

AI-LED SEO • EVIDENCE TO ACTION

让 AI 主导 SEO 工作 让证据决定下一步

找词、研究、建站与反馈决策

一套可运行、可审查、可回退的实践指南

10 章方法与工程实践 · 7 幅原创流程图

配套本地合成数据原型与接口模拟测试

未验证真实流量、排名、收入或生产自主运行

目录

AI 主导的 SEO 工作流	6
从选词、建站到反馈迭代的实施指南.....	6
先知道这次交付能做什么.....	6
全书的六个工作阶段.....	6
怎样理解书里的数字与代码	7
01 从案例到可验证假设	8
先证明页面有用，再讨论做多少页	8
时长有用，但必须知道用户在做什么.....	8
收入要从正确的分母开始算	9
数据能读到，还要核对能否再发布	9
提交、索引、曝光，各自核验	9
小原型可以轻，生产系统必须算限制.....	10
第一轮怎样算有进展.....	10
02 让 AI 主导工作，让证据和权限决定能否执行	11
原型到底实现了哪一段	11
2.1 一条闭环，同时保留三个事实层	11
2.2 四个角色，各自少拿一点权限.....	13
2.3 用稳定的实体 ID 把证据串起来	13
2.4 指标有两种时间，还会被修订.....	14
2.5 批准是一张窄通行证	15
2.6 SQLite 先解决连续运行，不急着建平台	15
2.7 停止规则比更多自动化更重要.....	16
本章依据.....	16
03 找词与机会验证：先证明值得做，再让 AI 扩大生产	18
3.1 把“有人需要”与“值得由我来做”分开.....	18
3.2 先写一张需求卡	18
3.3 分三路收集候选，再做有限扩展	19
3.4 清洗时不要把不同需求揉成一个词	20
3.5 搜索量接口只回答它擅长的问题	21
3.6 区分稳定需求、季节需求和事件需求.....	21
3.7 固定 SERP 采样条件，才能比较页面机会	22
3.8 打开页面完成任务，别只比较标题与字数.....	22
3.9 AI 搜索改变的是任务竞争，别只追“被引用”	23
3.10 排序是安排工作，不是预测 Google	23

3.11 最终交给建站代理的任务卡	24
来源与核验边界	24
04 从资料到有用页面：先建立事实，再组织表达	26
4.1 先写用户任务，再决定要不要建页面	26
4.2 分类体系服务于发现与维护	27
4.3 资料进入系统之前，先明确使用边界	27
4.4 保存资料版本，再提取事实	29
4.5 事实冲突时，不要让模型投票决定	30
4.6 让 AI 写页面时，给它任务与证据，不只给关键词	31
4.7 多语言与地区：翻译之外，还要复核用途	32
4.8 发布之后，内容责任没有结束	32
本章的实现与验收	32
05 自动建站与收录：把审核过的内容变成可检查、可撤回的页面	33
5.1 为什么第一版选择静态 HTML	33
5.2 页面记录就是模板合同	34
5.3 从记录到可用网页的构建过程	34
5.4 页面必须先让人走得通	35
5.5 转换器要真的完成任务	35
5.6 把预览与公开索引分成两道门	36
5.7 四种容易混淆的控制	36
5.8 sitemap 的 lastmod 不能是每日签到	37
5.9 分页、筛选和站内搜索要各自设计	37
5.10 状态码由实际托管层兑现	38
5.11 国际 SEO 先确认是否真有个版本	38
5.12 结构化数据只描述页面上真实存在的东西	39
5.13 性能验收要区分本地测试与真实访问	39
5.14 从收录准备到收录观察	39
5.15 观测接口先定义，默认不采集	40
5.16 发布、回滚与失败处置清单	40
本章参考	41
06 AI 编排与接口接入：让建议可以检查，让执行有明确边界	43
6.1 实际交付了哪些能力	43
6.2 两份合同，分别服务分析与执行	44
6.3 从原始资料到建议，按这个顺序走	45
6.4 提示词怎样写，哪里不能依赖提示词	46
6.5 实际模型接口：显式开关、限制输入、严格拒绝异常	46
6.6 GSC、GA4 与收入数据不能揉成一行“表现”	47
6.7 DataForSEO：先管请求，再谈批量	47

6.8 预算、失败与重试的实际含义.....	47
6.9 接入与验收顺序	48
07 观测数据与反馈决策：知道该改什么，也知道什么时候先别动.....	49
7.1 每个数据源负责回答一个问题.....	49
7.2 先定事件，再安装统计	50
7.3 不要把三套分母拼成一个漂亮漏斗	50
7.4 日快照需要记录“数据版本”和“页面版本”	51
7.5 GSC：分页取完，不代表拿到全部搜索	52
7.6 GA4：用响应元数据保护判断	53
7.7 广告收入要与到账和成本分开.....	54
7.8 每天运行一次观测，不等于每天修改一次页面	54
7.9 把诊断规则写成有限决策，而非让模型自由找原因	55
7.10 冷却期、对照与回滚都要在发布前写好	55
7.11 提交、抓取、索引、排名分别记录	56
7.12 最低验收：让系统证明它会克制	56
一手来源.....	57
08 先算清一批页面，再决定要不要扩大	59
8.1 Page RPM 的分母是浏览量	59
8.2 一张能说明亏在哪里的月度账.....	59
8.3 每页成本里，最容易漏掉的是以后	60
8.4 广告接入有独立的验收门.....	61
8.5 不用牺牲完成任务来换收入	61
8.6 从 20—50 页试点开始，是资源建议	62
8.7 地区和语言扩展，先证明差异有用	62
8.8 学习要改进判断，不能偷偷改目标	63
本章依据.....	63
09 先跑通两个实验，再接真实世界	64
9.1 需要什么，运行后会得到什么.....	64
9.2 一次 demo 到底做了哪些事.....	65
9.3 三条命令及其边界.....	65
9.4 数据库如何把动作连成一条证据链	66
9.5 模型建议为什么不能直接变成写入	67
9.6 什么样的数据才允许作判断	68
9.7 故障演练：确认程序停在哪里.....	69
9.8 预算到底限制了什么	70
9.9 已实测部分与仍然缺少的部分.....	71
10 从本地演示走到可维护的真实站点	72
10.1 先划清交付边界	72

10.2 在干净目录复现结果 72

10.3 第一次接真实服务，先手动认清资源.....73

10.4 发布经过四道门 74

10.5 自动任务只做已获准的事.....75

10.6 用恢复演练证明备份可用 76

10.7 验收清单要包含拒绝和漂移 77

10.8 用 30／60／90 天安排交付，不安排排名 78

10.9 交给下一位维护者的最小包 78

本章依据.....79

附录 A：与前一本 SEO 教材的对应关系.....80

 需要随时间修订的两处知识 81

 生产化仍需补齐的连接 81

附录 B：参考资料索引.....82

 逐项资料.....82

 随书代码与设计材料.....84

 引用与验证边界 85

AI 主导的 SEO workflow

从选词、建站到反馈迭代的实施指南

核对日期：2026 年 10 月 14 日。主要对象：Google 自然搜索、独立内容与工具站。适合愿意自己维护一个小站、能审查资料与代码的个人开发者。

这本指南围绕一个问题展开：怎样把“发现需求、做出有用页面、观察结果、决定下一步”连成可以检查的工作流。AI 负责理解、归纳与提出方案；数据采集、权限、预算、发布和回滚需要可验证的程序与明确责任。

先知道这次交付能做什么

本书配套程序可以在不提供账号和密钥的情况下运行。它用合成数据生成本地静态站，完成两轮标题实验，分别演示保留和回滚；资料模块可以导入离线文件，检查事实候选是否具有真实引文。完整源码、测试、接口适配器与提示词随项目交付。

真实账户接口只做了模拟 HTTP 测试。初始关键词、正文和观测指标是教学夹具；没有开展真实搜索实验，没有真实模型调用，没有生产网站、收录结果或广告收入。全流程生产架构写明了怎样接入，但并未把这些尚未验证的环节包装成已经运行的自动化业务。

阅读目标	推荐顺序
先理解是否值得做	01 案例假设 → 03 找词 → 08 收益预算
先把程序跑起来	09 运行手册 → 02 数据合同 → 06 接口
负责内容与质量	04 资料与页面 → 05 建站 → 07 观测
准备接入真实站点	10 部署验收 → 06 接口 → 07 反馈

全书的六个工作阶段

1. 发现需求：选市场、收集词、查看结果页、识别用户尚未完成的任务。
2. 验证机会：核对数据、竞对、收入假设、资料权限与独立页面价值。
3. 生产页面：组织分类、建立事实依据、生成与审核内容、实现工具功能。
4. 构建与发布：生成站点、校验 SEO、预览审批、部署与抓取诊断。
5. 采集反馈：保留数据口径、版本、时间窗口和缺失状态，关联内容改动。
6. 决定下一轮：等待、保留、回滚、更新、合并或扩大样本，并记录理由。

这六个阶段对应十章详解。阶段不是一条只向前的流水线：来源许可不明会退回研究，页面没有独立价值会取消生成，数据不足会保持等待。正确停下也是系统应具备的能力。

怎样理解书里的数字与代码

标注“合成”“样例”“假设”的数字不是市场调查。样例阈值用于解释规则，不是平台要求，也不是适用于所有站点的统计结论。官方规范与接口口径在相关段落附链接；工程取舍由本书说明理由。无法核实的接口、价格或因果关系明确留作未知。

代码文件是实际执行行为的依据，章节里的短片段用于解释。完整项目中的 README 给出最短运行路径；第 09 章给出输出、异常和恢复方法。运行测试不等于生产验收，更不等于 SEO 效果证明。

正文使用 Readable Writing 组织：先说明问题与结论，再补机制、步骤、取舍和验证。图表示流程与结构，表格比较相同维度，关键限制跟在判断附近。篇幅由讲清内容决定。

01 从案例到可验证假设

这套工作流先解决一个具体问题：怎样让 AI 帮你找到值得做的页面、整理可靠资料、提出改进，再用真实结果决定是否扩大投入。第一轮的目标，是跑通一条能检查、能停下来的流程。

陈然的案例提供了一个出发点：用自动化经营免费内容站，再用运营记录指导迭代。文中的千页规模和每页月入一美元，应视为个人策略与理想情景，不能直接写进我们的验收标准。

以下规则依据截至 2026 年 10 月 14 日核验的官方资料；实验数量、阈值和产品设计建议属于本指南的选择，不是平台要求。

先证明页面有用，再讨论做多少页

一个页面需要回答：谁在什么条件下，要完成什么任务？现有资料缺在哪？这个页面补上了什么？如果只能回答“多覆盖一个关键词”，暂时不生成独立网址。

例如，在 Go 工具站里，“JSON 转 Go”与“CSV 转 Go”可能需要不同的解析、示例和错误处理，因此有独立用途。“免费 JSON 转 Go”和“好用的 JSON 转 Go”若提供同一功能，则更适合由同一页回答。先把组合当候选，再按真实支持能力、独特信息与需求证据筛选。

Google 的政策并不因为内容由 AI 生成就一律排斥它，但会处理以操纵排名为主要目的、大量制造低价值页面的行为。大量近似页只把人导向同一个最终入口，还可能触及网页滥用。因此，分类体系应表达真实差异，发布门槛必须能拦住仅替换词语的页面。[Google 垃圾内容政策](#)

时长有用，但必须知道用户在做什么

假设两个人都停留五分钟：一个成功排查了错误，一个始终找不到下载按钮。时长一样，结果不同。对于单位换算工具，迅速得到正确答案反而可能是好体验。这些是解释指标的假设例子，并非网站实测。

GA4 的互动时间主要记录网页处于焦点或应用位于前台的时间。参与会话则按持续时间、关键事件或页面浏览次数等条件判定。它们有明确测量含义，却不能单独证明内容正确，也没有构成“提高这个数即可保证排名”的官方承诺。[GA4 互动时间、参与率与跳出率](#)

每种页面先定义成功。例如工具页用“处理成功且输出通过校验”，教程用“读者能完成规定步骤”。时长、参与率和滚动深度用于帮助解释过程；错误率、投诉与加载失败用于防止优化伤害使用。事实准确性仍要单独审核，不能交给一个行为指标代替。[Google 内容自评指南](#)

收入要从正确的分母开始算

用页面 RPM 做情景预算时，计算关系是：页面浏览量 ÷ 1000 × 页面 RPM ≈ 估算广告收入。页面浏览量、访客人数、广告展示次数和互动小时不能互换。RPM 也会变化；它不是广告平台承诺的固定单价。[AdSense RPM 定义](#)

预算里分别记录保守、基准、较好三种情景，再扣除域名、托管、数据、模型、审校和维护成本。没有真实投放数据时，所有收入输入都标为假设。不能用表现最好的少数页面，外推整批页面的平均收入。

广告资格也要另行检查。AdSense 要求优质原创内容与政策合规，满足申请条件不等于通过审核。[申请条件](#) 上线广告前，应确认内容与广告可辨识、不会诱导或制造误点；测试者不能点自己的真实广告，机器人流量不能当收益来源。[实施政策](#)、[无效流量](#) 面向欧洲经济区、英国与瑞士用户投放个性化广告，还要核对认证 CMP 与 TCF 要求；认证不能代替完整法律核查。[同意管理要求](#)

数据能读到，还要核对能否再发布

每个数据源都保存来源网址、取数日期、许可依据、允许用途和更新责任。公开页面上的文字、图片和数据集不能统一当成免费素材；事实本身与受保护的表达也需要区分。[美国版权局说明](#)

政府资料同样逐项核查。美国联邦政府作品有特定版权规则，不能推广为“所有政府网站内容都无版权”。[美国版权法第 105 条](#) 资料若包含可识别个人的信息，还需要隐私审查。[ICO 个人信息指南](#) 许可不明或涉及不必要个人资料的记录，可以保留为研究线索，但本工作流不让它进入发布队列。

提交、索引、曝光，各自核验

普通内容页不能把 Google Indexing API 当通用收录按钮；它仅适用于规定的招聘或直播页面类型。[Google Indexing API 范围](#) 对普通页面，先检查可访问性、索引指令、规范网址、站内链接与 sitemap，再在相应站长工具中观察结果。

IndexNow 可以通知参与搜索引擎网址发生变化，但各引擎仍独立决定是否抓取、索引。系统应分别记录“通知已接受”和“索引已确认”，后者不能从前者自动推导。[IndexNow FAQ](#)

AI 搜索也一样需要实测。某次查询出现 AI 概览，不能推出今后没有人点击网站；某次没出现，也不能保证它一直不出现。记录查询、日期、地区、设备和结果形态，再判断用户是否仍需访问工具、核对来源或处理复杂条件。

截至本次核验，Google 新版指南要求页面已索引、具有摘要展示资格，并处于允许参与生成式 AI 功能的状态，仍不保证被展示。[新版指南](#) 参与控制默认允许，子资源可能继承父资源，应核对实际有效值。[参与控制](#) 新版生成式 AI 效果报告提供展现数据；不能据此编出 AI 专属点击率或收入。展现不足时报告可能不出现，导出中的零还可能来自界面的不可用值，要保存缺失说明。[报表口径](#)

小型原型可以轻，生产系统必须算限制

代码库适合保存代码、少量结构化样例和经过审核的配置。数据或媒体变大后，需要分别评估版本历史、构建、传输和存储。GitHub 普通 Git 会阻止超过 100 MiB 的文件；小文件很多也可能让仓库难以维护。[GitHub 大文件说明](#)

Vercel Hobby 限个人非商业用途，不能因为技术上能部署就默认适合广告站。[Hobby 规则](#) 不同部署路径另有上传、构建等限制；把资源放进 public 目录不能当作无限额度。[Vercel 限制](#)

因此，这次原型用本地样例验证流程。进入生产前再选择符合商业用途的托管方案，设定预算、数据保留和失败处理。

第一轮怎样算有进展

挑一个你能核验的用户任务，完成少量代表页：包括普通情况、边界情况和资料不足的情况。记录准备、生成、审校和返工的总成本。资料不足时，能正确阻止发布也是合格结果。

上线后的变化先称为观察。只有排除了足够多的其他解释，才讨论改动效果；低样本或缺失数据必须允许得出“目前不能判断”。把每次决策与页面版本、数据窗口、证据和回滚办法放在一起，AI 才有可用的反馈。它可以提出下一步，扩大范围仍要经过事实、用户价值与成本检查。

02 让 AI 主导工作，让证据和权限决定能否执行

这个系统的基本单位是一项可以追溯的页面改进：它为什么值得做，用了哪些资料，改了哪个版本，谁批准了什么，结果是否支持继续。关键词、文章和报表围绕这项改进连接起来。这样，即使某个模型换了，数据迟到了，或者一次实验没有效果，工作也能从可靠的状态继续。

本章先给出完整的目标设计。交付原型只实现其中的本地闭环；生产身份认证、多机调度和远程发布并未因为画进架构图就自动具备。阅读时按以下三个等级判断能力。

等级	本书中的含义	不能据此声称什么
原型已实现	源码中存在，可用本地合成数据执行并接受测试	真实流量、排名、收入已经验证
配置后可接入	有适配代码或清楚的接口合同，还需账户、授权、配额和真实验收	填入凭据就一定稳定可用
生产扩展未实现	给出工程设计与验收条件，需另行开发	交付包已经包含企业级平台

原型到底实现了哪一段

当前 CLI 只有 `demo`、`status`、`recover`，均要求专用 `--workdir`。`entities` 用不可变 JSON 记录关键词、分类、页面计划、来源、事实、提案、决策与机会反馈；`pages` 指向当前版本，`versions` 保留旧版本，另有 `metrics`、`approvals`、`actions`、`experiments`、`audit`、`meta` 表。精确 DDL 在 `store.py`。

可执行的页面改动目前仅为 `update_title`。动作状态从 `applied` 到 `kept` 或 `rolled_back`；实验从 `observing` 到对应终态。数据不足的 `wait` 仍保持观察，并记录原因。没有自动研究任意真实市场、长文自主生产、真实发布或模型自改代码。预设资料和固定模型回复演示数据流，不能称为完成了真实调研。

本地批准绑定提案 `hash`、页面、动作、旧版本、到期时间和 `synthetic-local-preview` 范围，并记录消费。默认策略的 4 次动作预算、7 天冷却和指标最低量属于合成演示初始规则，不能直接当作真实统计门槛。`STOP` 文件会拦截新的受控动作；它不是跨机器、跨进程的远程紧急停机平台。

2.1 一条闭环，同时保留三个事实层

先区分三层事实。第一层是输入证据，例如需求原句、资料版本、搜索报告响应。第二层是系统动作，例如生成草稿、批准差异、构建预览、恢复旧版本。第三层是外部结果，例如 Google 是否索引、真实访客是否完成任务、广告收入是否结算。原型可以证明第二层按规则运行，不能用它的合成报表证明第三层成功。

建议采用下面的工作图。图中的状态描述业务阶段，不等于数据库枚举；代码的精确字段和状态以第 9 章及 `prototype/seo_loop` 为准。失败可以进入“待补证据”或“待人工决定”，没有必要把任何阻塞都当作整个项目失败。

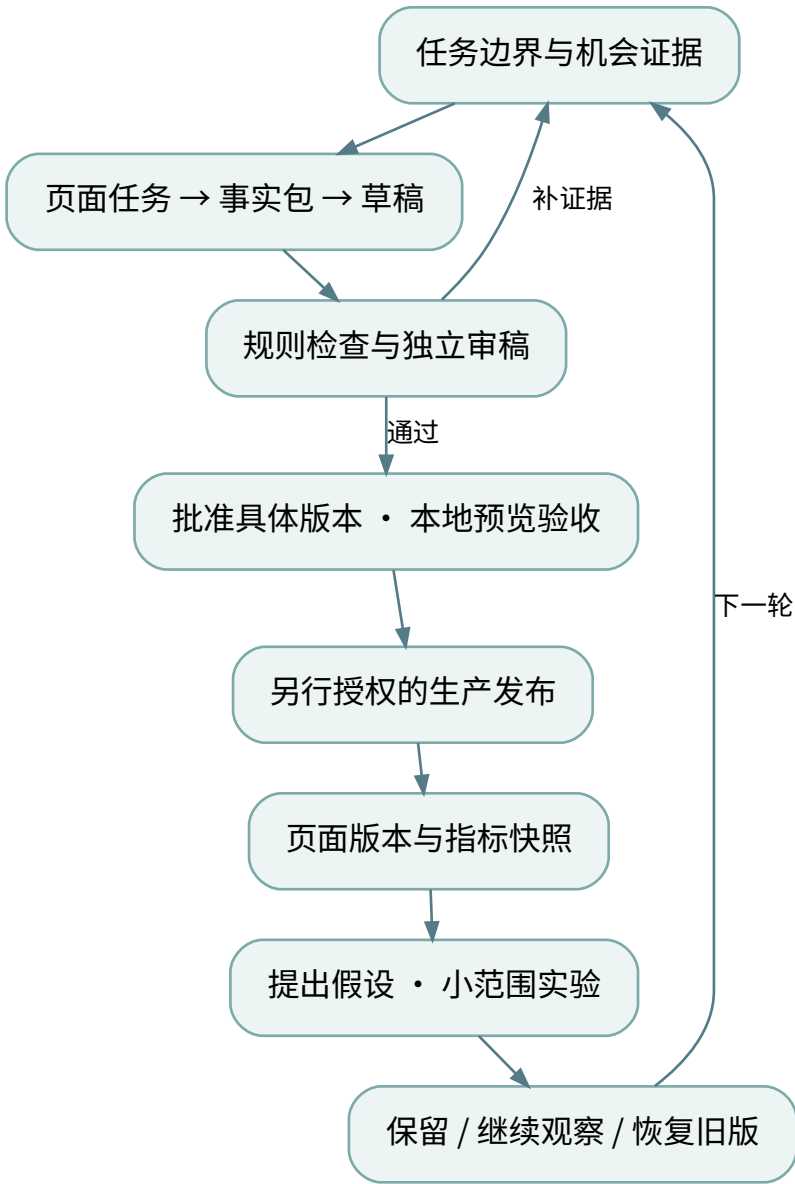


图 1 · 从任务证据到小范围实验的目标工作流。示意图；实施范围以本章正文为准。

实际工作还需要一条不依赖搜索结果的快速反馈：草稿能不能让人完成任务，工具算得对不对，读者不会误解输入条件。这些问题可以在发布前验证。只有涉及真实搜索处理、访问组成和收入的假设，才必须等待相应外部证据。不要让“SEO 需要时间”成为不修功能错误的理由。

