 章不将“公司工作产出”视为默认训练许可。具体用途、处理依据和权利边界须分别核对，不能用通用同意书替代判断。 可查线索：中国人大网，《中华人民共和国个人信息保护法》，2021 年 8 月 20 日，npc.gov.cn/npc/c2/c30834/202108/t20210820_313088.html。（第 8 章注③）

Gartner：2028 年一半的安全事故响应工作量 它是什么：Gartner 2026 年 3 月给出的预测——到 2028 年，企业网络安全事故响应的一半工作量，会集中在涉及定制开发的 AI 应用的事故上。 书里拿它说明什么：第 13 章提醒提前安排权限、审计和事故响应；限定的是响应工作量，不是事故数量，“涉及”不等于“由其引发”，定制开发不限定为企业内部自研。预测尚非经营结果。 可查线索：Gartner，《Gartner Predicts AI Applications Will Drive 50% of Cybersecurity Incident Response Efforts by 2028》，2026 年 3 月 17 日新闻稿，2026 年 9 月 6 日核官方正文，gartner.com/en/newsroom/press-releases/2026-03-17-gartner-predicts-ai-applications-will-drive-50-percent-of-cybersecurity-incident-response-efforts-by-2028。（第 13 章注④）

Indeed 招聘实验室与世界经济论坛《未来就业报告》 它们是什么：Indeed 2026 年 1 月报告记载，截至 2025 年 12 月美国数据与分析类招聘广告中约 45%

提到 AI 相关词；WEF《未来就业报告 2025》记录受访雇主对 2030 年岗位变化的预期。书里拿它们说明什么：第 6 章区分招聘广告观察、就业预期与具体岗位命运，不以词频证明岗位安全，也不将大数据专家增长预期改写成所有分析岗都将增长。可查线索：Cory Stahle，Indeed Hiring Lab，2026 年 1 月 22 日，hiringlab.indeed.com/2026/01/22/january-labor-market-update-jobs-mentioning-ai-are-growing-amid-broader-hiring-weakness/；World Economic Forum，《The Future of Jobs Report 2025》，第 2 章 Jobs outlook，weforum.org/publications/the-future-of-jobs-report-2025/in-full/2-jobs-outlook/。核查日期 2026 年 9 月 6 日。（第 6 章注①）

麦肯锡全球研究院：技能不是在消失，是在换个地方用 它是什么：两份关于工作未来的研究。2024 年那份给了一组数字：到 2030 年，社会情感类技能的需求，欧洲会涨百分之十一，美国会涨百分之十四。2025 年那份里还有一句更值得管理者看的话——今天的技能里超过七成，在自动化和非自动化的工作场景里都用得上。书里拿它说明什么：第 6 章用它支撑“技能不是在消失，是在换个地方用”这个判断。可查线索：麦肯锡全球研究院，《A New Future of Work: The Race to Deploy AI and Raise Skills in Europe and Beyond》，2024 年 5 月；《Agents, Robots, and Us: Skill Partnerships in the Age of AI》，2025 年。据官方页及媒体引文，第二份报告原文为 more than 70 percent。（第 6 章注③）

AFTERWORD

结语：把 AI 放回组织里

写到最后，我反而想把话说简单一点。这些年我坐过很多老板的对面，开场白几乎都一样：我们也在看 AI，就是不知道从哪儿下手。听多了以后我明白，“用不用 AI”这个问题其实早就过去了。真正的问题是：

> AI 进来以后，你的组织有没有能力接住它。

接不住，AI 就是个人外挂；接住了，AI 才可能变成组织能力。

这本书前面拆过五个断点，讲过人在回路中和组织整体在回路中，讲过 FDE 怎么在业务现场做翻译，讲过双元试点、产能分配、组织记忆和责任链，最后还回答了一个更大的问题——这些都做完以后，这家公司到底长成什么形状。合上书以后，这些名词你不必都记得。真正需要留在手边的只有一个问题：

> AI 做完这件事以后，组织有没有变得更稳定、更聪明、更能负责？

如果没有，先别急着把问题归给组织，也别拿工具使用量来交差。模型、数据、流程接口、资源和责任安排，都可能卡住结果。查清卡在哪里，才知道该改什么。

判断标准很土。你去问那条流程上的人：这一步现在归谁做，做完怎样检查，看出问题找谁改，改完的经验记在哪儿。再看这些安排能不能接住现在的任务，拿回约定结果。答案变了，不等于组织变强；答案没变，也不等于 AI 没进来。原安排仍然有效，就保留；哪处接不住，就改哪处。别为了证明转型，先把组织折腾一遍。

很多老板会在 AI 面前犯一个很自然的错误：看见速度，就想裁人；看见 Demo，就想上线；看见工具，就想推广；看见员工会用，就以为公司会用。这些反应都可以理解，但它们都太快了。

AI 释放速度以后，组织要问的是速度流向哪里。AI 生成结果以后，组织要问的是谁判断、谁复核、谁负责。AI 学走经验以后，组织要问的是员工为什么还愿意贡献真实业务上下文。AI 做出错误以后，组织要问的是这次错误有没有让系统变好，而不是有没有找到一个人背锅。这就是组织 OS。它不是一个概念，而是一套追问。

AI 时代最可怕的不是企业不学 AI。不学，至少还知道自己没开始；更可怕的是叶公好龙。嘴上说要 AI 转型，真到知识库治理、RAG、Agent 落地、流程重写、责任复核，就觉得这事很简单：找两个人，整理一下资料，上个系统，接个模型，就能完成。这不是轻视技术，这是轻视组织。