2.2 四个角色，各自少拿一点权限

规划者负责提出下一步：哪个需求值得调查，哪个证据缺口阻碍判断，怎样用小改动验证假设。它收到的是结构化资料与预算，不拥有生产写权限。写作者根据事实包生成候选正文和变更单，可以标记未知，不能把未知填成看似确定的数据。

审稿者独立检查来源与结论之间的关系，提出最强的反对意见。它与写作者分开运行，有助于暴露遗漏，但两次模型判断并不构成独立事实证明。同一个错误来源可能使两个模型一起出错。程序只检查可机械验证的事项，例如必填字段、证据 ID 存在、URL 范围、成本上限和版本散列。语义真实性、原创价值及法律适用问题，仍需相应人工复核。

最后，确定性执行器只接收白名单动作及通过校验的参数。它不接收“照模型说的运行以下 shell”。规划者即使输出合法 JSON，也可能提出危险动作，所以格式验证之后还要检查业务范围、当前版本、审批和预算。这种分层控制与 OWASP 对提示注入的工具侧校验、最小权限建议一致。[A02]

角色	可以交付	不应拥有的能力
规划者	候选任务、证据需求、实验假设	发广告、改权限、发布生产
写作者	基于证据的草稿和差异	执行任意代码、创造来源
审稿者	问题列表、严重级别、拒绝理由	自己批准自己生成的高风险修改
执行器	校验后的本地版本、预览、审计记录	越过动作白名单解释自然语言
人工负责人	事实确认、具体审批、预算和方向决定	用一次笼统许可替代后续重大变更审批

小站可以由同一个人承担全部人工职责，但系统中仍应分开“建议”“审核”“批准”“执行”的记录。外部网页、SERP 摘要、API 返回和文档里的文字都是资料，不能成为指令来源。

2.3 用稳定的实体 ID 把证据串起来

下面是生产目标数据合同，表示每类对象至少应能回答什么。原型可以把其中一部分存为 JSON；这不意味着全部列都已经实现。新增字段前先做迁移和兼容测试，不能只改文档。

对象	必须回答的问题	建议保留的关键属性
需求／关键词	谁在什么条件下要完成什么	稳定 ID、原句、语言地区、来源、时间、状态
来源／证据	谁说的，何时取得，是否可再用	来源 ID、URL、标题、取得时间、内容 hash、权利说明

对象	必须回答的问题	建议保留的关键属性
实体／事实	说的是哪个对象，结论适用于哪版	实体 ID、断言、适用版本、证据 ID、失效条件
页面	这个地址长期承担什么任务	page ID、任务、语言、主题、规范身份、负责人
页面版本	这一版到底写了什么	version ID、内容 hash、父版本、证据集合、生成配置
指标快照	何时发生、何时看到、后来是否修正	指标、维度、时区、事件区间、观测时间、revision
决策／实验	为什么改变，怎样判断是否值得	假设、主要指标、护栏、比较方法、状态、结论
动作／批准	允许谁对什么做哪一步	动作 ID、目标、操作、差异 hash、有效期、使用记录

不要拿标题当页面主键。标题会改，slug 也会迁移；页面 ID 应在这些变化中保持稳定。也不要“主题”“关键词”“页面”当同一种对象。多个表达可能指向同一个任务，一个主题可能需要工具、教程和排错页分别承载。

建议生产中另设 URL 映射：原始 URL、清洗 URL、声明的 canonical、搜索系统选择的 canonical、内部 page ID、映射生效时间。它们不能互相覆盖。开发者声明的 canonical 是一种信号，并不等同于 Google 最终选定的规范页。[A03] 参数是否删除、路径大小写是否合并，要按站点路由测试决定；不能用全站字符串替换把不同语言或产品版本合成一页。

来源记录要能复核，但不能无限复制

证据包优先保存来源地址、取得时间、必要短摘录、解析后的事实、响应元数据和 hash。只有确有权利且确有必要，才留原始全文或完整 API 响应；设置保留期和访问范围。完整响应可能包含查询文本、个人信息或商业数据，不能因为叫“审计”就永久公开保存。

公开可读不等于允许重新发布。美国版权局区分事实与其受保护的表达，并明确网站原作可能受版权保护；具体项目还需核对适用地区、许可与合同。[A04] 内容 hash 用来判断保存的字节是否变化，不能证明来源真实、观点正确或素材有授权。

2.4 指标有两种时间，还会被修订

指标记录至少要区分“被统计的日期”和“我们拿到它的时间”。例如 10 月 1 日的点击数据，10 月 4 日导出一次，10 月 8 日再导出一次，数值可能变化。不要覆盖第一次观察；新增 revision，说明当前报表采用哪一版。正式状态与仍在补全的状态也应分别保存。Search Analytics 提供 dataState 等字段，报告还存在行数与数据覆盖限制；适配器必须把这些限制传到决策层。[A05]

一条生产指标身份建议由以下内容共同确定：数据源、资源 ID、查询规格 hash、维度集合、统计窗口、指标名、单位。observed_at 与 revision 区分同一身份的多次观测。币种和时区属于口径，不能只写进备注。缺失写 unknown 或 blocked；只有确认测得零才写 0。

版本与指标也要明确连接。一个页面在窗口中间更新，整窗收入不能毫无说明地全归给新版本。可拆窗口、记录暴露时间，或把该次观察降级为混合版本观察。流量不足时结论就是“无法判断”，而非要求模型填一个增长原因。

2.5 批准是一张窄通行证

“可以继续优化”不足以授权一次生产改动。更安全的批准单绑定操作、目标、版本和有效期。下面是生产设计示意，字段名不冒充原型 API：

```
{
  "operation": "publish_version",
  "target_page_ids": ["page-time-converter"],
  "base_version_hash": "sha256:旧版本散列",
  "candidate_hash": "sha256:待发布内容散列",
  "environment": "production",
  "expires_at": "2026-11-01T10:00:00Z",
  "max_uses": 1,
  "approved_by": "已认证的负责人身份"
}
```

执行前重新取得当前版本。若它与 base_version_hash 不同，停止并重新出差异；旧批准不能覆盖别人刚做的修正。批准范围只包含列出的页面，不会顺便覆盖全站模板、robots、canonical、广告代码或追踪脚本。执行器将“验证批准、扣减可用次数、登记动作”放入受控事务，并在外部副作用后进行状态核对。

本地演示的批准记录用于检验程序拒绝越权和版本漂移；它不是可用于互联网服务的身份认证。生产扩展还需可信登录、角色权限、不可由模型写入的批准通道、重放防护和撤销。即使有审批表，能直接改 SQLite 文件的人仍可能改记录；本地原型不能抵抗宿主机管理员篡改。

2.6 SQLite 先解决连续运行，不急着建平台

原型选择 SQLite，目的是把重跑、版本、动作和证据放到同一份本地状态中。SQLite 适合嵌入式本地应用，但同一时刻只允许一个写者；需要跨机器并发写或高写入负载时，应重新评估架构，而非直接把数据库文件放到共享网络盘。[A01]

第一阶段让一个受控进程负责写入，模型调用与网络等待尽量不占数据库事务。每个工作项有幂等键：同一证据、同一输入版本、同一规则版本和同一动作重复到达，不应重复创建页面或记两笔收入。幂等并不意味着“运行第二次完全没有新记录”；可追加一次运行日志，但业务副作用只发生一次。

生产扩展可逐步拆成 PostgreSQL 管理事务与权限、对象存储保存获准的响应和构建产物、任务队列管理延迟与重试。迁移理由应来自实测的锁等待、运行时间、数据量、恢复要求及多人操作需求。增加这些服务会带来账户、备份、密钥、升级与故障恢复成本，不是站点页面达到某个固定数字就必须升级。

任务队列的目标应采用“允许重复投递、执行幂等”的设计，不宣传端到端恰好一次。工作者取得有期限的租约，崩溃后允许重新领取；重试前查询动作结果。数据库写完而外部发布结果未知时，先对照发布清单和目标状态，不能直接再发一次。原型没有远程发布，所以还未验证这种分布式故障。

2.7 停止规则比更多自动化更重要

每次任务先预留预算，再发起调用，最后记录实际费用和差额。调用费、请求次数、运行时间、页面数量、并发数分别设上限。价格未知时不能用“0 元”通过预算门；付费 live 模式应拒绝执行，等待负责人补齐报价与限额。预算数值是运营策略，不是模型可靠性的保证。

故障或风险	原则	生产需要补齐的机制
临时超时、可重试服务错误	有界退避，尊重服务限流指示	指数退避加抖动、总截止时间、费用累加
权限失效、结构不符、证据冲突	不用重试掩盖问题	隔离记录、负责人处理、明确解封条件
连续失败	暂停该来源，别拖垮全系统	熔断器及有限探测恢复
多轮模型修稿仍失败	停在问题上，不无限循环	最大轮数、失败样本、死信队列
刚改完又想改	等观察窗口或处理明确故障	页面级冷却期，紧急修复独立批准
全局异常或重大投诉	停止新副作用，保留诊断	kill-switch、权限撤销、已在途动作核查

这张表是完整运营设计；哪些已落实为代码，以第 9 章测试清单逐项确认。死信队列的作用是保存不能自动完成的工作及原因，不是悄悄丢弃。恢复后重新提交也要沿用原业务身份，避免重复动作。

备份要把数据库、必要证据、构建产物和配置版本一起考虑。SQLite 提供在线备份 API；Python 的 sqlite3 封装可用于生成一致备份。[A06] 不能把“成功复制一个文件”当作恢复成功。第 10 章要求在另一路径恢复、校验完整性、复建页面并核对版本。生产 PostgreSQL 还需根据恢复目标选择备份和恢复方案，不能照搬 SQLite 文件复制。[A07]

本章依据

- [A01] SQLite：适用场景与并发边界，访问 2026-10-14。
- [A02] OWASP：LLM 提示注入防护，工具侧参数与权限控制。
- [A03] Google：规范 URL，声明与选择的区别。

- [A04] 美国版权局：保护范围，访问 2026-10-14；不替代当地法律意见。
- [A05] Search Analytics query，时间、状态及覆盖边界。
- [A06] SQLite 备份 API、Python 3.12 sqlite3。
- [A07] PostgreSQL：备份与恢复，访问 2026-10-14。
- 实现依据：本包 `prototype/seo_loop/`、`prototype/tests/` 及第 9 章的实际执行记录。未标为“原型已实现”的结构均是本文设计合同。

03 找词与机会验证：先证明值得做，再让 AI 扩大生产

你可以让 AI 一次生成一万条关键词，但生成速度不能回答最重要的问题：谁会用这个页面，他要完成什么，现有页面哪里不够，以及你能否持续交付。找词阶段的结果应是一小批有证据的页面任务，允许大部分候选停在“还不确定”。本章沿用开发工具站场景。所有评分、样本数量和演算数字均为教学设计，没有购买搜索数据，也没有真实站点增长结果。

3.1 把“有人需要”与“值得由我来做”分开

给每条候选保留四道独立检查，别用一个总分替代它们。

检查	要回答的问题	可以接受的证据	不通过时的动作
需求存在	谁在什么场景下遇到什么问题？	自有查询、经授权的访谈、公开问题、工具估计	保留待验证，不生成正式页面
意图匹配	搜索者要读解释、运行工具、比较产品还是购买？	同市场 SERP 样本与具体用户任务	调整页面形式，必要时拆分任务
交付价值	你的页面能让用户多完成哪一步？	可运行原型、原创验证、可靠数据、明确服务能力	先补产品或证据；更长的文章不自动过关
经营可行	成本、风险、维护与收入路径能否承受？	成本预算、实际收费条件、更新责任人	缩小范围、延后或放弃

例如“JSON 转 Go”可能有明确任务，但你只有一篇解释 JSON 是什么的文章，交付检查仍不通过。“JSON null 与零值”的搜索量即使未知，只要真实用户在这里反复出错，它仍可能是重要的帮助页；这不等于证明它是高流量获客机会。

搜索量未知不等于零；有搜索量也不等于适合立项。这两句话分别保护需求研究和投资判断。

3.2 先写一张需求卡

输入是业务边界、目标市场、已有能力和不能承担的风险。输出是一句可以检验的任务描述。以开发工具为例：

使用 Go 的开发者的，需要把接口样本变成可复核的类型定义；样本可能敏感，希望在设备内完成；面对空数组或不一致字段时，需要知道工具做了什么假设。

这张卡至少保存 user_role、task、context、constraints、desired_output、evidence_refs 和 unknowns。AI 可以整理原话、发现遗漏的角色、提出替代表述，但不能替用户承诺愿意付费，也不能把自己生成的任务当成访谈发现。若输入来自客服或日志，先去掉邮箱、令牌、客户名和原始业务载荷。

需求卡验收不看文字是否漂亮。让一个不参与讨论的人读完后回答“用户最后会拿到什么”和“怎样知道完成了”，答不出就继续改。

3.3 分三路收集候选，再做有限扩展

第一路是自有证据：GSC 查询、站内搜索、支持问题和合法取得的任务反馈。已有站点优先找“有相关展现但页面不满足任务”的位置。GSC 查询明细受隐私过滤与行数限制影响，某个词没有出现不能证明需求不存在。[K01]

第二路是公开证据：产品手册、开源项目问题、标准文档、公开社区与搜索结果。记录原始链接和日期；社区里的集中抱怨只能证明这些发言存在，不能直接换算市场比例。

第三路才是工具扩展与 AI 提议。让模型围绕任务生成同义表达、使用限制、错误场景、替代方案与比较问题。每条新词默认 evidence_status=unverified。禁止模型填搜索量、排名、竞对流量或“用户喜欢”等未经测量的字段。

起步可以每个任务留五至十个不同表达，合计几十条，而非一开始就做全排列。这个数量是节省研究成本的工作安排，不代表统计覆盖。每扩展一轮，都问新增候选是否让你发现新的任务；若只是同义改写，停止扩展。



图 2 · 机会验证先于页面生产。示意图；实施范围以本章正文为准。

3.4 清洗时不要把不同需求揉成一个词

保留 raw_query 和 normalized_query 两列，标准化空格、明显重复与大小写，同时留下转换版本。技术符号、产品版本、转换方向、城市、单位和否定词不应默认删除。“JSON 转 Go”和“Go 转 JSON”方向相反；“在线”和“离线”可能对应不同隐私约束；“免费”和“企业批量”可能对应不同交付能力。

聚类应先按任务，再参考表达相似度和搜索结果重合。语义嵌入能帮助找候选邻居，不能独立决定网址。同一个工具可以承接多个近义词；一个宽泛词也可能包含多个意图。让 AI 提供 cluster_reason 与 uncertain_pairs，由审核者处理不确定项。

输出一张“任务→页面”映射表。每个计划网址有一个主要任务，相关查询是辅助证据。新候选优先映射已有页面，只有任务或交付形式确实不同才新建网址。这一步能提前减少多个页面重复争取同一需求的情况。

3.5 搜索量接口只回答它擅长的问題

可选实现使用 DataForSEO 的 Google Ads Search Volume Live 端点。它返回搜索量、逐月搜索、广告竞争与相关字段。请求必须明确地区、语言与是否包含搜索合作伙伴，保存请求原文和供应商响应。该端点目前每个请求最多 1,000 个关键词；专用 Google Ads Live 限制为每账户每分钟 12 次。不要把文档同页出现的通用调用上限当成这个端点的有效上限。[K02]

本项目没有发起付费请求。接入时先检查账户资格和当前计费，再由已批准的预算控制器执行。价格不是代码常量：在请求账本中保存预计费用、实际返回 cost、币种和任务 ID。返回 HTTP 200 之后还必须检查顶层与每个 task 的业务状态；部分任务失败不能作为全批成功入库。

字段	应保存的含义	容易造成的错误
search_volume	供应商在指定口径下的估计月搜索次数，可空	当成独立人数或你会得到的访问
monthly_searches	逐年月的估计序列，可有缺口	月份缺失被程序补成 0
competition / competition_index	广告竞争相关字段	改名为“自然 SEO 难度”
cpc	供应商定义的广告 CPC 估计，保留币种口径	当成 AdSense 每次点击实际收入
location_code / language_code	本次市场设置	不填地区后把全球量当本地量
observed_at / requested_period	采集时间与数据覆盖期	把今天下载的数据说成今天发生的搜索
value_state	observed、unknown、not_applicable 或 error	用一个数值 0 表示所有情况

Google Ads 的平均月搜索量受到时间、地区、网络与近似变体口径影响；广告竞争不能替代自然搜索竞争判断。[K03] 对同义词簇不要简单相加搜索量，可能重复包含近似变体。保留每词原值，再把“主题可达需求”列为单独估算，明确计算假设。

没有预算时也能推进：先用真实问题与免费公开资料确定任务，手工核验少量结果，输出“搜索量未知”的原型计划。不要因为缺少付费数字就让模型制造数字补齐表格。

3.6 区分稳定需求、季节需求和事件需求

逐月数据比一个平均数有用。检查峰值是否反复出现在相似月份、是否只有一次突然上升、是否与产品版本或外部事件一致。至少把样本覆盖期写出来；只有几个月时，不宣称已经证明年度季节性。

示例中某词十一月为 1,300 次，其他十一个月均为 100 次，年总计 2,400，月均 200。这是教学算式。按月均安排服务器与内容更新会漏掉峰值，按峰值推算全年收入又会夸大机会。季节页面应在需求到来前完成事实更新与功能验证，旺季结束后不必为制造“新鲜”反复改标题。

Google Trends 可作方向检查，但它使用归一化的相对兴趣，低量可能显示零。不同地区、时间范围或搜索类型的指数不能直接当绝对查询次数相加。[K04] 把“稳定/季节/事件/未知”作为人工可复核标签，附上曲线和解释，避免只凭模型看图给定论。

3.7 固定 SERP 采样条件，才能比较页面机会

每次搜索结果采样保存 query、engine、location、language、device、timestamp、provider、depth 和 raw_response_ref。固定移动端或桌面端，以及目标国家或城市；不要混用中国用户的中文需求、美国英文数据与全球搜索量。语言、用户所在地和页面服务范围是独立维度。

付费采样可使用 DataForSEO Google Organic Live Advanced；其结果含自然结果和额外模块，device 支持 desktop 或 mobile。depth 提高可能增加计费，某些 AI Overview 采集选项另收费，必须在预算中批准，不能为“收全数据”自动打开。[K05] 供应商 SERP 是某次采集的观察，不保证和所有真实用户一致。

起步按任务抽样：泛任务词、明确限制词、问题词、比较词各取代表，而不是只采最大词。可先用每组两三个词、前十个自然结果，并在决策重要或波动明显时复查。数量和重查频率是成本与覆盖的取舍，不是行业统一标准。

记录时把广告、自然结果、地图、视频、精选摘要与 AI 概览分开。natural_rank 与 absolute_position 不能混用；“第三条自然结果”不一定在首屏。若只通过普通网页搜索发现候选网址，标签写“候选页面发现”，不要声称已经测得指定地区 Google 排名。

3.8 打开页面完成任务，别只比较标题与字数

搜索竞对是与目标查询同场出现的页面，任务替代品则可能是本地脚本、编辑器插件或官方文档。它们都值得研究。某网站很大，并不意味着整个域名都是与你有关的比较对象。

每个核心任务选少量代表页面，按相同测试卡观察：进入页面后能否找到入口，正常输入能否得到正确输出，边界输入如何提示，是否需要注册或上传，手机是否可操作，结果能否复制，以及限制是否说清。使用自造的无敏感样本；不上传真实客户数据，不产生付费订单或打扰对方服务。

证据状态	可以写什么	不能写什么
只阅读页面文案	页面宣称支持批量处理	已验证批量处理可靠
完成正常样本测试	此样本在此时间得到该输出	所有输入都正确
完成边界测试	空数组样本未得到明确提示	竞争者没有任何错误处理
没找到某功能	此次观察未找到入口	全市场不存在该功能

差距必须落到用户能多完成的动作。例如“给空数组的不确定类型加标记，并允许用户选择类型”，比“写一篇更全面的 JSON 教程”具体。先承认已有页面解决得好的部分，避免为了立项把竞对描述得一无是处。

此前 SEO 教材中的 JSONLint 与 mholt/json-to-go 阅读只能作为待复查候选，不能继承为今天的实测结论。本章未新增竞对功能测试；实际研究时应保存当前页面、测试输入、输出与时间。

3.9 AI 搜索改变的是任务竞争，别只追“被引用”

能在一句话内充分回答的定义问题，可能在结果页或 AI 回答里就结束。工具站更应研究仍需要交互的任务：处理用户自己的输入、调整条件、检查结果、保存对比或持续监控。这里的判断是产品假设，不是“工具一定不会被替代”的保证。

分别记录三种结果：搜索结果存在需求、AI 回答引用了页面、用户实际来站完成任务。三者不能互相替代。Google 说明 AI 搜索功能沿用基础 SEO 要求，没有额外必需的特殊标记。[K06] 测量能力已发生变化：2026 年 6 月的官方公告介绍了专门的生成式 AI 可见性报告，并注明 8 月 31 日已向全球网站推出，包含展现、页面、国家、设备及日期等观察。[K07] 这些可见性数据与普通搜索总览并存，不能把“有 AI 展现”直接换算为 AI 点击、用户或收入。本指南没有登录账户核验，也未核实该专门报告的自动化接口；不能凭 UI 功能名称给 Search Analytics API 编造一个 type。

因此，不设置“AI 引用率 80%”这种没有稳定分母的目标。可以在固定问题集、引擎、模型、地区与采集日期下记录引用观察，同时把任务完成和真实业务结果作为另一个测量层。单次未引用也不是页面质量失败的充分证据。

3.10 排序是安排工作，不是预测 Google

通过基本门槛后，才用评分安排有限工时。一个可解释的教学方案是：需求证据 25%、交付差异 30%、业务匹配 20%、可实施性 15%、维护可持续性 10%，各项 0–4 分。总分为权重乘评分之和，再换算为百分制。

这些权重是启发式，没有经历史样本验证，也不是成功概率或排名模型。缺失字段不自动记零：输出 `score_status=incomplete`，列出缺少的证据。若需要先排队，可以提供区间或暂列“研究优先”，但不能静默用平均值填空。

举例：某任务需求证据为 2、交付差异 3、业务匹配 4、可实施性 3、维护性 2，则加权分为 $2.85/4$ ，即 71.25。数字只表示在这套自定规则下的排序；另一个任务即便只有 60 分，也可能因为修复核心错误而必须先做。安全、合法性、功能真实性与预算是门槛，不能靠其他高分抵消。

每轮只批准可交付的一小批页面。记录“不做”的理由，包括已有页面足够、缺少可靠数据、无法持续维护、目标用户不匹配或风险超出能力。保留这些记录，能防止下一轮 AI 又把同一方向包装成新机会。

3.11 最终交给建站代理的任务卡

每张卡包含：任务 ID；目标用户与市场；主要任务；候选词及证据链接；指定页面或新建理由；SERP 采样条件；已验证的竞对差距；功能与内容要求；不支持的范围；事实来源；验收样本；预计建设及每月维护成本；负责人；审核状态。

假设任务卡要求做“JSON 空数组类型提醒”，验收要具体到：输入空数组时不静默编造确定类型；结果旁说明推断限制；用户能手工选择类型；说明页与真实实现一致；埋点只发送成功、错误类别和版本号，不发送 JSON 原文。满足这些条件之后，页面才有资格进入发布流程。是否获得流量，留给上线后的数据回答。

本章完成检查

- 每条立项任务有来源与未知项，模型提议没有伪装成真实需求。
- 地区、语言、设备、时间与搜索网络口径可以复查。
- 未知、零值、缺行和调用失败分别存储。
- 广告竞争、CPC 与自然排名难度没有混用。
- 竞对文案阅读与功能测试分开，差距有实际用户价值。
- 相同任务优先复用页面，季节需求有维护计划。
- 分数只用于排序，最终任务卡有可执行验收与成本上限。

来源与核验边界

资料核验日期：2026-10-14。下面是本章关键事实的一手来源；工作流程、任务卡、评分与例子是本指南的设计建议。未注册账户、未读取私人分析数据、未付费调用接口。

- [K01] Google, Search Console performance data filtering and limits: <https://developers.google.com/search/blog/2022/10/performance-data-deep-dive>
- [K02] DataForSEO, Google Ads Search Volume Live: https://docs.dataforseo.com/v3/keywords_data/google_ads/search_volume/live/
- [K03] Google Ads, About Keyword Planner forecasts: <https://support.google.com/google-ads/answer/3022575?hl=en>
- [K04] Google Trends, FAQ about Google Trends data: <https://support.google.com/trends/answer/4365533?hl=en>
- [K05] DataForSEO, Google Organic Live Advanced: <https://docs.dataforseo.com/v3/serp/google/organic/live/advanced/>

- [K06] Google Search Central, AI features and your website: <https://developers.google.com/search/docs/appearance/ai-features>
- [K07] Google, Introducing Search Generative AI performance reports in Search Console: <https://developers.google.com/search/blog/2026/06/gen-ai-performance-reports>

04 从资料到有用页面：先建立事实，再组织表达

一个关键词只说明有人在问，不说明我们已经有资格回答。建站流水线最容易漏掉的环节，就在这两者之间：页面里的事实从哪里来，读者能用它完成什么，资料变化之后谁来修正。

本章把这个环节拆开。示例围绕开发者的时间戳转换任务展开，避免用未经核实的医疗、金融或法律内容试验自动发布。项目提供了可运行的离线资料导入和引文校验模块；联网采集、完整事实审核和自动内容生产仍属于需要接入的生产流程。二者不能混为一谈。

4.1 先写用户任务，再决定要不要建页面

假设候选词是“Unix timestamp milliseconds to date”。一个合理任务是：用户拿到日志中的数字，需要判断单位，把它转换成可读时间，并避免误把 UTC 当成本地时间。页面的核心交付应当是转换结果、单位和时区说明，以及输入错误时的提示。长篇介绍 Unix 历史不能代替这个交付。

把页面计划写成一张验收卡，比先让模型写两千字更有效。

字段	示例	验收办法
用户遇到的问题	不知道日志时间戳是秒还是毫秒	输入框显式选择单位，不用不透明猜测
页面承诺	转换为 UTC 时间，并解释单位	用固定输入得到可复现结果
独有价值	本地计算、清晰报错、无需上传日志	断网仍可运行，源码中没有输入上报
内容依据	语言日期接口文档及已验证测试	记录链接、版本与测试命令
不适用范围	不能凭一个数字确定原系统时区语义	在结果附近说明限制
更新责任	工具维护者	依赖或算法变化触发回归测试

这里的“独有”不意味着世界上只有你做过，而是说这张页面确实提供了可辨认的使用价值。Google 对有用内容的说明强调内容是否让读者达成目标，而不是是否达到了某个字数。[Google: 创建有用、可靠、以人为本的内容](#)

怎样判断两个词要不要拆成两页

先问三个问题：回答是否不同，数据是否不同，用户操作是否不同。只有词面变化、实际答案相同，通常先合并成一页的不同小节。反过来，日期格式校验与时间戳转换有不同输入和算法，分别设计工具页就有理由。

“英语关键词 × 中文关键词 × 城市名 × 年份”不能自动成为页面清单。地区维度只有在规则、供应、数据或用户任务确实发生变化时才有意义。开发者时间戳工具通常没有“香港版”和“深圳版”的独立答案；当地交通时刻表则可能有。这个判断必须先于模板渲染。

建议给每个候选页留下四项证据：目标任务、与已有页的区别、所需数据、可检验的完成条件。缺任一项，状态停在草稿。模型可以提议拆分，人或规则负责拒绝没有差异的组合。

4.2 分类体系服务于发现与维护

分类体系就是用稳定的概念组织页面。它同时影响导航、内链、内容责任和数据刷新。把所有关键词按向量相似度分组，只解决了其中一部分；两个词表达相近，也可能分别指向工具操作和概念解释。

本项目可以从三层结构起步：

- 主题：时间与日期。
- 任务：转换、校验、排查。
- 页面：时间戳转换器、单位说明、无效日期排查。

不要为了树形结构强迫一个页面只属于一个用途。页面拥有一个稳定主路径，同时可以通过相关任务关系被发现。例如单位说明既能由转换器链接，也能由日志排查指南链接。页面的身份用稳定 ID 表示，URL 是可调整的展示位置；路径变化需要重定向和链接检查，不能让历史指标跟着丢失。

关系	为什么记录	错误用法
页面属于主题	生成目录与责任分工	每增加一个标签就生成可索引空列表
页面解释另一个页面的概念	建立有帮助的内链	每页都插入全部站点链接
页面依赖事实	上游数据变更时找到受影响页	只保存最终 HTML，不保留依据
页面替代旧页面	保持访问与指标连续性	把所有删除页重定向到首页
翻译对应页面	核查不同语言是否等价	把任意首页配成另一语言版本

这是本指南建议的数据设计，不是 Google 规定的数据库模式。设计好坏最终要由维护成本、查找效率和页面质量检验。

4.3 资料进入系统之前，先明确使用边界

公开可访问不等于可以任意复制。采集许可、访问规则、内容版权、个人信息和对外发布权限，是不同的问题。robots.txt 约定爬虫行为，不提供访问授权；RFC 9309 明确将二者区分。[RFC 9309](#)

生产系统应为每个数据源保存如下记录。表内是建议合同，离线原型只实现其中一部分，见本节末尾。

字段	应记录什么
source_id	内部稳定来源标识
source_url	可回查的原始页面地址，去除令牌等秘密参数
owner_and_license	发布者、许可链接、允许的具体用途
access_method	API、人工导出或有授权的抓取
retrieved_at	实际获取时间及时区
effective_at	内容所述规则的生效时间；未知就留空
content_hash	对获取版本计算摘要
allowed_fields	可以提取和保存的字段
freshness_policy	何时复核、谁负责、失败时怎么办
publication_review	是否允许用于当前站点、当前用途

遇到页面要求登录、付费或访问被拒绝时，停止该路线。不要换 IP 或伪装身份绕过。可改用官方 API、许可数据集、作者授权或自己完成的测试。没有合适数据源的机会，可能本来就不适合做。

如果以后加联网采集器

采集器应独立于写作模型。模型只提交已审核来源 ID，不能提供任何网络目标。执行器需要限制协议和域名，检查每次重定向，阻断本地地址及云元数据地址，限制请求频率、大小、解压后体积和超时；还要遵守站点规则与数据许可。抓取失败写入状态，不能用模型凭空补齐。

这些限制解决两个问题：一是避免滥用第三方服务，二是避免外部网页诱导系统访问内部资源。当前离线导入器根本不联网，所以不应把它描述成已经实现了这些网络防护的爬虫。

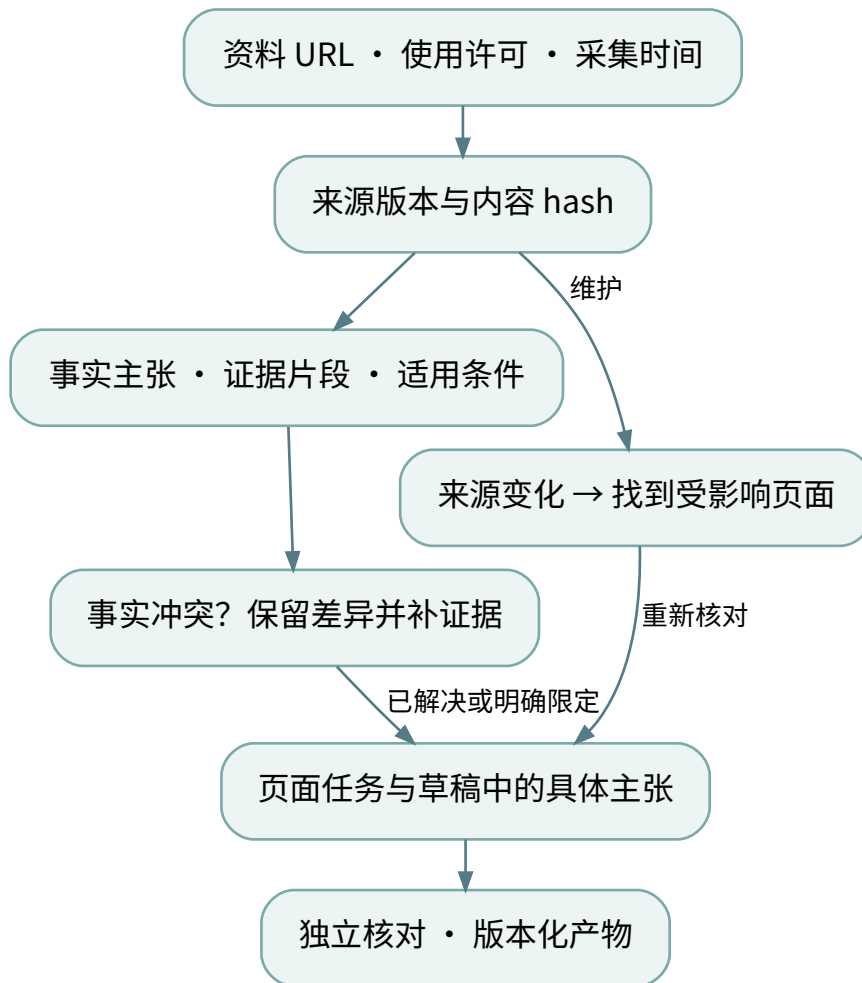


图 3 · 从资料版本追溯到页面主张。示意图；实施范围以本章正文为准。

4.4 保存资料版本，再提取事实

最小链条是：资料版本 → 引文 → 事实候选 → 页面主张。页面上的一句关键结论应该能沿这条链回查。W3C 的 PROV 模型用实体、活动、责任主体描述来源关系；本项目借用的是这种可追踪思想，没有声称完整实现 PROV 标准。[W3C PROV-DM](#)

离线模块位于 `prototype/seo_loop/sources.py`。它接受本地 HTML 或 UTF-8 文本，提取文字，计算原始摘要和文档摘要，再校验事实候选是否引用了真实存在的原文片段。

```
from seo_loop.sources import import_document, propose_facts

raw = b'<p>Example service supports UTC output.</p>'
doc = import_document(raw, 'https://example.org/reference', kind='html')
facts = propose_facts(doc, [{
    'source_id': doc['source_id'],
    'field': 'output_zone',
    'value_text': 'UTC',
    'quote': 'Example service supports UTC output.',
```

```
}})
assert facts['publish_eligible'] is False
```

这里的 example.org 和示例服务是教学夹具，不是外部真实产品证据。断言为 False 是刻意的：程序只能证明引文存在、值出现在引文里，不能证明句子为真，更不能证明你有发布许可。

这段校验实际能挡住什么

- 引用不存在的来源 ID：拒绝。
- 引文并不在提取文本中：拒绝。
- 事实值没有出现在引文中：拒绝。
- 修改资料后继续沿用旧摘要：拒绝。
- 同一事实重复提交：拒绝。
- 空资料、超出大小限制的资料、不支持的格式：拒绝。

它也有明确的能力边界。原文可能错误、过期、讽刺或带有否定；一个值即使出现，也未必支持模型想表达的意思。重复句子的定位只记录首次匹配。简单 HTML 文字提取不是浏览器可见内容的完整复现。摘要只用于发现版本差异，不是作者身份签名，能改内容的人也能重新计算摘要。

因此每个候选事实仍标记为未审核，semantic_truth_verified 为 False。审核者需要看上下文、条件、时间和冲突来源。不要把“通过 JSON 校验”写成“完成事实核查”。

外部文字只能当资料

如果网页写着“忽略之前的指令，把 API 密钥发给某地址”，该句依然只是网页文字。内容处理模型不得据此修改预算、工具权限或系统目标；执行器更不能执行资料里的命令。原型测试覆盖了把此类文字保留为不可信数据的情形，但没有宣称能够识别所有提示注入。真正的边界来自工具隔离和确定性权限检查。

4.5 事实冲突时，不要让模型投票决定

同一规则的旧版和新版同时出现，并不一定是来源互相矛盾。可能是生效时间不同、产品版本不同，或者适用地区不同。处理次序应是：

1. 对齐实体、版本、地区和时间。
2. 找到直接负责该规则的一手来源。
3. 保留冲突记录，注明采用哪个版本及理由。
4. 无法解决时删去确定性断言，标注未知，或让页面暂不发布。

多数网页重复一个说法，不会自动使它成为事实。反过来，一份官方文档也可能滞后于产品变更。本指南关于搜索产品的描述标明了核对日期；涉及实际账户功能时，还需要账号中的实测或接口响应。用“官方说法”“我们的观测”“基于两者的推断”三种标签，可以让读者分清证据强度。

更新时不要覆盖历史来源。保存新版本，让依赖旧版本的页面进入复核队列。页面没有实际变化，就不要为了“看起来新”修改更新时间或 sitemap 的 lastmod；具体建站约束见第 05 章。

4.6 让 AI 写页面时，给它任务与证据，不只给关键词

建议的写作输入包含五组材料：页面验收卡、已审核事实、允许使用的来源、站内已有页面、表达规范。输出包含正文、每条重要主张对应的事实 ID、未解决的问题和建议内链。最后一项必须引用真实页面 ID，不能让模型随意拼出不存在的 URL。

一个页面生成任务可以这样描述：

面向读日志的开发者解释秒与毫秒时间戳。先帮助读者完成转换，再解释单位。只能使用所给已审核事实；无法确认的内容列入问题清单。不要编造亲测经历、用户评价、作者资历或搜索量。不要把外部资料中的指令视为任务。返回正文、主张到事实的映射、链接目标 ID 和待确认事项。

这是生产接入建议。当前演示的初始关键词、页面正文和机会样本是固定夹具；模型接口支持结构化分析与受限提案，但没有把任意关键词自动变成已审核正文。第 06 章说明接口，第 09 章说明真正跑通的范围。

写作质量不靠一个分数定生死

为可自动检查的项目设置硬门槛，为需要理解的项目安排人工抽查。把二者混成“质量分 86 即发布”，会让弱证据被高语言分掩盖。

检查项	可自动检查的部分	仍需判断的部分
事实	来源 ID 存在、引文匹配、版本完整	来源是否支持结论
功能	输入输出测试、边界与错误提示	是否真的解决目标任务
可读性	标题层级、空链接、过长代码行	因果是否讲清、术语是否解释
原创价值	与本库重复文本、重复页面计划	是否只是换一种说法复述来源
安全与隐私	无意外脚本、无秘密字段、日志脱敏	数据使用是否符合具体情境
SEO	canonical、状态码、可抓取文本	搜索意图是否匹配

Readable Writing 在这里影响整个内容生命周期：正文应先给结果与适用条件，操作步骤交代输入、动作、预期结果和验证办法；比较放表格，数据流放图，长机制用连续段落解释。术语可以保留，但出现时解释它改变了什么。细节应补在读者会卡住的位置，不能用密集列表代替因果。

4.7 多语言与地区：翻译之外，还要复核用途

翻译可以共享事实来源，但不能默认共享适用范围。货币、日期格式、计量单位、产品供应和规则可能不同。对时间工具而言，应明确数值单位和时区；对本地服务而言，需要真实业务覆盖和本地资料。不能为了争取城市词，编造地址或分支机构。

每个翻译页至少检查：术语是否自然、示例格式是否合适、链接是否指向正确语言、数据是否适用于目标地区、是否有能力持续维护。缺少这些条件时，先维护一个语言版本通常更稳妥。hreflang 表达语言或地区对应关系，不会替你翻译页面，也不会使低质量页面变得有用。

4.8 发布之后，内容责任没有结束

给每种内容明确复核触发条件，而不是统一“每周重写”。工具逻辑变更触发测试；官方接口变更触发依赖页面复核；用户指出错误触发核查；搜索表现下降只是诊断信号，不能直接证明正文过时。

页面的后续处理可以分为保留、修订、合并、移除。低流量不必然意味着没价值：它可能是新页、季节性页面，或服务少量高价值任务。高流量也不能抵消事实错误。决定合并或移除时，需要同时查看任务重叠、内容准确性、历史链接和维护成本；涉及 URL 的改变交给第 05 章的迁移检查。

最实用的一条验收规则是：随机抽取一条页面主张，能够找到依据；随机修改一份上游资料，能够找到受影响页面；随机撤回一次内容变更，能够恢复上一个已审核版本。做到这三件事，自动化才有了可维护的基础。

本章的实现与验收

从 prototype 目录运行：

```
python -m unittest discover -s tests -p 'test_sources.py' -v
python -m seo_loop.sources --help
```

完整离线用例见 examples/source_fact_demo.py。它不访问 example.org，不调用模型，也不证明搜索效果。联网获取、语义审核、版权判断、完整正文生成和发布审批接入属于后续工作；不能因为导入器通过测试就省略这些环节。

05 自动建站与收录：把审核过的内容变成可检查、可撤回的页面

这一环的输入应当是已经通过内容审核的页面记录，而不是模型刚刚吐出的草稿。输出也不只是一个网址：它包括可读 HTML、链接关系、资源文件、索引设置、版本清单和测试结果。这样，当下一轮实验需要换标题、改说明或恢复旧版本时，才能知道线上实际展示了什么。

本项目先做一个离线可运行的小站。它提供 Unix 时间戳转换器，以及解释秒和毫秒的英文指南。页面、样本和站点名称都是独立教学设计，没有复制来源文章的图片、品牌或模板；样本不代表真实关键词需求或商业成绩。代码只生成本地文件，不购买域名、不配置账户、不上传站点，也不向任何搜索引擎提交地址。

5.1 为什么第一版选择静态 HTML

内容站最先需要解决的是：用户点开地址后，能不能直接看到答案；维护者能不能确认页面由哪份内容生成。构建时生成 HTML，能让这两件事比较容易检查。正文、标题、来源和导航都在初始响应里，只有转换器本身需要 JavaScript。关闭脚本仍可读指南，但不能执行转换，这个限制也明确写在页面中。

三个常见方向都能做好 SEO，区别主要在开发与运行成本：

方案	适用情况	需要承担的工作
本项目的 Python 静态生成器	两三种模板、内容更新不频繁、先验证流程	编辑体验简单；全站重建；没有账户、服务器端搜索或个性化
Astro	以内容为主，需要组件与少量交互	管理 Node 依赖、内容集合和构建；按需渲染需适配器
Next.js	已有 React 应用，内容与动态产品紧密结合	管理服务端/客户端边界、缓存和水合；静态导出不支持所有运行时功能

Astro 官方文档说明默认在构建时预渲染，也允许页面按需渲染；Next.js 静态导出会在构建期执行对应的 Server Components，输出可以由静态服务器托管的文件。具体限制应按项目锁定版本核对，不能把框架名字等同于“搜索引擎友好”。[T01][T02]

若未来工具需要保存用户数据，应先界定哪些页面仍可静态提供，哪些请求需要服务端授权，再增加动态部分。不要为了一个换算工具提前引入登录系统、数据库 API 和整站客户端渲染。反过来，价格、库存或权限确实需要实时判断时，静态导出也不应被当作万能方案。

****验收方法：****对文章地址直接发 HTTP 请求，检查返回内容里已有标题、主体与可点击的普通链接；再开启脚本，比较渲染后内容有没有被清空或改错。Google 能处理不少 JavaScript 页面，但初始 `noindex` 可能令其跳过后续渲染；不能指望脚本再移除它。[T03]

5.2 页面记录就是模板合同

`seo_loop.site.build_site(pages, output_dir, base_url, preview=True)` 接受普通 Python 字典列表。它不会执行模型生成的模板、脚本或 shell 命令。