技术的进步很快，但企业里的具体门道，不会仅凭换一个模型就自动齐备。业务规则、历史判断、客户状态、员工经验、系统字段、责任边界、隐性流程，哪些有记录，哪些能接入，哪些获准使用，都要查。AI 可以帮助提取，不能把尚未取得的信息当成已经知道。

举个最小的例子。系统里有个客户等级字段，写着 A 类。老销售一看就知道这个 A 是三年前谈返点时给的面子，早就不作数了，报价照 B 类走。这条规则没写在任何文档里，只长在那个人脑子里。AI 读那个字段，读到的是 A。

问题不在这些经验是否永远只能靠人摘录。系统可以收集记录、识别冲突、提出规则，但读到一个字段，不等于取得了它背后的全部依据；发现一种惯例，也不等于获准照着做。哪些材料可用，怎样验证、授权和更新，组织要把这些安排接清楚。

所以我写这本书，不是想告诉企业“不要用 AI”。正好相反，我认为企业必须用 AI。但企业更必须明白：AI 不是来替你省掉组织管理的，AI 是来放大组织管理后果的。

流程清楚，AI 会让流程更快；知识沉淀，AI 会让知识更好调用；责任清楚，AI 会让复核和协作更稳定；信任机制清楚，员工才愿意把真实判断贡献给组织。反过来，如果流程乱、知识散、责任虚、绩效歪、信任薄，AI 也会把这些东西放大。它不会帮你绕过去，它会把问题照亮。

这就是为什么 AI 转型不能只交给技术部门，也不能只交给 HR，更不能只交给某个创新小组。技术部门能让系统跑起来，HR 能推动岗位、绩效和信任机

制，业务负责人知道真实流程和结果，一号位必须回答组织目标、能力承接和组织代价。

这些东西没有拉通，AI 项目就会在组织里变成很多碎片：一个部门一个工具，一个老板一个想法，一个员工一个外挂。看起来很热闹，但组织没有升级。

如果你是企业一号位，读完这本书以后，先别急着问“下一个 AI 工具是什么”。前面各章反复出现的那五个问题，此刻值得再过一遍。

五个不必一次答完，先答开头那个，剩下四个放到下次班子会。AI 正在改变哪些真实工作片段。这些变化有没有进入流程，还是停在个人电脑和 Demo 里。员工和 AI 一起打磨出来的好判断，有没有变成组织记忆。AI 的输出影响了结果以后，怎样复核、怎样验收、谁负责。授权、留痕、异常阻断和后果处置，有没有进入日常运营。

答不清，继续买工具只会让问题更复杂。答清以后，再谈工具、模型、Agent、自动化、裁员、增产。顺序很重要。AI 时代，速度很诱人，但组织改造不能只看速度。它要看结果能不能稳定，责任能不能兜底，知识能不能复用，信任能不能维持。

所以最后的建议很朴素：不要从买工具开始，从一条真实流程开始。

先挑一条。要高频，要痛点明确，要有稳定的输入输出，要能被衡量——报价、开票、售后工单、图纸会签，都可以拿来检查。然后把它拆开：哪一段是任务，哪一段是判断，哪一段是关系，哪一段是责任。拆完再问四句：AI 获准做什么，靠什么检查，出了问题谁有条件处理，经验怎么沉淀下来给下一个执行者用。

可以先约一个 90 天观察窗口，但别把日历当验收标准。具体多久，按任务、风险和结果出现的周期事前约定；明显越界，该停就停，不等到期。结束时，把结果、总投入、错误和处置记录摆出来，再决定沿用、调整还是停止。需要改的接口、分工和控制，要留下记录；原来就适用的安排，不必换一套名字。

流程改过、台账齐了，还不能宣布成功。取得了什么结果，付出了什么代价，后果能否承接，换人以后还能不能运行，这些要分别验。

对大多数做组织的人来说，让他去搞 AI，是为难他；对大多数做 AI 技术的人来说，让他去搞组织，也是为难他。

写书这件事，不能只靠句子漂亮。句子漂亮没用，要能解释真问题，要能给出真方法，要能让一个老板看完以后，少犯一个大错。我写这本书，是因为我越来越相信，这个时代需要有人把问题从“AI 怎么用”往前推一步：

当 AI 进入企业以后，组织到底要怎么改？

这个问题不是 HR 的小问题，不是 IT 的小问题，也不是某个工具厂商的交付问题。这是企业在 AI 时代能不能继续有效运行的问题。

如果这本书有一点价值，我希望它至少让读者记住三句话。第一，AI 转型不是买工具，是组织 OS 升级。第二，AI 替代的不是岗位，是任务、判断和流程片段；裁员不是起点，是产能分配后的结果。第三，人在回路中不是按钮，组织整体在回路中才是 AI 规模化之后的责任机制。

前进的生产力不会停下来等任何人。企业要么把 AI 放回组织里，要么被 AI 照出组织里的旧问题。

最后，再说得更白一点。AI 不会自动让一家企业变强，它只会让企业更像自己。强组织会被 AI 放大，弱组织也会被 AI 放大。

原来流程清楚、责任清楚、知识沉淀、信任机制稳定的公司，AI 进来以后，有机会把这些变成更高的产能。原来流程混乱、知识散落的地方，则要留意错误是否也跟着加速。同一套模型，同一个价钱，未必有同一种结果。技术和数据要查，组织怎样使用它们，也要查。

所以 AI 进入业务之后，组织要检查原来的安排是否仍然够用。检查不能省，重配不能先判。

这本书能给你的，到此为止。剩下的事，得在你那条流程上发生。AI 会照亮你的公司，问题只是它照出来的是能力，还是旧账。

这就是我想写清楚的事。