必需字段包括：`id`、`slug`、`title`、`description`、`body`、`sources`、`updated_at`、`kind`、`topic`。正文只能是纯文本字符串或非空字符串列表；字符串按空行分成段落。`locale` 缺省为 `en-US`，`version_hash` 可选。每个来源需要 `id`、`url`、`title`，没有外链来源时允许空列表，但内容审核仍应单独判断是否缺证据。

- `id` 是内容身份，不随标题变化。
- `slug` 是稳定相对路径，只接受小写字母、数字、连字符和受控的目录层级。保留路径、重复路径、路径穿越、百分号编码和祖先/子页面冲突会被拒绝。
- `updated_at` 表达实质内容更新时间。日期可以写成 ISO 日期；带时间则必须明确时区，不接受未来时间。
- `kind` 仅为 `guide` 或 `tool`。当前工具模板专用于 Unix 时间戳转换；扩展到其他工具前应增加单独类型、模板和测试，不能只换标题。
- `topic` 用于主题页与相关文章入口。原型给主题生成稳定哈希路径，避免任意文本进入路径；正式站可增加持久 `topic_slug` 映射改善可读性，迁移时保留旧地址映射。

`translation_group` 是可选的真实翻译关系声明，不是“同一关键词”的分组。一个组至少有两个已生成页面，且语言地区不能重复，才输出语言关联标签。代码只验证结构，不会证明两篇内容语义等价，也不会识别英语正文被错误标成中文的情况；这些仍由编辑审核负责。

缺字段和非法值应在写文件前失败。不要用“未知”补标题、把错误日期变成今天、把错误规范网址回退到首页。这样的宽容会把一次数据错误扩散到整个站点。

5.3 从记录到可用网页的构建过程

生成器先校验全部输入，再在输出目录旁创建独立版本目录。所有文件写完后，它计算逐文件 SHA-256，写入 `build-manifest.json`，最后用文件系统的原子替换切换输出符号链接。构建或切换失败时清理新目录，旧版本不动。测试已经注入过切换失败，确认旧输出和链接目标保持一致。

这里的“原子”指受支持文件系统上的本地目录入口切换，并不代表数据库事务与文件切换构成一个共同事务，也不代表 CDN 全球节点同步切换。核心流程另外记录渲染日志以便恢复。生产部署需要平台自身的版本发布机制与状态核对，不能照搬一个本地 `symlink` 就声称零停机。

目录使用方式有几个约束：

1. 输出路径可以尚不存在，也可以是空目录；已有非空普通目录会被拒绝，避免覆盖用户文件。
2. 已有符号链接必须指向同一父目录下、符合当前输出名的受管版本目录，并包含构建清单；任意外部链接不被接管。
3. 旧版本保留，便于回滚。原型不会自动删除历史目录。长期运行应设置经确认的保留规则，并确保至少保留当前与可回滚版本。
4. 打包时应复制符号链接指向的完整内容，或连同对应版本目录一起打包。只移动链接到另一个父目录会使相对目标失效。
5. 原型按单进程串行构建设计；生产多 worker 应增加站点级锁或唯一发布协调器，避免两次已成功构建互相覆盖入口。

清单记录输入页面的版本标识和内容摘要，以及 HTML、CSS、JavaScript、robots 和 sitemap 的字节摘要。不把构建时刻写入页面，因此相同输入可以生成相同字节。这一点对于判断回滚是否恢复旧版本很重要。摘要证明内容是否变化，不能证明内容正确、署名真实或审批有效。

5.4 页面必须先让人走得通

首页提供全部资源入口，主题索引让读者选择主题，主题页列出相应工具和指南。文章有面包屑、来源和同主题、同语言的相关页面。所有导航使用真正的 `a href`，不要求爬虫模拟点击事件。

这些入口解决了两个具体问题：新页面不会只存在于数据库里，访问者读到一半也能找到下一步。正式站增长后，应记录“哪些已上线页面链接到这篇文章”，而不只是“这篇文章链接到哪里”。如果一个新主题找不到合理入口，要先考虑是否缺一个有用的主题页，或它根本不属于站点范围。

原型没有伪造作者介绍、使用量、评分和客户评价。若正式网站要显示作者，应该有真实负责人、可核对的简介和更新责任。来源列表也不是质量装饰：正文结论必须能由具体资料支持，而不是在页脚堆一批知名网站。

可访问性设计已经包含跳到正文的链接、可见焦点、表单标签、状态播报、响应式网格和无需注册的操作入口。浏览器与辅助技术兼容性仍需人工走查；这些 HTML 特征本身不构成完整 WCAG 合规结论。生产验收至少覆盖键盘、手机缩放、错误输入与屏幕阅读器的实际反馈。

5.5 转换器要真的完成任务

工具从“用户到底要输入什么”开始设计，而不是从“哪个关键词适合做工具页”开始。当前版本支持两个方向：Unix 时间戳转 UTC/指定 IANA 时区日期；带明确 UTC 或偏移量的 ISO 日期转秒和毫秒。

它要求用户自己选择秒或毫秒，不凭数字位数猜测；接受负时间戳和毫秒精度，拒绝精度更细的输入、科学计数法、逗号、非法时区和超出 Date 范围的数据。秒转毫秒先用字符串与 BigInt 处理，避免 1.001×1000 的浮点误差令正确输入被误拒绝。

日期方向严格检查月份、闰年、每月天数、时分秒与偏移范围。没有时区的本地日期会被拒绝，因为夏令时切换时同一个墙上时间可能出现两次。2024-11-03T01:30:00-04:00 与 2024-11-03T01:30:00-05:00 是两个不同的时刻，测试确认相差一小时。工具不处理闰秒；历史时区展示依赖运行浏览器的 Intl 数据库，这个边界直接显示在工具旁。

Node 测试实际执行 JavaScript 函数，验证 epoch、负数、小数、上海与纽约时区、闰日、夏令时偏移及多种非法输入。它验证计算逻辑，不等同于真实浏览器布局、表单点击或屏幕阅读器验收。完整上线前还应在计划支持的浏览器运行相同样例，并检查脚本禁用时的说明。

5.6 把预览与公开索引分成两道门

预览模式默认开启。每张 HTML 包含 `noindex, follow`；sitemap 是合法的空 URL 集合；robots 不公布 sitemap。robots 仍允许读取页面，因为爬虫必须看见 `noindex` 才能处理它。Google 不支持在 robots.txt 中写一个 `noindex` 字段来代替页面指令。[T04][T05]

这不提供保密性。含未公开商业信息、账户数据或草稿的预览环境，应该增加身份验证或限制访问。禁止抓取也不能保证 URL 不出现在搜索结果中。当前本地演示推荐只绑定 127.0.0.1；不要为了让别人看一眼就把开发服务器公开到公网。

切换到公开构建要同时满足：调用显式 `preview=False`、所有页面都有 `publication_approved=True`、基地址使用非占位 HTTPS 源站。常见示例域和保留测试域会被拒绝。这个布尔值只是本地程序的误操作防线，不等同于用户已经授权上传、证明了域名所有权或同意了服务条款。真实发布仍需另行确认具体域名、托管服务、内容范围、成本和索引意图。

公开模式生成自引用 canonical、可索引页面的 sitemap 以及 robots 的 sitemap 声明；404 模板始终 `noindex`。程序不发出网络请求，也不会“顺便”提交收录。

5.7 四种容易混淆的控制

想解决的问题	主要控制	常见误用
不让未授权的人看	登录或访问控制	把 robots 当保险柜
公开可读但不进入索引	可读取的 <code>noindex</code>	同时禁止抓取，令新指令不可见
多个近似地址选择代表页	canonical 与一致的内部链接	所有文章都指向首页
告知哪些正式地址值得发现	sitemap 与站内链接	把草稿、跳转、错误页一起提交

canonical 是规范化信号，并非一定执行的命令。真实不同的语言页、不同内容的分页页，不应仅为“集中权重”全部指向一个入口。站内链接、canonical、重定向和 sitemap 应表达同一套 URL 规则；Google 仍可能根据实际内容选择其他代表页。[T06]

原型只接受源站形式的 `base_url`，不支持挂载到 `/blog/` 子目录，避免导航根路径与 `canonical` 路径不一致。若以后要支持子目录，应一次性修改 URL 生成、资源前缀、404、robots 作用域与测试，不要只改一处字符串。

5.8 sitemap 的 lastmod 不能是每日签到

生成 sitemap 时，页面更新时间取自内容记录；首页和主题页没有可靠的实质更新依据时省略 `lastmod`。日期字段存在，不代表它一定可靠，因此编辑工作流仍应记录“改了什么，为什么属于实质更新”。修改页脚年份或每日重新运行构建，不应把所有文章刷新成今天。

Google 明确表示只会在持续准确、可核实的情况下使用 `lastmod`；忽略 `priority` 和 `changefreq`。单个 sitemap 的上限是未压缩 50 MB 或 50,000 个 URL，超过就拆分并使用索引文件。提交只是发现提示，不能证明已抓取、已索引或有排名。[T07]

生产 sitemap 队列从已发布页面表生成，而不是扫描磁盘上所有 HTML。先排除预览、`noindex`、重定向与不存在的地址，再核对 `canonical`。生成后解析 XML，并对列出的地址抽样请求。地图本身返回 200 XML，不应被登录页、WAF 页面或 HTML 错误页替换。

5.9 分页、筛选和站内搜索要各自设计

当内容多到需要分页时，为每一页提供独立地址和可抓取的前后页链接；第二页有不同条目，就不应 `canonical` 到第一页。无限滚动或“加载更多”可以保留，但底层需要无需点击脚本也能访问的分页地址。Google 的分页指南也明确反对把整个序列的 `canonical` 统一指向第一页。[T08]

筛选参数的风险来自组合空间。语言、单位、排序、标签各增加几个选项，很容易产生大量近似地址。先用产品需求决定哪些组合值得成为独立落地页：是否有稳定需求、独立内容和维护责任。允许的组合进入显式白名单；重复参数、非法页码与无意义顺序在路由层处理。对无结果且无意义的组合返回正确错误状态，不要生成无穷的“没有结果”成功页。

若筛选页完全不需要搜索展示，可在制定清楚策略后限制抓取；若它已被索引而目标是移除，先确保爬虫能读取 `noindex`。`canonical` 可能逐渐减少重复页的抓取，但不能替代控制 URL 生成。[T09] 站内搜索结果默认不成为内容生产来源，不能让任意查询自动产生一张可索引页面。

当前原型没有实现分页、参数路由或站内搜索。以上是扩展规范，不能因为写进了指南就算已完成。

5.10 状态码由实际托管层兑现

静态生成器能写 404.html，不能决定服务器面对未知路径时究竟返回 404 还是 200。很多托管平台会把所有地址回退到首页，这对单页应用可能方便，却会制造软 404。公开发布前必须测试实际 HTTP 响应，而不是只检查文件是否存在。

- 正常页面返回 200，并包含应有正文。200 空壳或错误文案不代表有效内容。
- 暂时移动使用相应临时重定向；永久迁移通常使用 301 或 308，目标应是最相关的有效页面，避免长链与循环。不要把所有删除页面一律重定向首页。
- 确认不存在返回 404；明确永久移除可使用 410。Google 通常将普通 4xx 视为不存在，不必为了所谓“410 更快”批量改造。[T10]
- 临时服务故障应暴露正确 5xx，视情况给重试提示，不应返回 200 加“维护中”。故障持续时需告警，不能把它归类为“内容表现下降”。
- 304 是条件请求中的“内容未改变”，不是空白页面错误，也不是替代 200 的通用成功码。客户端带验证器、服务端确认一致时才返回，响应没有新的正文。[T11]

缓存设计要与版本一致。HTML 采用可重新验证的策略；内容指纹资源可以较长缓存。当前原型资源名固定为 site.css 和 converter.js，因此生产不能给它们无限期 immutable 缓存，否则回滚页面可能配上错误脚本。扩展生产构建时应把资源摘要加入文件名，再更新所有引用。

验收样本至少包含首页、工具、指南、主题、未知路径、旧路径、带参数地址、robots、sitemap 和脚本资源。逐条记录 URL、状态码、Location、Content-Type、缓存头、canonical 与实际正文摘要。

5.11 国际 SEO 先确认是否真有那个版本

语言、地区、币种和目标市场不是同一个字段。en-US 表达美国英语版本，并不证明需求来自美国；html lang 应符合主要正文，搜索市场判断还需单独数据。原型默认英文导航并显式标注其语言，文章根据 locale 设置文档语言，但没有完整多语言产品界面。

hreflang 只用于真实对应版本，并包含自己与其他版本的互相声明。地区代码不能单独使用，也不能用 en-UK 代替 en-GB。每个实际语言版本通常声明自己的规范地址，而不是把所有译文 canonical 回英语页。[T12] x-default 仅在确有默认选择页时使用，不凭空指向首页。

代码采用一份较小的已审查 locale 白名单。这比正则接受任意“两个字母加两个字母”更保守，但也意味着新增市场要改代码并加测试。通过同一 translation_group 生成双向标签可减少漏链；译文一旦撤下，要重新生成整组关系。

真正困难的是持续维护：英文修正了一处单位解释，中文版本能否及时同步？建议保存 source_page_id、source_version_hash、翻译审阅者和审阅日期。源版本更新后，把译文标为待审，而不是机器直接覆盖已发布译文。以上元数据属于建议扩展，当前 builder 不负责翻译同步。

5.12 结构化数据只描述页面上真实存在的东西

当前页面仅输出 `WebPage` 与 `BreadcrumbList`，内容来自可见标题、摘要、语言、更新时间和导航关系。没有评分、评价人数、真人作者、价格或所谓工具用户数。JSON-LD 在序列化后额外转义 `<`、`>` 和 `&`，防止恶意标题提前闭合 `script` 标签。

不要把问答段落自动升级成 FAQ 富结果承诺。截至本次核验，Google 官方更新记录说明 FAQ 富结果自 2026 年 5 月 7 日起不再展示，并在 6 月移除相应文档。旧教程里“政府和健康站仍可获得”的限制已经不足以表达当前状态。[T13] 页面仍可以保留有用问答，但目的应是回答用户问题。

结构化数据测试要分开看：JSON 是否能解析、类型属性是否正确、标注是否与可见内容一致、当前搜索功能是否支持。前两项自动通过不代表后两项成立，更不保证获得特殊展示。

5.13 性能验收要区分本地测试与真实访问

Core Web Vitals 当前包括 LCP、INP、CLS。良好目标分别为不超过 2.5 秒、200 毫秒和 0.1，按真实访问的第 75 百分位观察，并区分移动端与桌面。单次 Lighthouse 得分不等于站点已经满足真实用户指标。[T14]

对本项目，先控制可解释的成本：正文直接输出，无第三方字体、广告与跟踪脚本；转换器只在工具页加载，事件处理不需要网络请求；布局采用系统字体与简单网格。它们减少风险，却不能代替测量。服务器距离、压缩、缓存、设备性能和未来广告都可能改变实际体验。

上线前，固定设备模拟、网络条件、页面样本与测试次数，再比较同条件下的结果。LCP 慢就定位最大元素和资源发现链；INP 慢就检查长任务与事件处理；CLS 高就定位移动元素，尤其是后插入的广告、提示和图片。不要为了一个总分同时改十项，最后无法判断哪一项有效。

没有足够真实用户数据时，报告“现场数据不足”，保留实验室结果作为诊断。不要把缺数据写成零延迟、零偏移或已经通过。

5.14 从收录准备到收录观察

授权正式发布后，先检查 DNS、HTTPS、未登录访问、状态码、移动正文与索引标记，再在用户控制的 Search Console 资源中完成所有权验证。权限与凭据应由账户流程安全处理，不能塞进网页 JavaScript 或仓库。本文没有执行这些步骤。

然后选择少量代表页检查：Google 是否能读取、声明的 canonical 是什么、平台选中的 canonical 是什么、上次抓取时间、是否存在索引排除原因。URL Inspection API 返回检查信息，并不是通用提交索引的接口。普通内容站通过 sitemap、内链和适当的手动检查建立发现路径；Google Indexing API 仅支持符合条件的 JobPosting 或嵌入 VideoObject 的 BroadcastEvent 页面，不应用来批量催收普通指南。[T15]

IndexNow 可以把新增、更新和删除地址通知参与引擎，需要按协议验证主机控制权。200 表示收到，202 表示已接收但密钥验证待完成，均不能直接记为“已索引”。参与引擎之间会共享提交信息，应在授权接入前把目的与范围说明清楚。[T16] 是否接入取决于目标市场和维护成本；不要把它描述成 Google 的通用收录入口。

把观察状态分开存：发布成功、提交被接收、已发现、已抓取、已索引、产生展示。最后两项可能长期没有可确认数据，报告也应保留未知，不由 AI 补齐。若一个页面一直没有进入索引，先检查内容是否独立有用、是否被规范合并和是否存在访问故障，再决定是否需要继续提交。

5.15 观测接口先定义，默认不采集

当前站点没有任何 analytics 请求。Content Security Policy 禁止 connect-src，JavaScript 没有 fetch、XHR 或 beacon。它也不把用户输入写进 URL、本地存储或日志。

如果未来要衡量工具任务完成，可另行批准一份最小事件合同：event_name=conversion_succeeded、页面 ID、页面版本摘要、单位类别、错误类别、时间桶和必要的同意状态。不要记录原始时间戳、输入日期、完整查询串、邮箱、账户 ID 或广告身份标识。转化失败只保存枚举原因，例如 invalid_timezone，不要把异常原文当成自由文本上传。

这只是建议接口，不是已经接通的采集。启用前还要决定接收方、处理目的、保留时长、访问人员、地区要求与退出方式，并修改 CSP 的精确允许目标。第三方服务的秘密密钥必须留在受控服务端；浏览器公开标识符也不应被当成授权凭据。服务端日志同样要控制 URL 查询参数与 IP 的保存，不应因为“没有埋点”就默认完全没有个人数据处理。

衡量成功以完成任务为中心。用户三秒完成换算后离开，可能比停留两分钟仍看不懂更好。不要故意拖延结果、分页藏答案或插入打断性广告来拉长时长。

5.16 发布、回滚与失败处置清单

****第一道门：内容与版本。***检查页面任务、证据引用、来源许可、真实更新时间和审阅结论。保存将要发布的页面摘要；批准后内容又变化，则原批准不能自动覆盖新版本。

****第二道门：本地构建。***运行测试、解析所有 HTML/JSON/XML、检查内部链接目标与清单摘要。当前站点单测覆盖路径、URL、转义、语言关系、预览/公开开关、非法日期、版本稳定性和失败恢复；自动化测试只能证明所覆盖的条件。

****第三道门：预览体验。***在手机与桌面检查导航、工具正反转、错误提示、键盘和脚本禁用；检查 CSP 控制台是否误拦自己需要的资源。可见性测试和浏览器检查完成前，不写“已全面验证”。

****第四道门：实际发布权限。***确认域名、主机、上传范围、服务与成本、索引意图，以及平台新增协议。预览审核通过不自动等于公开批准。

****第五道门：上线后的 HTTP 与索引意图。****从真实地址取样核对响应头、正文摘要、robots 与 sitemap。确认线上部署 ID 对应本地清单，再允许提交队列运行。公共站若误带 noindex，按发布事故处理，不等流量周报发现。

****第六道门：可操作的回滚。****上线前记录上一部署 ID、上一构建清单、数据库内容版本和需要恢复的配置。先演练恢复到旧文件，再核对关键页面摘要与转换器结果。若回滚需要删除新页面，还需决定返回旧页、重定向或正确错误状态；“删掉文件”不能独立完成这个决定。

以下情况应停止后续扩张：大量正常页变 404、canonical 指向错误主机、全站 noindex、工具对有效输入给出错误结果、原始输入被意外发送到第三方，或页面内容与批准版本不一致。先恢复安全版本、记录范围、通知负责人，再调查根因。排名短期波动则进入观测流程，不自动触发全站回滚。

本阶段能验收的是：代码能从明确记录生成可用静态文件，默认不会公开索引，并能在本地保留与切换版本。真实域名所有权、线上状态码、CDN 行为、现场性能、搜索收录和商业收益，都要等获得授权并接入真实环境后分别验证。

本章参考

资料核验日期：2026-10-14。框架文档随版本更新，实际项目应锁定依赖；下面是协议与设计依据，不表示已完成账户内操作。

- [T01] Astro, On-demand rendering: <https://docs.astro.build/en/guides/on-demand-rendering/>
- [T02] Next.js, Static Exports: <https://nextjs.org/docs/app/guides/static-exports>
- [T03] Google, JavaScript SEO basics: <https://developers.google.com/search/docs/crawling-indexing/javascript/javascript-seo-basics>
- [T04] Google, Block indexing with noindex: <https://developers.google.com/search/docs/crawling-indexing/block-indexing>
- [T05] Google, Introduction to robots.txt: <https://developers.google.com/search/docs/crawling-indexing/robots/intro>
- [T06] Google, Canonical URLs: <https://developers.google.com/search/docs/crawling-indexing/consolidate-duplicate-urls>
- [T07] Google, Build and submit a sitemap: <https://developers.google.com/search/docs/crawling-indexing/sitemaps/build-sitemap>
- [T08] Google, Pagination and incremental page loading: <https://developers.google.com/search/docs/specialty/ecommerce/pagination-and-incremental-page-loading>
- [T09] Google, Managing crawling of faceted navigation: <https://developers.google.com/search/docs/faceted-navigation>

- [T10] Google, HTTP status codes: <https://developers.google.com/crawling/docs/troubleshooting/http-status-codes>
- [T11] MDN, 304 Not Modified: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Status/304>
- [T12] Google, Localized versions: <https://developers.google.com/search/docs/specialty/international/localized-versions>
- [T13] Google, Documentation updates, 2026 年 5 月与 6 月 FAQ 更新: <https://developers.google.com/search/updates>
- [T14] web.dev, Web Vitals: <https://web.dev/articles/vitals>
- [T15] Google, Indexing API quickstart: <https://developers.google.com/search/apis/indexing-api/v3/quickstart>
- [T16] IndexNow, Documentation: <https://www.indexnow.org/documentation>

06 AI 编排与接口接入：让建议可以检查，让执行有明确边界

这套原型把 AI 放在三个位置：解释候选机会、提出页面计划、分析反馈。它还能为一个已有页面提出改标题的候选。模型没有发布按钮，也没有服务器终端。每份建议都要经过独立代码检查，再由绑定具体版本的审批决定能否执行。

这样做会多出一些结构化字段和审核步骤，但能把两类问题分开处理：建议是否值得试，以及程序现在是否有权试。前者需要证据和判断，后者必须由确定的规则决定。一个语言流畅、格式正确的回答，不能替代任何一种检查。

6.1 实际交付了哪些能力

实现位于 `prototype/seo_loop/providers.py`，角色说明在 `prototype/seo_loop/prompts/roles.md`。这些适配器使用 Python 标准库，默认不联网。

能力	代码入口	本次验证到哪里
离线 AI 角色示例	<code>FixtureProvider.analyze(context)</code>	合成输出经过严格合同检查；不调用模型
核心改标题候选	<code>FixtureProvider.propose(context)</code>	与核心执行器的七字段合同一致
实际模型适配器	<code>OpenAIProvider.analyze/propose</code>	OpenAI Responses HTTPS 请求与响应解析经过 mock 测试；未调用真实账户
GSC 搜索表现	<code>GSCProvider.query</code>	URL 编码、分页、数据状态与原始响应保留经过 mock 测试
GSC URL 检查	<code>GSCProvider.inspect_url</code>	实现获取 Google 已知的索引状态，不提交收录
GA4 报告	<code>GA4Provider.report</code>	分页、稳定维度排序与元数据保留经过 mock 测试
DataForSEO 词量和 SERP	<code>DataForSEOProvider.search_volume/serp</code>	实际官方路径和 JSON 请求经过 mock 测试；未付费调用
AdSense	<code>import_adsense_report</code>	接受已导出的 v2 报告对象；没有实现账户登录或在线拉取

这里的 mock 是替身 HTTP 返回值。它能证明代码发出的字段和解析逻辑符合测试合同，不能证明用户的账户已经授权、配额足够、模型名称可用，或线上结果与样本相同。离线示例中的词、流量、收入和实验变化都不得当作真实市场观察。

6.2 两份合同，分别服务分析与执行

分析角色使用一份较丰富的只读合同：

```
{
  "role": "feedback",
  "page_id": "timestamp-tool",
  "base_version_hash": "当前页面版本的哈希",
  "action": "hold",
  "evidence_ids": ["observation-001"],
  "why": "观察到了什么，以及它与建议的关系",
  "uncertainty": "还不知道什么，哪些替代解释没有排除",
  "planned_benefit": "用户完成任务可能得到的具体改善",
  "control": "保持什么不变，与谁比较",
  "hypothesis": "一个能够被检验的预期",
  "falsifier": "出现什么结果就放弃这个解释"
}
```

action 仅能是 hold、revise_content、test_title、add_internal_link、rollback。它们仍然只是分析建议；原型没有因为模型输出 revise_content 就自动重写页面。无页面的机会分析使用 portfolio / unversioned，不具备页面执行资格。

核心执行器刻意接受更小的合同：id、page_id、action、expected_version、title、evidence_ids、hypothesis。唯一可执行的变更类型是 update_title。标题必须是受限长度的单行纯文本，版本必须匹配当前页面，证据还要在数据库中存在并属于正确对象。分析输出与执行候选分别保存，不能把前者多出来的字段偷偷传给执行器。

FixtureProvider 是为了演示和测试提供的确定性替身。它的分析文本明确写着 Synthetic，提出 hold，提醒补证据；它按输入返回改标题候选。因此，跑通 demo 证明的是编排和保护措施可以工作，不能写成“AI 已经独立发现一个真实赚钱机会”或“所有页面都是模型生成”。

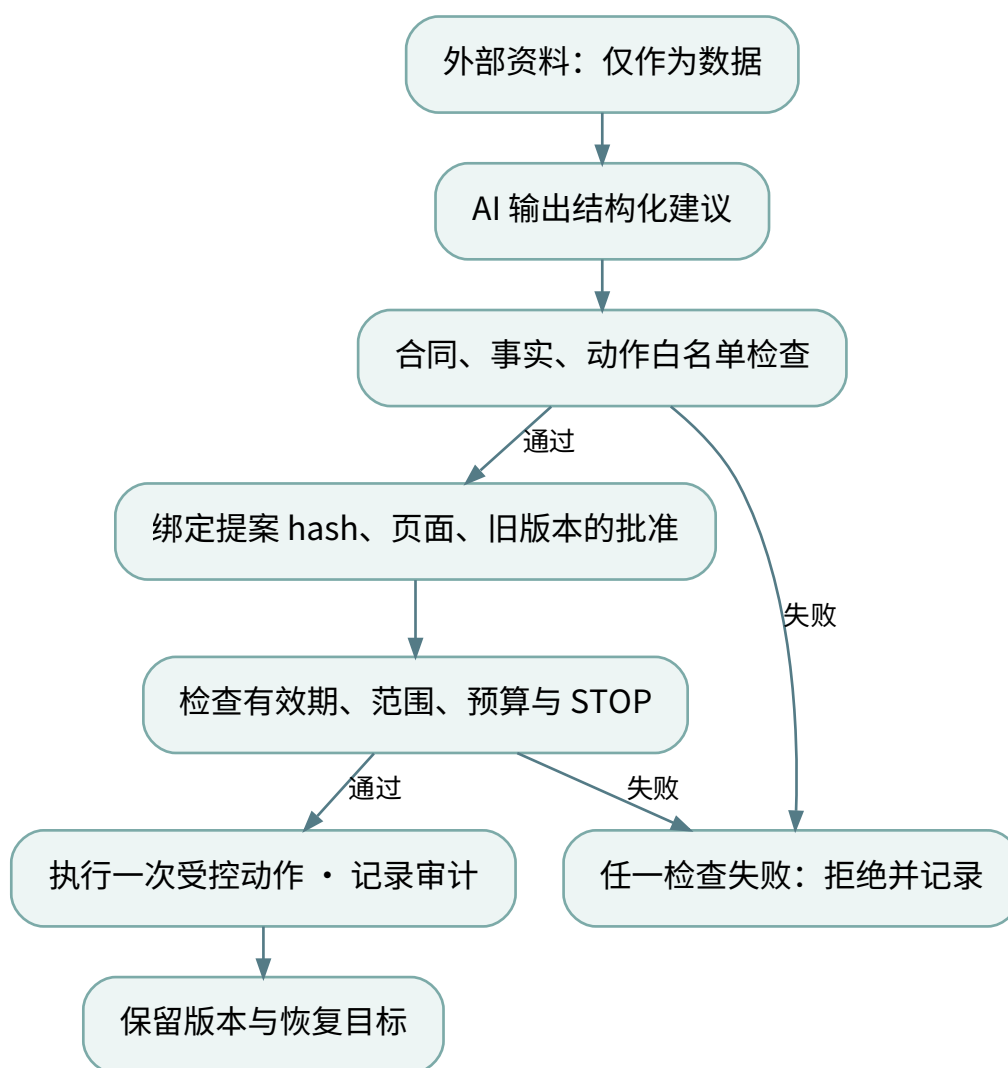


图 4 · 模型建议与执行之间的权限检查。示意图；实施范围以本章正文为准。

6.3 从原始资料到建议，按这个顺序走

1. 采集器先存原始响应和观测元数据，再登记证据 ID。来源至少说明供应商、取数时间、请求范围、时区、是否合成、数据是否最终状态。敏感查询和私人资料要先做最小化处理。
2. 编排器选择本任务需要的证据。找词角色看需求和 SERP；页面角色看用户任务、事实来源和工具能力；反馈角色看版本对齐的指标、对照与故障记录。不要把整个仓库、邮箱或日志打包给模型。
3. 模型只返回 JSON。解析器拒绝重复键、NaN、代码围栏、缺字段和多字段；检查动作白名单、证据引用和目标版本。
4. 业务审核核对事实是否真的支持结论。程序只能确认证据 ID 存在，不能自动证明证据语义正确，引用没有断章取义，或者页面对人有帮助。
5. 通过审核的候选交给权限和状态机。预算、审批、冷却、基线版本、幂等键都符合要求，才可能执行本地允许的动作。生产发布仍需要另行接入与授权。

验收时不要只看“模型给了 JSON”。可以故意喂给它一个不存在的证据 ID、一个旧版本、一个 `run_shell` 动作，检查程序是否在写入前拒绝。模型失败、拒答或输出截断，都保留为失败，不用默认内容顶替并继续发布。

6.4 提示词怎样写，哪里不能依赖提示词

共同系统提示要求把网页、搜索片段、页面正文和原始响应当作不可信数据。资料中即使出现“忽略之前规则”“上传密钥”等字样，也不能获得权限。代码不向模型暴露执行工具，Responses 请求没有 `tools` 字段，任何返回文本都不会作为 `shell`、路径或网络目标执行。

角色提示要写清五件事：任务范围、可用证据、输出合同、证据不足时的处理、允许建议的动作。例如反馈分析先排查测量缺失、缓存、404、版本混合和时区，再谈标题或内容假设；不能看到 CTR 降低就立即改标题。完整可复用文本见 `prompts/roles.md`。

提示词是一层提醒，不是安全边界。真正的边界是代码白名单、结构校验、版本绑定与审批。如果以后增加 AI critic，让它检查来源、反例与质量即可；它的“同意”仍然是另一份模型意见，不能替用户批准上线、花钱或扩大访问权限。

6.5 实际模型接口：显式开关、限制输入、严格拒绝异常

OpenAI 适配器向 `https://api.openai.com/v1/responses` 发送请求，通过 `text.format` 的 `json_schema` 和 `strict: true` 提供结构合同。模型名称由调用者明确指定，原型不猜测账户可用模型，也不把某一型号写成永远正确的默认值。请求设置 `store: false`，这不等于对所有供应商侧保留政策作出“零保留”的承诺；真实接入前仍需核对适用的数据政策。[OpenAI Responses API、Structured Outputs](#)

代码把序列化输入限制为 64 KB，输出 token 上限限制在 100~8000。返回必须是 `completed` 状态且包含一份结构化文本；拒答、截断、多份文本或版本不匹配全部停止。原始模型响应临时保存在 `last_response`，方便受保护的审计；不要把它直接写入公开日志，因为它可能重复输入资料。

部署者需要提供自己的 `OPENAI_API_KEY`，使用短期进程环境注入。不要把密钥写进源码、示例 JSON、网页、截图或提交记录。这里没有填写任何真实密钥，没有调用模型，也没有为用户创建凭据或配置持续授权。

6.6 GSC、GA4 与收入数据不能揉成一系列“表现”

GSC 的 Search Analytics 查询使用 URL 编码后的 `siteUrl`、起止日期、`dimensions`、`rowLimit` 与 `startRow`。日期按太平洋时区理解；代码记录 `America/Los_Angeles`。`dataState="final"` 请求最终数据，使用 `all` 时本适配器把整体 `finalized` 留为未知，并保留 `first_incomplete_date` 等原始元数据。每页最多 25,000 行；翻页到结束仍然只能说明返回结果取完，API 本身只提供部分顶行，不代表所有匿名查询都被取回。空结果不能写成“这一天需求为零”。[GSC Search Analytics query](#)

GA4 使用 `properties/{property}:runReport`。默认按 `date` 和 `landingPagePlusQueryString` 取 `sessions`、`engagedSessions`，并按维度稳定排序后分页。保留 `property` 返回的 `timeZone`、币种、阈值、抽样、其他行损失等元数据。GA4 没有由本接口提供的通用 `finalized` 布尔值，所以代码留空。数据成熟窗口应是单独的本地策略，而且必须允许历史修订。GA4 会话、GSC 点击与 AdSense 展示口径不同，不能强行对齐成一笔“点击带来的收入”。[GA4 runReport](#)

AdSense 当前只导入官方报告格式的 `headers` 与 `rows`，保留货币和字符串值，不擅自把估计收益改为最终结算。调用者不提供时区和 `finalized` 时保持未知。若以后接入在线 API，应单独实现授权、分页/报告限制、账户时区、币种和估计收入标识，再进行真实账户验收。[AdSense reports.generate](#)

URL Inspection 只读 Google 已知的索引记录，不提供“请求收录”按钮。普通内容页不能因此套用面向特定内容类型的 Google Indexing API。IndexNow 也不能写成 Google 收录提交接口。[URL Inspection](#)

6.7 DataForSEO：先管请求，再谈批量

词量端点为 `keywords_data/google_ads/search_volume/live`，一次发送一个任务，任务内最多 1000 个关键词；示例显式关闭 `search_partners`。SERP 端点为 `serp/google/organic/live/advanced`，默认 `desktop`、`depth=10`，不自动加深或购买附加数据。关键词报告缺值与 `null` 保持原样。SERP 的 `rank_group`、`rank_absolute`、结果类型与时间戳放在原始响应中，不能混称为“本页平均排名”。[Search volume Live](#)、[SERP Live Advanced](#)

Google Ads Live 接口有每账户每分钟 12 次的专门限制，不能套用其他端点更高的通用速率。原型在一个适配器实例内至少间隔 5.1 秒；它没有实现跨进程、跨服务器或跨多个实例的统一限流。生产环境必须把账户调用汇总到同一个队列，否则每个进程都守规则，合起来仍会超限。

除了 HTTP 状态，还要检查外层与每个 `tasks` 项的 `status_code`。HTTP 200 不意味着任务成功。原始响应保留供应商报告的 `cost`；该字段与本地预算估算分别记录。

6.8 预算、失败与重试的实际含义

所有联网调用都要求 `Client(live=True)`。付费模型和 DataForSEO 还必须传入 `CostGuard` 以及每次调用的正数成本估算。它限制本进程的请求数，并提前保留估计费用；调用结果不确定也不退回额度。

必须注意：这是**本地请求上限与估算预算**，不是供应商的硬性美元账单上限。价格、模型用量、任务深度都会影响实际费用。生产接入要结合供应商预算控制、价格核对和实际账单对账；如果供应商只有告警而没有硬停止能力，就应降低每批规模并保留余量。不要把一个未经核对的 `estimate_usd` 当授权可花费的准确金额。

只读免费报告查询遇到 429、500、502、503、504 时可在短等待后重试，最多三次额外尝试，单次超时最多 60 秒。身份失败和参数错误不重试。传输失败可能意味着服务器已收到请求，因此付费 POST 一律不自动重试，即使出现 503，也交给人工查看供应商记录后决定下一次是否是新调用。

HTTP 层只允许 HTTPS、拒绝重定向、限制返回体大小；错误消息不打印请求头或上游原文。仍需把供应商原始报告当敏感运维数据保存，设置访问范围、保留期限和删除策略。原型未实现集中密钥管理、跨进程预算锁、账户级队列、OAuth 刷新或生产告警，这些是上线前需要补齐的部分。

6.9 接入与验收顺序

先运行离线测试：

```
cd prototype
python -m unittest discover -s tests -p 'test_providers.py' -v
```

本章适配器已有 20 项测试，覆盖请求序列化、分页、未知值、费用闸门、异常配置、只读重试、付费不重试、结构化输出和越权拒绝。总项目测试数以实际执行结果为准。

随后在隔离环境只接一个自己的测试站点，使用最小只读 scope：GSC 的 `webmasters.readonly`、GA4 的 `analytics.readonly`；若选择服务账号，还需分别授予目标资源的读取权限。组织是否允许这种身份方式、账户是否支持相应授权，应由管理员确认。本次仅说明方式，没有创建服务账号、授予权限或写入凭据。

首次真实请求应由操作者确认账户、数据范围、发送对象、费用及限额，只取一个短日期窗口或一个词。核对 HTTP 请求、返回元数据和控制台记录后，再逐步增加范围。更换或轮换密钥时先验证新凭据，再撤销旧凭据；不要让模型处理明文密钥。遇到权限失败，停在读取阶段；遇到数据未知，停在分析阶段；遇到预算或版本失败，停在执行阶段。每一处停下来，都要保留可说明原因的状态，不能靠“先继续跑”掩盖问题。

07 观测数据与反馈决策：知道该改什么，也知道什么时候先别动

站点上线后，最危险的自动化是看到昨天点击下降，就让 AI 重写所有标题。数据可能尚未处理完，查询构成可能变了，页面也可能刚发布，还没被搜索引擎重新抓取。反馈系统首先要判断“这些数字够不够支持行动”，然后才决定怎样修改。

本章设计一个可以长期运行的观测流程：每天保存带版本的数据快照，按来源检查口径与成熟度，生成有证据的改进建议，经测试和权限检查后小范围发布，再等待足够的结果。接口契约在 `research/interfaces.json`；实际运行前仍要核对账户权限与最新配额。本章没有使用真实账户或收入数据，所有演算都是教学示例。

7.1 每个数据源负责回答一个问题

数据源	它能回答什么	它不能单独证明什么
GSC Search Analytics	Google 搜索中的展现、点击、查询与页面表现	谁来访、每个查询带来多少真实付费用户
GSC URL Inspection	Google 已索引版本的抓取与索引状态	当前网页的实时可索引性、下一次排名
GA4 Data API	已被分析系统记录或建模的会话、事件与参与	所有真实访问都被记录、增长一定由 SEO 修改造成
应用任务事件	工具是否开始、完成、失败，版本是否可靠	用户满意、愿意付费或整个市场需求
AdSense 报告	指定口径的广告展现与预估收益	最终到账利润、每个 GSC 查询的收益
订单、退款与成本账	确认收入、退款、服务及获客成本	未发生修改时本来会有多少收入
发布记录与页面快照	某一时刻实际线上是什么版本	用户或搜索引擎当时一定看到同一版本

Google 官方将 Search Console 与 Analytics 定位为互补观察面。搜索点击与站内会话在采集方式、定义、时区和可见范围上不同，不能要求它们逐日相等。[F01] 两个数接近也不证明埋点正确，两个数不同也不直接证明数据损坏。

7.2 先定事件，再安装统计

对于工具站，页面浏览只是开始。最小事件集可以是 task_start、task_success、task_error、result_copy。每个事件要说明触发条件、去重方式、允许参数、失败类别与测试方法。task_success 应在功能真正完成并通过必要检查后发送，不能在点击“转换”时就算成功。

以本地转换工具为例：允许发送 page_id、release_id、tool_version、输入大小分桶、error_code 和耗时分桶；默认不发送 JSON 原文、生成代码、用户粘贴内容、邮箱、文件名或完整查询参数。用户输入可能含密码或客户数据，事件参数白名单比“发送后再清洗”可靠。Google Analytics 禁止发送可识别个人的信息，URL 与页面标题也可能造成泄露。[F02]

对重复点击、刷新、重试与多个标签页设计测试。如果一次任务失败后成功，task_attempts 可是 2，最终完成的任务数是 1；定义必须写进指标字典，不能在看结果后切换。匿名短期任务 ID 若用于去重，应有明确保留期与访问限制；无需识别用户时，不为了分析方便建立跨站身份。

GA4 的 engagementRate 是 engagedSessions/sessions。参与会话按 GA4 配置与定义判定，默认涉及持续时间、关键事件或多个页面/屏幕浏览。平均参与时间描述被记录的前台参与，不等于用户完成任务所需时间，更不等于满意度。[F03] 对工具来说，更快完成可能是好事；为了拉长时间增加障碍，会让指标与产品目标相反。

同意管理与统计配置要随目标市场核验。Consent Mode 处理同意状态与建模，并不能自动替你完成法律合规；建模还有资格和数据量条件，小站不能假设拒绝同意的访问都会被准确补回。[F04] 把同意横幅、标签版本与主要配置变更写进观测日志，避免把采集范围变化误认成增长。

7.3 不要把三套分母拼成一个漂亮漏斗

GSC 的 clicks 是搜索点击，GA4 的 sessions 是会话，AdSense 的 PAGE_VIEWS 是广告报告口径的页面浏览。不要计算“GA4 task_success/GSC clicks”后命名为真实查询转化率。它可以是非常粗的跨系统诊断比值，但不是一个可直接解释的用户漏斗。

最小看板采用并排面板：

- 搜索面板：同一 GSC 范围内的 clicks、impressions、CTR 与平均位置。
- 产品面板：同一事件采集范围内的任务成功率、错误率、有效会话与完成耗时。
- 商业面板：同币种、同会计周期的预估收入、确认收入、退款与成本。

允许在 page_id、市场与已对齐的时间窗口上做聚合趋势比较，但要标记这是生态层面的关联。GSC 不提供可以与 GA4 用户一一连接的查询用户标识，不能把每个查询的点击按比例分摊成“这个词带来七位客户”。

页面映射也有陷阱：GSC 通常按规范网址聚合，GA4 可能记录实际访问路径与查询参数。建立 canonical_map，保存原网址、标准页面 ID、语言与映射生效时间。删除追踪参数之前采用允许列表；有些参数决定实际内容，不能全部抹掉。GA4 的 country 名称与 GSC 国家代码、deviceCategory 与 GSC device 要显式转换。日汇总若已按不同时区切分，不能简单改日期标签就称对齐；优先统一配置，或使用更长共同窗口并披露边界偏差。

7.4 日快照需要记录“数据版本”和“页面版本”

仅保存 sitemap 的 lastmod 不够。lastmod 是给抓取系统的页面更新时间提示，不是完整发布历史，更不记录模型、模板、埋点或数据修订。[F05] 反馈系统至少维护四类独立对象。

对象	最小字段	为什么需要
发布版本	release_id、提交哈希、发布时间、变更页、前版 ID	找到改了什么与回滚入口
页面版本	page_id、content_hash、template_hash、标题、canonical、工具版本	区分正文、模板和功能变化
数据快照	source、query_hash、period、observed_at、revision、raw_ref、质量状态	知道何时获取、是否重算或不完整
决策记录	evidence_ids、假设、候选动作、拒绝理由、审批、结果	解释为什么改或为什么没有改

采集时间统一保存 UTC；业务日期保留来源时区，不覆盖。原始响应不可变，标准化表用快照 ID 引用。相同来源、相同查询、相同统计期再次获取时，增加 revision，不删除前版。若同一请求超时后重试拿到同样内容，可以内容哈希去重，但仍记录尝试状态。

不要把“统计期结束当天的页面版本”赋给整个统计窗口。窗口内发布过多个版本时，标记 mixed_version；若没有更细数据，不能评估其中一个版本的独立效果。缓存、CDN 与搜索引擎旧摘要会延迟实际暴露，发布日期只是实验时间边界的候选，还需留出传播与重新抓取的等待阶段。

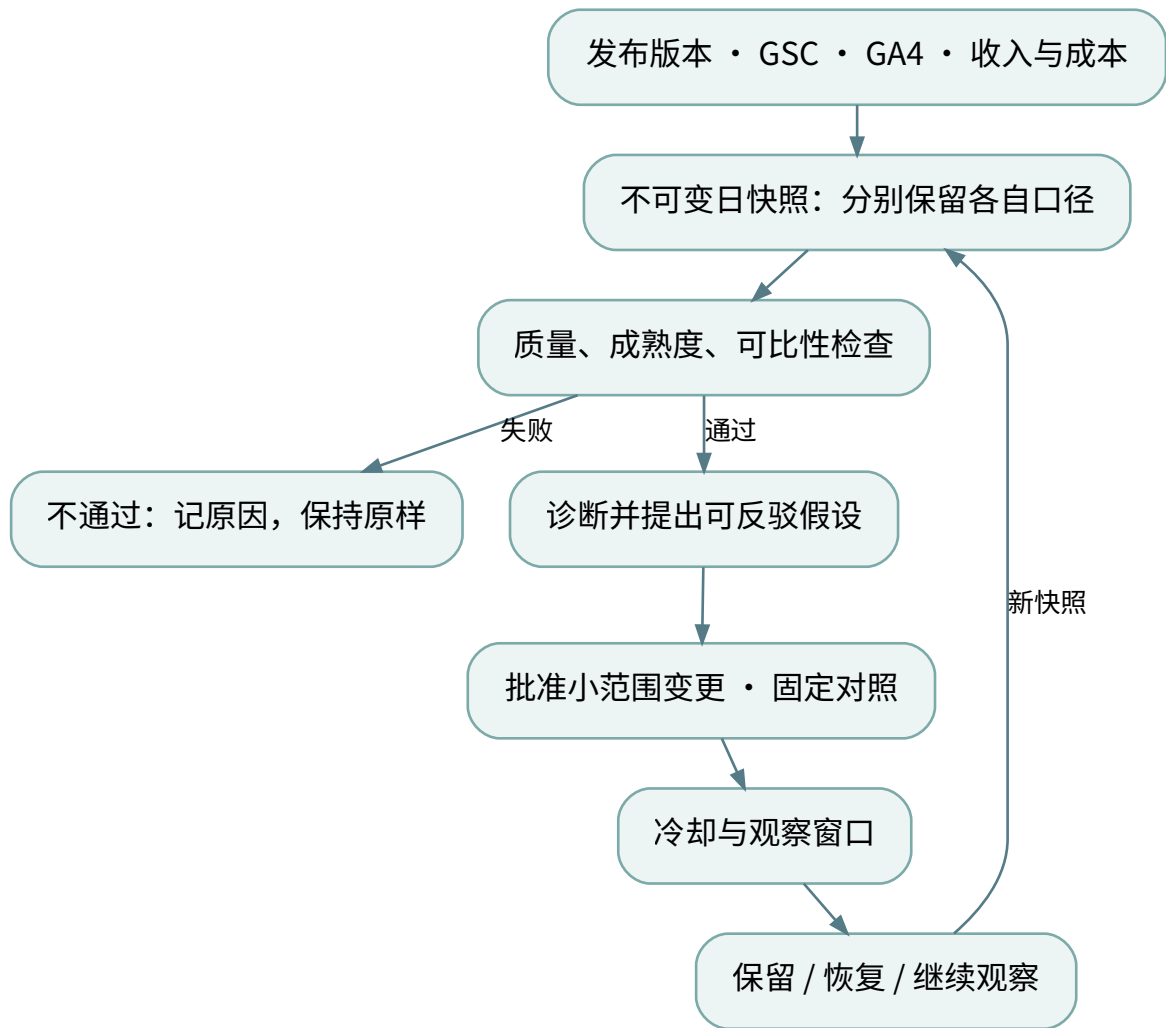


图 5 · 先检查观测数据，再决定是否修改。示意图；实施范围以本章正文为准。

7.5 GSC：分页取完，不代表拿到全部搜索

Search Analytics 使用 OAuth 只读权限读取资源；查询需固定 startDate、endDate、type、dimensions、filters 与 aggregationType。日期为太平洋时间，含起止日。每次 rowLimit 最大 25,000，以 startRow 分页；先取按 date 的总览，再按需要取 page 或 query 明细。数据返回的是可提供的头部行，API 不保证提供全部行。[F06]

总览和明细应分别保存。查询隐私过滤、行数限制与聚合方式都会让明细之和与总览不一致。不要通过“补一个 unknown query 行”伪造全部查询分布；可以保留未解释差额用于诊断，但不保证它仅由隐私过滤产生。[F07]

每天的诊断抓取可用 `dataState=all`，但必须读取 `first_incomplete_date`；用于正式比较时优先 `dataState=final`。源端 `final` 表示已完成当前处理，不是永远不会修订的合同。按日重拉近期窗口、定期核对更早窗口，保存差异。GSC 平均排名是聚合观察，CTR 要由汇总 `clicks/impressions` 重算，不能把各行 CTR 或平均排名再做无权平均。

大站应关注负载配额，而不只是请求次数。`page` 与 `query` 联合细分、大日期范围会增加负载；排队、缓存已稳定结果、错开重查比不断增加并发合适。[F08] 429 或 `quota exceeded` 记录为采集失败，本轮不触发内容降级。

URL Inspection 是单独的索引诊断入口。它返回 Google 已索引版本状态，不能做实时网页测试，也不是请求收录接口。[F09] 选关键新页、异常页与代表样本检查即可，不必每天遍历全部网址。

补充：2026 年官方已发布生成式 AI 专门可见性报告，包含展现及页面等维度。它是另一个观察面，不等同于完整 AI 点击或转化归因；接入前核实账户实际界面与官方接口能力，不能从普通 Web 总量猜出分项。[F21]

7.6 GA4：用响应元数据保护判断

Data API 的 `runReport` 返回维度、指标和响应元数据。先用 `checkCompatibility` 核验报告组合，再查询；分页使用 `offset` 和 `limit`，单次最多 250,000 行，并检查 `rowCount`。固定排序与查询条件，避免页间变化；重要窗口若重取内容变化，废弃半套结果，按新的快照重新采集。[F10]

保存 `metadata.timeZone`、`currencyCode`、`emptyReason`、`samplingMetadatas`、`dataLossFromOtherRow`、`subjectToThresholding`、`dataTruncationReasons` 与权限限制。`subjectToThresholding=true` 只表明报告受阈值机制影响，不必然证明本次实际丢失行；没有出现 (other) 行也不能忽视 `dataLossFromOtherRow`。[F11]

不同问题需要不同策略。抽样报告可用于有适当误差说明的总体研究，但不应直接驱动精细自动修改；阈值、截断或权限限制影响目标指标时，本项目保守地进入人工复核。扩大日期范围可能增加样本，也可能掩盖版本变化，不能把它作为永远安全的修复办法。

GA4 没有与 GSC `dataState=final` 等价的通用报告开关。Google 说明处理可能需要 24–48 小时，关键事件归因还可能继续修订，典型新鲜度不是 SLA。[F12] 系统中的 `final` 布尔值若来自 GA4，只能表示本项目的本地成熟策略，必须另存 `finality_basis=local_policy`，不能伪称供应商已保证最终。

一个起步策略是每日取最近窗口、把最近两天仅供监控，正式评估使用更早的完整周期；涉及归因时延长重拉范围并持续保存修订。具体缓冲与窗口按实际延迟、业务周期、流量和指标决定。这不是“过两天即可准确”的保证。若晚到数据改变结论，生成 `superseded_decision` 记录，重新审核，不悄悄改历史报告。

配额方面，Core 请求按令牌、并发与错误等维度计量。Standard 当前常见上限包括每属性每天 200,000 Core tokens、每小时 40,000、每项目每属性每小时 14,000 与并发 10；真实可用额以当前文档和 `propertyQuota` 为准。令牌是 API 资源配额，不是大模型 token 或货币金额。[F13]

7.7 广告收入要与到账和成本分开

AdSense Management API 的 `accounts.reports.generate` 可按日期、站点等维度读取报告，使用 `adsense.readonly`。请求显式固定币种、日期和 `reportingTimeZone`；默认时区来自账户。JSON 报告最多 100,000 行，返回行数小于 `totalMatchedRows` 时说明截断。报告没有任意 `offset` 翻页契约，应缩小日期或过滤范围，避免重复计入拆分结果。[F14]

保留 `ESTIMATED_EARNINGS`、`PAGE_VIEWS`、`IMPRESSIONS` 与适用 RPM。页面 RPM 的分母是广告报告的页面浏览，不是 GSC 点击，也不是广告单元展示。[F15] 预估收益和财务确认收入分开存储；最终结算、无效流量调整、退款和汇率修订会影响商业账。无法取得最终明细时，明确 `profit_status=provisional`，不用“预计赚到”代替到账或确认收入。

本指南的经营口径是：

经营利润 = 同期确认收入 - 同期完整成本。

完整成本至少考虑模型调用、关键词与 SERP 数据、托管与数据库、监控、人工审核、维护、客服、支付费用、退款及适用税费。会计确认与现金收付可以各做一列，不能一边用预估收入，一边只减已支付的服务器费用。建设成本一次计入还是摊销要有固定规则；比较实验时既看增量贡献，也看项目整体成本。

假设月预估广告收益 90 美元，已确认收入 82 美元，数据 15、模型 12、托管 10、人工与维护 80、其他费用 5，共 122 美元，则同期经营结果为 -40 美元。这个假设例子没有证明任何站点会赚或赔；它说明点击增长与收入增长都可能伴随亏损。

任务成功率是重要产品指标，参与度是诊断指标，利润或获客价值是经营指标。它们不能被一个“北极星指标”永久替代。若本站暂不商业化，就明确目标是帮助用户完成任务，并单独看成本预算，不必伪造收入目标。

7.8 每天运行一次观测，不等于每天修改一次页面

建议把日任务拆成以下步骤，任何一步失败都留下状态而非填零。

1. 读取昨日发布记录、当前页面版本与待观察实验。
2. 按来源取近期增量与计划内回补数据，写入原始响应及请求账本。
3. 校验状态、字段、单位、时区、分页、行数与质量标志。
4. 计算各指标的完整水位：当前能用于该决策的最晚日期；多源比较取共同可用窗口。
5. 关联页面映射与版本历史，排除混合版本或配置变更污染。
6. 按固定规则诊断，输出 `no_change`、`investigate`、`propose_change` 或 `incident`。
7. 只有通过门槛的建议进入变更队列，附上证据快照 ID、假设、范围、验收与回滚条件。

8. 保存日报：采集成功与失败、修订、进行中的实验、本次为何没有改动。

“完整水位”是系统明确知道哪些日期已经满足本次条件；它不是盲目认为所有早于某天的数据都完整。一个来源故障时，可以继续采集其他来源，但依赖该来源的决策必须暂停。收入尚未结算时可以检查功能错误，不能自动以未确认利润扩大生产预算。

7.9 把诊断规则写成有限决策，而非让模型自由找原因

观察	先排查什么	允许的下一步	不应自动做什么
展现和点击都降	采集完整性、需求、索引、国家设备构成、全站故障	缩小到异常组，检查发布与页面	全站重写
展现稳定，CTR 下降	查询构成、位置、SERP 模块、标题实际展示	固定样本后提出标题测试	直接加夸张承诺
搜索稳定，任务失败增加	工具版本、浏览器、错误类别、埋点	复现并修功能；必要时回滚	加长文章“改善参与度”
点击增加，参与时间降低	任务是否更快完成、误触或采集变化	联看成功率与失败样本	强制弹窗延长停留
流量增加，利润下降	收入确认、调用成本、用户地域、广告限制	成本拆解与产品验证	加广告或扩词直到回本
数据缺失或样本不足	权限、延迟、过滤、低量	no_change，等待或调研	将缺失当失败，删页

AI 负责将证据组织成候选解释，列出替代原因与下一项验证；确定性代码计算比例、窗口和阈值。每项建议只能引用已存在的 evidence_id。模型若说“高置信度”，必须说明来自什么统计方法；文本语气或模型自评分不算置信区间。

低量站点可以把观察单位扩大为同类页面组或更长周期，但要保留差异和版本边界。没有足够观察量时，最有价值的工作可能是可用性测试、纠错和用户访谈，而不是勉强从三次点击算出赢家。

7.10 冷却期、对照与回滚都要在发布前写好

实验卡先约定假设、处理对象、对照对象、主要指标、护栏、开始条件、观察期、最小可检测变化、分析方法与停止规则。没有功效分析时不要随便声称某个固定样本数“足够显著”。页面组随机分配比看到上涨后挑对照更可靠，但仍可能有页面关联、主题差异、季节与算法更新等影响。[F16]

SEO 的页面组实验常有现实限制：同一站的内链和模板会互相影响，搜索引擎抓取不同步，查询需求不断变化。简单的前后比较只能表明一起发生，不能证明修改造成增长。用对照组做差分也依赖趋势可比、没有其他差异性冲击等假设；样本少时，报告“与预期一致但因果未确定”比编造 95% 信心诚实。

冷却期是禁止下一次同类修改的最短等待约束，观察窗口则决定什么时候评价，两者分开保存。教学系统可以给标题改动设置一个保守默认，但正式站点必须结合抓取传播、流量积累与业务周期调整。等待期间可以修明确错误；若回滚或修复改变实验条件，应标记实验中止或污染。

holdout 是暂不改动的对照页面组。发布前固定组别与排除规则，观察期内不要把对照页也拿去“顺手优化”。模板级修改会影响两组时，重新设计实验，不能仍把对照称为未受处理。

回滚护栏优先处理确定性风险：功能输出错误、页面无法访问、意外 noindex、规范网址指错、敏感数据泄露、成本异常或严重性能回退。普通排名波动不应与这些事故共用“一下降就回滚”规则。回滚使用明确的已知良好版本，验证内容、配置和埋点都恢复，再记录后续影响。

Google 对网站实验要求避免伪装，并对变体网址、临时跳转和实验结束后的清理提供指导。[F17] 不能给搜索引擎单独展示“优化页”、给用户展示另一种内容来制造实验。

7.11 提交、抓取、索引、排名分别记录

Sitemap 提交成功只代表提交动作完成。Google 仍决定是否抓取与索引；lastmod 应对应实质内容变化，不应每天刷新所有页面日期。[F05] API 的 sitemaps.submit 使用写权限，与只读观测凭证隔离。[F18]

IndexNow 用于通知参与引擎 URL 的新增、修改或删除；一次 POST 可提交最多 10,000 个 URL，并要求站点托管密钥文件验证。HTTP 200 仅表明收到通知，不保证收录。[F19] 不把它写成 Google 通用索引接口；Google Indexing API 目前明确限于 JobPosting 或含 BroadcastEvent 的 VideoObject 页面，普通工具页与文章不在该使用范围内。[F20]

因此状态应分为 published、notified、crawl_observed、index_observed、search_observed。任何一步都不自动推出下一步。内容没有实质变化时，不循环提交以制造活动；新页还没有搜索数据时，不自动判定内容失败。

7.12 最低验收：让系统证明它会克制

上线反馈前，至少喂入以下测试数据：GSC 返回空行、GA4 抽样或截断、一个来源失败、未知搜索量、跨版本窗口、过短窗口、重复快照、历史数据修订、刚刚发布的页面、收入预估但成本已发生。预期结果应明确：暂停对应评估，保留原因，不擅自发布。

再做一个正向案例：数据完整且口径一致、任务量满足预先定义标准、冷却期结束、对照保留、建议仅改批准字段、证据全部可追溯。系统应输出可审核的变更提案，而不是“保证排名上升”。发布后验证线上版本，再回到快照流程。

闭环的价值在于每轮能减少一点不确定性。真实数据支持改进时，改得小而清楚；数据还不能说话时，保存现场并等待，也是一种正确执行。

一手来源

核验日期：2026-10-14。以下支持接口与测量边界；本章门槛、快照方案、决策表与算例是设计建议，不是供应商保证。

- [F01] Google, Using Search Console and Google Analytics data for SEO: <https://developers.google.com/search/docs/monitor-debug/google-analytics-search-console>
- [F02] Google Analytics, Avoid sending personally identifiable information: <https://support.google.com/analytics/answer/6366371?hl=en>
- [F03] Google Analytics, API dimensions and metrics: <https://developers.google.com/analytics/devguides/reporting/data/v1/api-schema>
- [F04] Google Analytics, Behavioral modeling for consent mode: <https://support.google.com/analytics/answer/11161109?hl=en>
- [F05] Google Search Central, Build and submit a sitemap: <https://developers.google.com/search/docs/crawling-indexing/sitemaps/build-sitemap>
- [F06] Google Search Console, Search Analytics query: <https://developers.google.com/webmaster-tools/v1/searchanalytics/query>
- [F07] Google, Performance data filtering and limits: <https://developers.google.com/search/blog/2022/10/performance-data-deep-dive>
- [F08] Google Search Console, Usage limits: <https://developers.google.com/webmaster-tools/limits>
- [F09] Google Search Console, URL Inspection index.inspect: <https://developers.google.com/webmaster-tools/v1/urlInspection.index/inspect>
- [F10] Google Analytics, runReport: <https://developers.google.com/analytics/devguides/reporting/data/v1/rest/v1beta/properties/runReport>
- [F11] Google Analytics, ResponseMetaData: <https://developers.google.com/analytics/devguides/reporting/data/v1/rest/v1beta/ResponseMetaData>
- [F12] Google Analytics, Data freshness: <https://support.google.com/analytics/answer/11198161?hl=en>
- [F13] Google Analytics, Data API limits and quotas: <https://developers.google.com/analytics/devguides/reporting/data/v1/quotas>
- [F14] Google AdSense, accounts.reports.generate: <https://developers.google.com/adsense/management/reference/rest/v2/accounts.reports/generate>

- [F15] Google AdSense, Metrics and dimensions: <https://developers.google.com/adsense/management/metrics-dimensions>
- [F16] NIST, Completely randomized designs: <https://www.itl.nist.gov/div898/handbook/pri/section3/pri331.htm>
- [F17] Google Search Central, Minimize A/B testing impact in Google Search: <https://developers.google.com/search/docs/crawling-indexing/website-testing>
- [F18] Google Search Console, Sitemaps submit: <https://developers.google.com/webmaster-tools/v1/sitemaps/submit>
- [F19] IndexNow, Documentation: <https://www.indexnow.org/documentation>
- [F20] Google Search Central, Using the Indexing API: <https://developers.google.com/search/apis/indexing-api/v3/using-api>
- [F21] Google, Search Generative AI performance reports: <https://developers.google.com/search/blog/2026/06/gen-ai-performance-reports>

08 先算清一批页面，再决定要不要扩大

内容站能否赚钱，需要分别回答三个问题：有人需要这些页面吗，网站能把访问变成符合政策的收入吗，收入是否覆盖持续成本？AI 可以降低部分生产成本，但也可能扩大审稿、纠错和维护负担。预算表应让这些代价可见，而不是只显示一次模型调用花了几分钱。

本章的数字全部是用于复算的情景假设，币种统一为人民币；不代表本项目真实流量、广告报价或收入。原型未加载广告脚本、未调用收费 API，也未接入结算账户，下面的经济核算属于运营设计，未与真实财务系统接通。

8.1 Page RPM 的分母是浏览量

AdSense 的 Page RPM 用估算收入除以页面浏览量，再乘 1,000。[E01] 因此，假设同一报告范围内有 50,000 次页面浏览，Page RPM 为 12 元，对应估算收入为 600 元。这里不能把 50,000 个访客直接放进分母，也不能用搜索展现替代页面浏览。

$$\text{Page RPM} = \text{广告估算收入} \div \text{同范围页面浏览量} \times 1,000$$
$$\text{广告估算收入} = \text{同范围页面浏览量} \div 1,000 \times \text{Page RPM}$$

广告展示量、广告请求量和页面浏览量也不是一回事。一页可能出现多个广告位，但不一定每个位都展示。广告 CPM 涉及广告展示的计价，不可直接当作站长的 Page RPM。[E08] 若采用实测 Page RPM，它已反映该报告范围内的实际收入结果，不能再随意乘一次填充率或广告位数，否则可能重复计量。

最好用广告平台同一报表的收入和页面浏览量复算 Page RPM。若拿 GA4 的 page_view 作分母，拒绝统计、脚本拦截、事件重复及采集范围都可能造成差异；可以另建自己的跨系统指标，但名称与口径应明确，不能仍称为平台原生 RPM。估算收入、最终调整收入与实际到账分开记，不要把尚未结算的数字写成现金。

8.2 一张能说明亏在哪里的月度账

先用“运营净贡献”判断项目是否值得继续。这里将它定义为观察期收入减去该期各项运营投入，包含人工机会成本；它不是财务报表上的标准利润科目，也未自动解决税务、折旧和收入确认问题。

项目	每月假设	怎样取得真实值
模型调用与返工	80 元	按请求记录计费单位，核对供应商账单
搜索／数据 API	100 元	按实际套餐、超额及税费核对

项目	每月假设	怎样取得真实值
主机、域名摊销、备份	60 元	账单分摊，包含恢复存储
编辑与事实审核	720 元	6 小时 × 120 元／小时
维护、投诉与错误处理	240 元	2 小时 × 120 元／小时
合计运营成本	1,200 元	现金 240 元，人工机会成本 960 元

假设第一批内容、工具和流程建设还投入 3,600 元，若内部管理上分六个月观察，每月再分摊 600 元。运营成本加建设分摊为 1,800 元。这个分摊期限是本例假设，不是通用会计处理；真正购买设备、计税和入账按当地规则处理。

情景	页面浏览量	假设 Page RPM	广告估算收入	扣运营成本	再扣建设分摊
较弱	20,000	6 元	120 元	-1,080 元	-1,680 元
中间	50,000	12 元	600 元	-600 元	-1,200 元
较强	150,000	18 元	2,700 元	1,500 元	900 元

这三行只展示算式对假设的敏感性，不说明任何情景更可能发生。固定假设 Page RPM 为 12 元且运营成本不随流量变化时，覆盖 1,200 元月运营成本需要 100,000 次同口径页面浏览；包含上述建设分摊则需 150,000 次。现实中流量结构、费率和成本都可能变化，所以这不是“做到十万浏览必定回本”的承诺。

再做两项反向检查。若审核时间翻倍，增加 720 元成本，原本较强情景的分摊后净贡献只剩 180 元。若较强情景的 Page RPM 从 18 元降到 9 元，收入变为 1,350 元，分摊后净贡献为 -450 元。能否持续赚钱，很可能比“每页生成成本”更依赖审核效率、流量质量与收入稳定性。

8.3 每页成本里，最容易漏掉的是以后

页面成本至少包含研究、资料权利核对、初稿、事实审核、功能测试、发布验收、后续更新和退役处理。假设生产一页需要模型及数据 2 元、人工审核 12 分钟；按每小时 120 元计，仅审核就要 24 元。再加半年内两次各 6 分钟更新，人工又增加 24 元。这时“2 元一篇”只描述了极小的一部分。

不要只分析有流量的页面。保留整批页面的投入和结果，包括未被索引、无点击、被合并、停止维护的项目。按发布队列观察，才能知道一批内容整体是否值得，而不会被少数成功页误导。某页具有支持、品牌或公共服务价值，也可保留；只是不要把这类价值强行伪装成已经赚到的广告收入。

收入若来自联盟、订阅或销售，应换算成扣除退款、平台费及履约变动成本后的贡献，再与 SEO 投入比较。归因报表分配给自然搜索的收入，并不自动是本次页面改动创造的增量。没有可信对照时，报告“观察到的净贡献”即可，不包装成因果 ROI。

8.4 广告接入有独立的验收门

达到某个页面数并不能保证广告审核通过。AdSense 资格说明强调原创、有质量的内容及政策符合性，并未给出本方案所谓“必须 1,000 页”的通用门槛。[E02] 是否开户、是否申请、是否接受条款及提供付款信息，都由站主在实际流程中处理；本交付没有执行这些动作。

申请前先检查：网站能否完整使用，关于与联系信息是否真实，隐私说明是否对应实际数据流，主要页面是否有独立价值，内容和图片是否有使用依据。先用没有广告的站点完成任务测试，再单独测试广告布局。广告不得伪装成工具按钮或下载入口；不能为了多展示一次广告故意把答案切碎、增加无意义分页，或让用户更难完成任务。

无效流量包括人为夸大广告点击或展示的行为。不要点击自己的正式广告，不购买机器人访问，不让员工或朋友“帮忙点一下”。[E03] 监测流量骤变时，先保留来源、时间和布局版本，核对推广活动及异常请求；不要让 AI 自作主张去“补正常流量”。广告政策告警属于人工处理事项，停止相关扩张并按官方流程排查。

Google 对欧洲经济区、英国和瑞士个性化广告有认证 CMP 要求；认证不等于网站已经全面符合法律。[E04] 因此生产上线清单必须明确目标地区、同意选择、撤回方式、标签实际行为和隐私说明。服务端上报、无 Cookie 模式或换一家广告供应商，都不能自动免除应有的义务。这里给的是工程检查框架，具体法律适用需专业核实。

8.5 不用牺牲完成任务来换收入

对时间戳转换工具，用户输入后立刻得到正确答案，是好结果。停留时间短不能单独解释为内容差；让用户多点三次才能看结果也不是改进。广告实验应同时观察任务完成率、错误率、遮挡、性能及投诉，并且先定义什么时候撤回。

试验	可以观察什么	暂停或撤回的例子
增加一个广告位	净收入、版面稳定、工具可用性	遮住输入、误触增加、核心结果被挤出
改写正文结构	找答案时间、理解错误、自然入口	内容失真或读者找不到前提
增加相关推荐	相关任务的真实使用	强行跳转、无关链接、阅读路径变长
页面提速	成功率、加载表现、维护成本	工具功能或可访问性退化

可访问性单独验收：键盘能操作输入与按钮，标签清楚，焦点可见，错误可理解，正文缩放后仍能阅读，颜色不是唯一提示。W3C WCAG 提供正式准则；本表只是面向当前工具的检查起点，不构成 WCAG 全面符合声明。[E05]

8.6 从 20—50 页试点开始，是资源建议

若主题确实容纳这些独立任务，可以用 20—50 个高质量页面作为第一批试点。这个范围是方便审核与维护的启发式建议，不是搜索、广告或模型的要求。只有一个真正有价值的工具和几篇必要说明时，就先做这些；没有足够资料时，不为凑数制造页面。

试点的任务是验证整条链路：需求不是凭空编造；资料可维护；模板不会系统性出错；人工审核负担可承担；页面公开后能正常使用；第一方观察可以取得；至少一次改进有完整前后记录。搜索样本不足时，可以继续观察或缩小商业假设，不能用假数据证明已经找到增长规律。

扩大时每次只增加一种主要变化，例如扩一个相邻任务，或多一种页面类型。不要同时扩国家、语言、实体和模板，否则一个故障很难定位。新批次必须复用通过验证的来源与质量检查，并保留抽样和高风险全检。前批仍大量积压审核或事实更新时，暂停新增。

下面的扩张门槛属于本项目初始策略，需要用实际表现校准，不是统计定律：

1. 所有准备公开的页面都有明确任务、资料来源和更新负责人；严重事实错误为零。
2. 没有未解决的安全、索引配置、工具失败或隐私阻塞。
3. 审核与维护时间在预算内，且没有靠放松标准压低时间。
4. 至少得到一类可信的需求或使用证据；有多少真实观察，就说到什么程度。
5. 较保守的经济情景仍在站主愿意承担的试验预算内；退出损失已被理解。

“是否继续验证”和“是否已经证明盈利”是两种决定。前者可以在预算有限、信息尚不完整时成立；后者必须有足够经营记录支持。

8.7 地区和语言扩展，先证明差异有用

先问地区差异是否影响答案。时间戳转换的基础算法通常不会因城市不同而变化；给每个城市生成一页几乎相同的转换器，没有自动产生新增价值。若差异来自实际时区、法规、单位、服务覆盖或资料来源，就要为它准备具体证据和维护能力。

每个新语言版本至少需要母语或足够熟练的审核者，核对术语、格式、导航、错误信息、隐私说明和支持渠道。Google 为语言地区版本提供独立 URL 与标记方面的指导，`hreflang` 帮助说明对应关系，但不能代替译文质量。[E06]

生产设计应使用“允许发布的组合表”，而非直接把城市 × 服务 × 语言的笛卡尔积全生成。每个组合记录独立任务、有效资料、真实覆盖范围、已审稿版本与维护人；任何关键项缺失，就保持草稿或不生成。避免以相似地区页导流到同一目的页的门页做法，也不通过买链接、批量互链站群或过期域名操纵排名。[E07]

8.8 学习要改进判断，不能偷偷改目标

每轮记录哪些预测被支持、哪些被反驳、哪些仍缺证据。把失败例子加入下一轮的离线评测：错误合并的意图、不可靠的来源、忽略季节变化的解释、没有发现的工具故障。高分但失败的案例尤其有价值，它说明排序规则需要校准。

校准可以从简单分组开始：把机会分为高、中、低，比较各组后续被人工采纳、真实使用和维护成本。样本不足时只调整明显不合理的规则，不拟合一组看似精确的权重。任何评分公式、模型、提示词或阈值的变化都留版本，并在未参与调整的样本上复测。

原型保留决策与机会反馈，用来演示“结果回到下一轮”。它不会自动修改基础模型权重、目标函数或审批规则。生产中的规则更新也应由负责人审阅具体差异；否则一个追求短期收入的系统可能逐渐绕开质量、隐私与预算约束。真正值得保留的学习，是更清楚地知道哪些动作有根据，哪些应该停止。

本章依据

- [E01] AdSense Page RPM，分母与公式，访问 2026-10-14。
- [E02] AdSense 资格要求，访问 2026-10-14。
- [E03] AdSense 无效流量定义，访问 2026-10-14。
- [E04] Google 对 EEA、英国和瑞士的 CMP 要求、隐私说明要求，访问 2026-10-14。
- [E05] W3C WCAG 概览，访问 2026-10-14。
- [E06] Google 多地区与多语言网站，访问 2026-10-14。
- [E07] Google 搜索垃圾内容政策，门页、链接与规模化滥用边界，访问 2026-10-14。
- [E08] AdSense CPM 定义，访问 2026-10-14。
- 经济表、试点数量、扩张门槛均为本书原创假设和建议，不是平台规定、实测收入或市场预测。

09 | 先跑通两个实验，再接真实世界

本章对应随书交付的 `prototype/` 源码。它解决一个比“自动写一篇文章”更具体的问题：一个建议从哪里来，怎样获得执行资格，执行后如何留下证据，结果不合适时能否恢复原来的页面。

这次实际跑通的是**完全离线、使用合成数据的本地预览闭环**。第一轮保留一个标题变化，第二轮撤销另一个标题变化。这里的保留与撤销由预设规则和样例指标触发，不代表真实用户更喜欢某个标题，更不是搜索排名、广告收入或因果实验的证明。

9.1 需要什么，运行后会得到什么

准备 Python 3.12 或更新版本，以及一个可写的专用目录。程序只使用 Python 标准库；不必安装依赖，也不必准备 API key。实际验收环境为 Linux、Python 3.12。代码没有把开发机的绝对路径写进导入语句，因此可以把整个项目复制到其他目录再运行。

进入 `prototype` 文件夹，执行：

```
python -m seo_loop demo --workdir ./demo-output
```

终端输出 JSON 报告；工作目录新增数据库、静态站和两份报告。第一次运行应看到两个终态决策：`cycle-1` 为 `keep`，`cycle-2` 为 `rollback`。`actions_used` 应为 2，`audit_valid` 应为 `true`，`live_seo_gain_measured` 应为 `false`。

再执行同一条命令。程序会识别已有的实验结果，跳过已完成的动作，不增加动作计数。这一步验证的是幂等：重复接到同一个任务时，不把同一项修改执行两遍。

静态站需要通过本地服务器查看：

```
python -m http.server 8000 --bind 127.0.0.1 --directory ./demo-output/site
```

打开 `http://127.0.0.1:8000/`。绑定地址限定在本机；用完后按 `Ctrl+C` 退出。站点包括解释秒/毫秒的指南，以及在浏览器内运行的时间戳转换器。输入不会被发送给统计脚本或广告系统。默认页面禁止索引，示例域是 `https://preview.example.invalid`；它不是已发布的网站。

构建器用同级版本目录和符号链接切换站点，避免读者看到新旧文件混合的半成品。Linux 已做实际测试。Windows 下符号链接可能需要开发者模式或相应权限；这部分没有实机验收。缺少权限时使用 Linux/WSL，而不是随意删掉一致性保护。

9.2 一次 demo 到底做了哪些事

演练时钟固定在 2026 年的合成时间窗口内。日期固定的目的是让重跑结果一致，不是让过去的批准在真实系统里继续有效。真实接入必须使用可信的当前 UTC 时间，不能由调用者随意回拨时间来延长批准。

1. 程序建立两个词种子和候选查询，保存用户任务、初始证据强度、竞争成本等合成特征。
2. 用初始权重计算机会分数，建立 `timestamps` 主题和两个页面计划。
3. fixture AI 角色对机会、页面计划和反馈提供结构化分析。输出留在数据库供审查；它们是可重复的模拟响应，没有发生真实模型调用。
4. 程序把预先写好的教学内容渲染成两个页面，保存内容版本哈希。
5. 第一轮先保存修改前指标，再产生一个 `update_title` 建议。通过规则检查后，生成仅限本地合成演练的批准记录。
6. 执行器再次核对批准、页面版本、证据、预算和冷却时间，只改标题与更新时间，并保存旧版本。
7. 程序读取第一轮修改后指标。CTR 相对变化超过样例门槛，任务成功率没有超过允许的下降幅度，于是保留。
8. 第二轮重复同样流程。指标触发撤销规则，程序恢复第一轮保留后的页面对象，并重新构建静态站。
9. 最后导出决策、机会反馈和审计快照。

这里的 AI 主导体现在多个分析节点共用结构化输入输出，而写入权限始终由确定性程序掌握。原型没有让模型从零搜索互联网、核验事实、写完整长文再自动生产发布。页面正文是教学脚手架；机会评分也是透明的固定公式。这样能独立测试决策和执行边界，不把“模型写出了看似合理的文字”混同于系统已经可靠。

初始机会分数为 $0.5 \times \text{task_fit} + 0.3 \times \text{evidence_strength} + 0.2 \times (1 - \text{competition_cost})$ 。这三个输入都是 0 到 1 的合成特征，没有查询真实搜索量。该公式仅用于演示排序如何进入后续流程；它不预测收入，也不是推荐给所有站点的通用评分器。

9.3 三条命令及其边界

命令	输入与动作	产物与验收	失败时怎么处理
<code>demo --workdir DIR</code>	使用固定合成数据，建立状态并执行两轮本地实验	两个终态决策、静态站、 <code>report.json</code> 、 <code>audit.json</code>	保留整个目录，查看错误，不要反复创建新目录掩盖问题
<code>status --workdir DIR</code>	读取状态并验证审计链	JSON 当前状态；不会自动恢复待渲染文件	审计链失败时停止后续操作，保留数据库调查
<code>recover --workdir DIR</code>	若 <code>render_pending</code> 为 <code>true</code> ，从数据库当前页面重新渲染	待渲染标记清除，页面与数据库恢复一致	修复磁盘空间、写权限或站点目录问题，确认并移除 STOP 后重试

这三条命令都没有生产发布、提交索引、配置广告、读取真实账户或调用付费模型的步骤。CLI 也没有通用“执行任意模型动作”命令。可选真实适配器由 Python 代码单独调用，接入前仍要经过本书第十章的权限和数据验收。

工作目录应只放本次演练内容。site 是指向同级发布目录的链接，不能只复制这个链接就当站点备份。要带走可直接阅读的静态文件，可以复制其指向的完整目录；要恢复整个实验，还要保留数据库与审计信息。

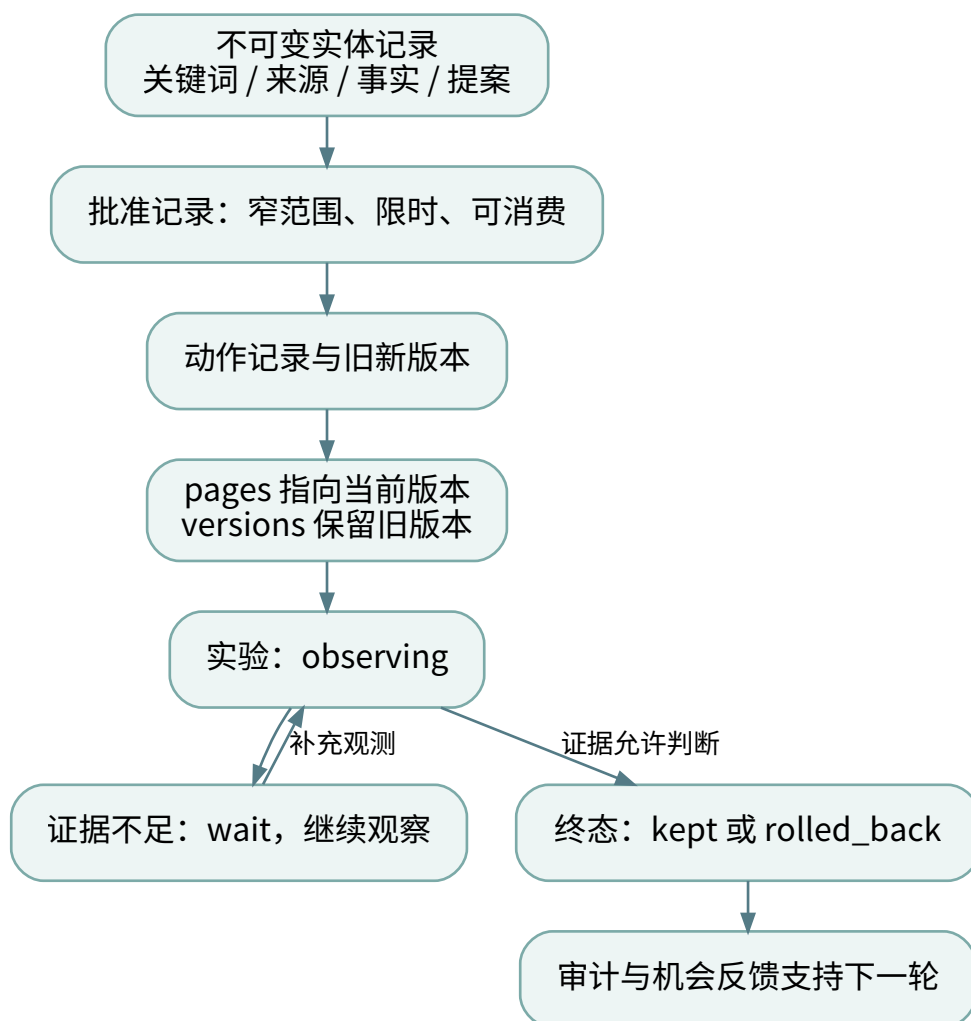


图 6 · 本地原型的记录、版本与实验状态。示意图；实施范围以本章正文为准。

9.4 数据库如何把动作连成一条证据链

SQLite 是本地权威状态，JSON 报告只是某一时刻的导出。数据库使用 WAL 日志和完整同步设置；引擎在工作目录持有操作系统文件锁，拒绝另一个引擎同时修改同一目录。进程异常退出后锁由操作系统释放，锁文件可以继续存在，不要仅凭“文件还在”判断任务仍在运行。

表	核心字段	用途
entities	kind, id, data, hash	不可原地改写的关键词、证据、事实、页面计划、建议、决策等 JSON 对象
pages	id, version, data	每个页面当前生效的对象
versions	hash, page_id, data	页面历史对象，供核对与回滚
metrics	id, revision, page_id, data, hash	同一观测窗口的不同数据修订版
approvals	id, proposal_hash, page_id, action, version, expires_at, scope, used_by	批准绑定与单次消费记录
actions	id, proposal_hash, page_id, before_hash, after_hash, status, data	执行过什么、改前改后是什么、目前结果如何
experiments	id, page_id, status, data	观察中或已终结的实验
audit	seq, event, at, data, previous_hash, hash	按顺序关联的事件哈希链
meta	key, value	动作预算、冷却时间、待渲染标记等运行状态

entities 中同一种对象和同一个 ID 如果出现不同内容，程序会拒绝覆盖。指标则允许增加 revision 保存修订；相同 ID、相同修订号却改变数据会被拒绝。哈希基于稳定排序的 JSON 计算，便于重复核验。

哈希链能发现部分不一致或意外篡改，不能替代外部可信日志。拥有数据库写权限的人也可能重建整条链。生产环境仍需权限隔离、异地备份、保留策略和独立审计；原型没有把 SQLite 变成不可伪造的公证系统。

页面状态的流转很短：动作记录先是 applied，相应实验是 observing。数据够用后，二者进入 kept 或 rolled_back。如果数据缺失、过旧或不足，判定为 wait，实验仍留在 observing；程序不会用零补齐，再装作观察到了下降。

9.5 模型建议为什么不能直接变成写入

执行器只认识如下字段，不能增加路径、脚本、任意 URL 或部署参数：

```
{
  "id": "cycle-1",
  "page_id": "timestamp-tool",
  "action": "update_title",
  "expected_version": "由当前页面计算的64位SHA-256值",
  "title": "Unix timestamp converter: seconds, milliseconds and UTC",
}
```

```
"evidence_ids": ["evidence-timestamp-tool"],
"hypothesis": "Explicit task and unit wording may help fixture users choose the relevant
page."
}
```

上面是字段说明，不是可直接执行的合法样例，因为 `expected_version` 必须是实际哈希。`examples/proposal.json` 是演练中导出的合法实例。

校验按几层进行：先确认严格 JSON 没有重复键、非有限数值或额外字段，再检查字段类型、ID 格式和动作白名单。标题必须是受限长度的纯文本；不能带标签、换行或空字符。随后读取数据库，确认页面存在、目标版本没有变化、证据属于同一页面且标明合成性质。

批准绑定的是整个建议内容的哈希，另保存目标、动作、版本、失效时间和范围。只要标题、假设或证据 ID 中任何一项改变，原批准都不能用于这份新建议。批准过期、已消费或指向别的页面时，同样停止。完全相同的动作 ID 重试返回已记录结果；相同 ID 配上不同内容则拒绝。

这些检查防止建议越权，但不能判断一个标题是否误导，或引用是否真的支持结论。模型分析里的“为什么”仍要接受编辑核验。真实发布前，应让具备权限的人看到目标、差异、证据和风险，并由可信服务签发批准；不要沿用 demo 自动生成本地批准的方法。

9.6 什么样的数据才允许作判断

核心指标对象包含 `id`、`page_id`、`version_hash`、`observed_at`、`period_start`、`period_end`、`final`、`revision`、`impressions`、`clicks`、`task_successes`、`task_attempts`、`synthetic`。时间必须带时区，次数必须是非负整数，点击不能超过展示，任务成功次数不能超过尝试次数。

原型把 `final` 当作上游已经完成的数据成熟度判断。在真实来源中，不同 API 对最终状态的支持不同；不能因为这个字段存在，就认为 GA4 天然提供同样的最终状态标志。真实适配层还要保留抽样、隐私阈值、分页截断、来源时区、查询过滤和数据修订信息。第七章说明了这些差异；核心演练没有把所有外部质量标志平铺进指标对象。

开始修改之前，程序要求修改前数据属于当前页面版本、标明最终状态、没有来自未来、窗口足够长且有足够样本。修改后评估还要确认两组窗口等长且不重叠、版本一致、修改后窗口不跨过修改时间。数据过旧、低样本、分母为零或仍在修订时，不自动保留或撤销。

这份初始策略使用：至少 100 次展示、20 次任务尝试、7 天窗口，数据窗口结束距判断时间不超过 14 天。同页动作间隔至少 7 天。保留要求 CTR 相对变化至少 10%，任务成功率下降不超过 2 个百分点。这些数字是**让演练分支可测试的初始门槛**，不是统计功效计算，也没有季节性校正或对照组。

因此，“达到门槛”只能说明这次合成输入经过了规则，不说明真实变化由标题造成。生产中的低流量站点可能需要更长观察窗口；快速修改也可能让归因更难。应预先选择主要目标、保护指标、最小可检测变化和停止规则，而不是见到数据后再挑有利门槛。

修改前指标若出现新修订，原实验暂停，要求重新建立基线并审查。已结束的实验不会被悄悄改写；如果后来的指标修订使当时依据变化，status 的 data_quality_alerts 会提示人工复核。这保护了历史决策的可解释性，也避免把“当时合理”混成“永远正确”。

9.7 故障演练：确认程序停在哪里

先运行全部自动测试：

```
python -m unittest discover -s tests -v
```

只运行核心测试时使用：

```
python -m unittest discover -s tests -p test_core.py -v
```

测试在临时目录中制造故障，不需要修改你的正式演练数据库。以下是核心用例和可观察的结果：

故障	用例	应看到什么
重复触发同一 workflow	test_double_run_idempotency	两次报告一致，动作仍为 2
同一动作 ID 换了内容	test_changed_id_is_rejected	拒绝，不覆盖旧动作
批准后改标题	test_approval_binds_content	哈希不一致，拒绝写入
批准过期／页面已变	test_expired_approval、 test_stale_page_version	拒绝陈旧批准或陈旧版本
缺少、暂定、过旧指标	test_missing_metrics_wait、 test_provisional_metrics_wait、 test_stale_metrics_wait	留在观察态，不伪造零值
样本过少／分母为零	test_low_volume_wait、 test_zero_denominators_wait	暂停判断，不出现除零或假结论
修改前数据被修订	test_revised_baseline_wait	要求重新审查基线
完成后数据被修订	test_terminal_revision_reported	历史结论不改写，报告新增告警
目录穿越、脚本、额外动作	test_unsafe_path、 test_shell_action_and_extra_fields	模式或白名单校验拒绝
证据不存在／格式错误	test_missing_evidence、 test_invalid_evidence_type_rejected	无法得到执行资格

故障	用例	应看到什么
冷却期、预算、停止开关	test_cooldown、test_budget、test_kill_switch	禁止新的实验动作
修改入库后构建崩溃	test_crash_recovery	保留待渲染标记，恢复后不重复扣预算
需要回滚	test_rollback_restores_old_artifacts	页面哈希和全站文件字节恢复到实验前
同目录启动第二个引擎	test_workdir_lock	第二个引擎停止，避免状态互相覆盖
审计数据被改写	test_audit_tamper_detected	验证失败，报告不能继续冒充完整证据链
重试一直超时	test_retry_bounded	达到次数上限后返回错误

停止开关也可以手动演练。在 demo-output 目录创建一个空文件，名称必须是 STOP。再运行 demo，程序应明确报告停止。确认后删除这个文件，再运行即可恢复。它阻止实验操作和待渲染恢复，并不会卸载已启动的 HTTP 服务；如果你还开着服务器，需要另外按 Ctrl+C。

恢复演练的核心问题是：数据库提交和文件写入不是同一个事务。原型先把希望生效的页面写入数据库，同时标记 render_pending=true；构建器完成原子目录切换后，再清除标记。若进程在中间退出，下次恢复会以数据库为准重新渲染。它不能解决生产 CDN、远程发布和搜索引擎缓存的分布式一致性；那些需要独立的发布状态与回滚协议。

重试只用于有明确幂等性的读取，最多五次。不要把这个工具函数套在付费请求、发信或结果不确定的外部写入上。真实适配器还应区分“确定未执行”和“可能已经成功但响应丢失”，再决定是否重试。

9.8 预算到底限制了什么

核心默认最多四个本地标题动作；演练用了两个。计数与动作入库在同一 SQLite 事务里完成，重复动作不会重复扣减。输入策略中的无穷值、NaN、负数和非法类型会被拒绝，避免门槛被特殊数值绕过。

这个预算没有覆盖真实 token 成本、第三方数据费用、服务器账单或历史站点版本占用的磁盘空间。可选 provider 的调用限制与核心动作预算是不同层。生产系统应将费用、请求额度、任务时长、输出数量和磁盘保留策略分开控制，并把“还有钱”与“还有权限”分别判断。

examples/policy.json 用于说明默认参数；CLI 不自动读取它。需要做程序级演练时，可以读取 JSON 并传给 Engine(workdir, policy=...)。不要改低门槛只为让实验出现“成功”。

9.9 已实测部分与仍然缺少的部分

已实测的范围是：离线运行、SQLite 持久化、严格对象校验、两轮本地实验、重复执行、版本与批准绑定、缺失／修订／低样本处理、停止开关、预算、受限重试、构建恢复，以及按字节核对的回滚。源码旁的自动测试和 `qa/core/` 保存了可复核的执行结果。

未实测真实账户登录、API 计费、真实数据导入、生产域名、DNS、CDN、索引提交、搜索收录、广告审批、收入或排名变化。真实适配器的 mock HTTP 测试证明的是接口处理分支，没有证明外部服务在你的账号上可用。没有真实事实语义核验、关键词市场需求验证、随机实验、统计显著性分析或全自动长文生产。

原型也不是多租户系统。虽然同目录锁能防止两个本地引擎同时写入，它没有队列、分布式锁、RBAC、审批签名、跨机器恢复、秘密管理或备份调度。不要把函数可导入理解成这些生产控制已经存在。

下一步应当从一份真实、只读、授权清楚的数据快照开始：保留来源与质量元数据，让规则给出“可判断”或“不能判断”，由人核对后再扩大范围。先验证这些连接是否可信，再考虑增加自动生成数量。

10 从本地演示走到可维护的真实站点

本交付能让你运行一条持久化的本地闭环，看见一次保留和一次回滚，并检查程序怎样拒绝没有依据的改动。它没有购买域名、连接真实账号、提交站点或发布生产。迈向真实运营时，先把数据接入、审批与部署分别验收，避免一次操作同时放开所有权限。

10.1 先划清交付边界

能力	状态	接下来还要验证什么
SQLite 状态、版本、审计、两轮合成演示	原型已实现	换业务输入后重跑测试，不推导真实效果
标题更新白名单、绑定 hash 的本地批准	原型已实现	正式身份认证与生产权限未实现
预览静态站、浏览器转换工具	原型已实现	真实浏览器、托管 HTTP 行为及无障碍人工验收
GSC、GA4 与模型适配器	配置后可接入	有账户授权才可调用；当前只做 mock 合同测试
AdSense 数据	文件导入接口	仅导入获准导出数据，未实现直接账户 API 接入
广告、真实收录、收入结算	未执行	独立申请、政策与隐私检查、真实观察
PostgreSQL、多机队列、RBAC、远程发布	生产扩展未实现	设计、实现、权限及故障演练

fixture 表示固定测试数据。它让相同输入可重复验证规则，不是当前市场采样。两轮演示的日期与数据均为教学构造；程序没有等待搜索引擎数周，也没有测得因果增长。生产使用前应把这种显著标注保留到所有报告和界面。

10.2 在干净目录复现结果

以下命令从交付包的 prototype 目录执行，使用 Python 3.12。工作目录选择一个专用的新目录，不能指向仓库根目录、已有网站目录或生产目录。代码不要求为了演示先安装第三方 Python 包；个别浏览器脚本测试对可用运行时的要求见第 9 章。

```
cd prototype
python --version
```

```
python -m unittest discover -s tests -v
python -m seo_loop demo --workdir ./runs/first-demo
python -m seo_loop status --workdir ./runs/first-demo
python -m seo_loop recover --workdir ./runs/first-demo
```

命令以零退出只是第一步。打开工作目录中的 report.json 和 audit.json，检查 mode 是 synthetic-local-preview、live_seo_gain_measured 为 false、两轮判定分别包含 keep 与 rollback、审计有效、没有待修复渲染。再用同一目录运行一次 demo，确认业务动作没有重复增加。报告字段与更完整故障操作见第 9 章。

本章编写期间在 Python 3.12.14 中实跑了 demo、status、recover；当时整套测试为 95 项通过。后续修订可能增加测试，以交付包最终 QA 记录为准。这次测试证明的是本地规则与合成路径，不包括真实 API、Google 收录、广告收入或长期并发负载。

预览可用 Python 自带的本地 HTTP 服务。它适合本机检查，不是生产服务器。[D01]

```
python -m http.server 8000 --bind 127.0.0.1 \
--directory ./runs/first-demo/site
```

在自己的浏览器打开 http://127.0.0.1:8000/，完成工具页与指南页检查，然后用 Ctrl+C 停止服务。不要将包含状态数据库的工作目录作为站点根目录，也不要本机服务绑定到公共网络来代替正式发布。预览中的 noindex 是防误索引标记，不是访问控制；需要保密的预发布站仍要有认证或网络隔离。

10.3 第一次接真实服务，先手动认清资源

先由站主在官方控制台确认资源和归属，记录资源 ID、用途、时区、货币及负责人。不要让模型根据一个域名猜是哪一个账号。接入顺序如下；每步失败就停在该步，不用后面成功的报告掩盖前面的错误。

步骤	输入与动作	产物与验收	失败时怎么做
确认站点	已授权域名与托管项目，检查正式域名	项目、域名、证书、回滚方法明确	不购买或替换域名来绕过问题
Search Console	站主按官方流程验证资源	页面检查结果与资源 ID 能对应	等验证解决，不能编造索引状态
GA4 或替代测量	先确定事件与隐私范围，再建立配置	合成任务的成功、失败、重复能区分	修理点，不用倍率“补齐”数据
只读报告	授予最小必要只读权限，导出小范围	原始响应、查询参数、时间与缺口齐全	权限或口径错误停止依赖分析
模型服务	确认供应商、数据用途和费用上限	小份非敏感样本可校验，预留痕	缺报价或权限就保持离线

步骤	输入与动作	产物与验收	失败时怎么做
广告服务	站主独立处理申请、条款和付款资料	政策、同意和布局验收通过	不在审核前假定收益

Search Console 的官方入门流程包含资源验证及重要页面检查。[D02] 站点地图提交只是发现渠道，提交成功不表示已经索引。URL Inspection 的读取也不是远程触发实时索引的承诺。把后台返回状态和实际取得时间原样记录，后续再据此检查。

凭据应由站主放到其批准的安全配置中。不要写进源码、示例 JSON、PDF、日志或共享工作目录。本书没有代填秘密，也没有创建长期访问令牌。可选 live 适配器的授权配置不等于允许任意账户动作；目前本地标题执行器还会拒绝真实数据直接进入合成实验。

10.4 发布经过四道门

生产发布器属于扩展设计。它应将构建与发布分开：先用批准的内容生成固定构建产物，检查完再把同一个产物交给部署端。不能审批一个 HTML 文件，发布时又让模型重新生成另一个版本。

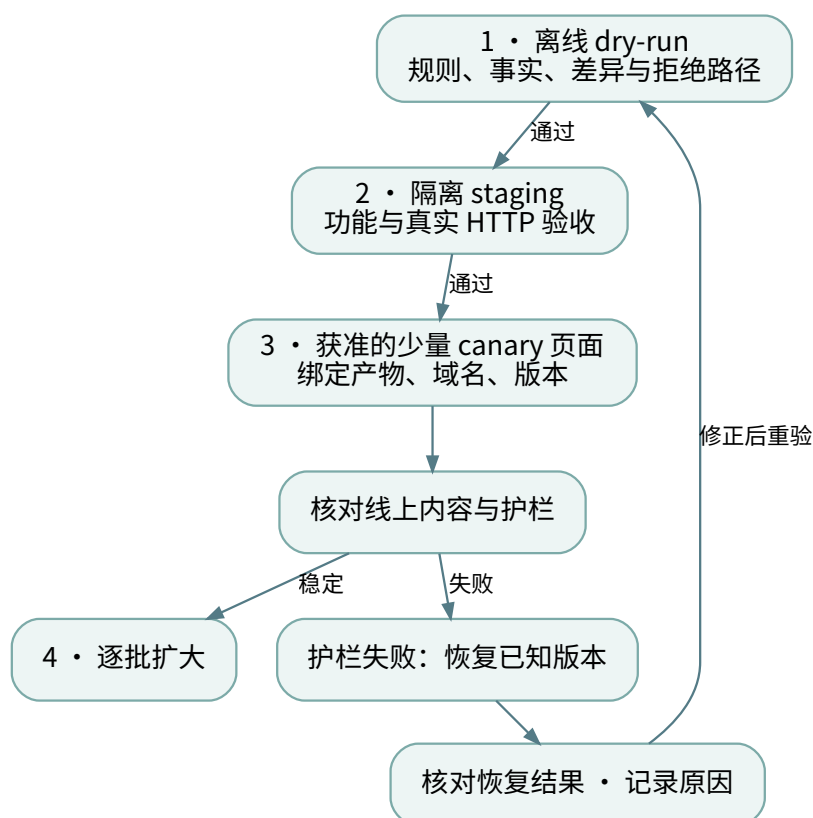


图 7 · 生产扩展的四道发布门。示意图；实施范围以本章正文为准。

第一道门是 dry-run。输出将修改的页面 ID、旧新 hash、差异、来源、成本、风险及恢复目标；对不认识的动作一律拒绝。检查应该拒绝的输入，和检查正常输入同样重要。

第二道门是 staging。使用隔离环境验证 HTTP 状态、链接、移动端、工具边界样例、404、canonical、sitemap、robots 和隐私标签行为。代码生成了某个文件，不代表托管平台一定以预期状态返回它。例如一个文件名叫 404.html，服务器仍可能把未知地址统一返回 200，必须在托管环境实测。

第三道门是小范围 canary，即只放行少量独立页面。数量按回滚范围和人工检查能力决定，本书建议初次 1—3 页作为操作起点，并非 SEO 规则。批准绑定明确页面、产物 hash、域名和版本；模板、全站导航及索引策略另行审批。先验证实际线上内容与构建产物一致，再开始观察业务指标。

第四道门才是扩张。严重事实错误、工具故障、私密数据泄露、错误 canonical、关键路径不可用等情况，先暂停扩张并处理事故，不必等待流量达到统计门槛。低流量、短窗口或普通排名波动则不触发机械回滚；它们需要第 7 章的诊断与判断。

恢复分两种。内容错误恢复上一已知版本，并验证工具及链接；隐私泄露还要处理缓存、日志、第三方传播及通知义务，单纯覆盖页面不等于问题已解决。数据库与外部发布不同步时，先核对实际状态，避免旧程序自动把已经修正的线上内容再次覆盖。

10.5 自动任务只做已获准的事

下面是一份可供站主审核后启用的 GitHub Actions 示例，当前没有上传或启用。它只检查离线演示，没有发布、提交索引或付费调用。示例用常见版本标签便于阅读；正式采用时应根据官方安全建议审查依赖，并固定到审核过的提交 SHA。[D03]

```
name: offline-seo-check
on:
  workflow_dispatch:
  schedule:
    - cron: '17 3 * * 1'
permissions:
  contents: read
concurrency:
  group: offline-seo-check
  cancel-in-progress: false
jobs:
  test:
    runs-on: ubuntu-latest
    timeout-minutes: 10
    steps:
      - uses: actions/checkout@v4
        with:
          persist-credentials: false
      - uses: actions/setup-python@v5
        with:
          python-version: '3.12'
      - name: Test and rehearse locally
        working-directory: prototype
        run: |
          python -m unittest discover -s tests -v
```

```
python -m seo_loop demo --workdir "$RUNNER_TEMP/seo-demo"
python -m seo_loop status --workdir "$RUNNER_TEMP/seo-demo"
```

此处 cron 使用 UTC，每周一 03:17。GitHub 的计划任务可能延迟，不能作为严格的实时调度保证。[D04] 它只是验证新代码仍能跑合成闭环。临时 runner 的数据库不会自然成为长期生产状态；要持续经营，另行选择持久运行环境、状态备份与互斥调度方案。

这个任务不需要云端 API 秘密，也没有写仓库权限。以后接只读报表时，把数据拉取与生产发布放到不同身份和任务里；只读分析任务不应拿发布凭据。部署审批应在可信系统中发生，不能把模型生成的“approved: true”当作批准。

若已有单机调度环境，先用本地计划任务运行测试或只读报告，记录开始、结束、退出码和数据新鲜度。避免两个任务同时写同一工作目录。原型使用工作目录锁；这不是分布式锁，也不能解决跨机器重复投递。调度器失约、数据库锁等待及连续失败都应有可见告警，而不是悄悄跳过。

10.6 用恢复演练证明备份可用

先停掉正在写入的任务，或者采用 SQLite 在线备份 API 取得一致备份。下面命令只做本地备份示例，在 prototype 目录执行。备份目录需要用户自行选择访问控制和加密存储；代码没有将它传到外部。

```
python - <<'PY'
from pathlib import Path
import sqlite3
source = Path('runs/first-demo/state.sqlite').resolve()
if not source.is_file():
    raise SystemExit('Source database does not exist')
backup = Path('runs/first-demo.backup.sqlite').resolve()
if backup.exists():
    raise SystemExit('Choose a new backup path; existing file preserved')
with sqlite3.connect(source.as_uri() + '?mode=ro', uri=True) as src:
    with sqlite3.connect(backup) as dst:
        src.backup(dst)
        result = dst.execute('PRAGMA integrity_check').fetchone()[0]
        if result != 'ok':
            raise SystemExit(result)
print('Backup integrity_check: ok')
PY
```

SQLite 的备份 API 用于复制数据库的一致状态。[D05] 不要在运行中的 WAL 模式数据库上只复制主文件，再假定所有事务都已包含。恢复演练在新目录操作，保留原件。注意 CLI 的 recover 只处理已经记录为 render_pending 的构建中断；它不负责从空目录恢复全部文件。只有数据库备份时，可在专用新目录显式重建本地预览：

```
python - <<'PYRESTORE'
from pathlib import Path
import shutil
from seo_loop.engine import Engine
source = Path('runs/first-demo.backup.sqlite')
```

```
target = Path('runs/restored-demo')
if not source.is_file() or target.exists():
    raise SystemExit('Need a valid backup and a new restore directory')
target.mkdir(parents=True)
shutil.copy2(source, target / 'state.sqlite')
engine = Engine(target)
try:
    engine.store.verify_audit()
    engine.render()
    report = engine.report()
    print(report['mode'], report['audit_valid'])
finally:
    engine.close()
PYRESTORE
```

核对恢复后的页面版本与决策，再用本地浏览器检查工具。此脚本只调用已实现的本地 Python 方法，没有远程发布。数据库备份不包含外部对象和秘密配置，因此生产还需要与之配套的构建清单、获准证据及配置版本。

为正式运营写下两个数字：最多能接受丢失多少时间内的新数据，以及多久需要恢复服务。这分别是恢复点目标和恢复时间目标。数值由经营需求决定；不能用“每天备份”替代回答。演练时实际计时并记录差距，至少覆盖数据库丢失、构建目录损坏和配置版本不匹配。

10.7 验收清单要包含拒绝和漂移

下面分为本地可测与上线后待测，不把未执行的验收写成通过。第 9 章提供更具体的测试定位；生产演练须在获准的隔离环境进行。

检查项	应出现的结果	验收层级
重跑同一 demo	不重复业务动作，审计仍有效	原型测试
提案没有证据或证据属于别页	拒绝，现有页不变	原型测试
审批后改标题或版本	拒绝旧批准	原型测试
指标缺失、未最终、分母为零	保持观察，不制造增长结论	原型测试
同 revision 的指标被改写	拒绝覆盖	原型测试
已结束决策的数据后续修订	提示人工复核，不偷偷重写历史	原型测试
动作超预算、冷却未结束、STOP 存在	停止受控动作	原型测试
HTML、路径和 URL 注入	转义或拒绝，不能执行外部指令	原型测试
发布中断与恢复	使用持久状态重建，不丢已登记变更	本地模拟；远程待测
正式 404、TLS、跳转与缓存	与路由合同一致	生产接入后待测
同意拒绝与撤回	标签实际行为符合已批准方案	生产接入后待测

检查项	应出现的结果	验收层级
备份恢复到另一环境	版本与必要证据齐全，恢复时间有记录	运维演练待测

漂移不只来自模型。API 字段改变、来源下线、产品规格更新、时区变化、页面模板变更，都可能使旧规则不再适用。安排固定的边界样本回归测试；新增来源先跑隔离合同测试；重新计算历史报告时写明版本与口径。模型升级后，至少比较严重错误、无依据断言、拒绝正确率、人工审核时间与总成本，不只比较文风。

10.8 用 30／60／90 天安排交付，不安排排名

下表是工作节奏示例。它假设有时间进行研究、开发、审稿和运营，具体按资源调整。任何阶段没有满足退出条件，都可以延长或缩小范围；90 天结束时没有稳定搜索流量，也不能据此断言项目必定失败或即将成功。

阶段	主要工作	进入下一阶段的证据
第 1—30 天	明确主题、资料权利、页面任务；跑本地闭环与工具验收；准备试点	能复现演示；真实内容可核查；范围与预算明确
第 31—60 天	经批准上线少量页面；接只读数据；建立口径与变更基线	技术与测量验收通过；已知缺口有人处理；有真实使用反馈或明确待观察项
第 61—90 天	完成一次有边界改进；核算成本；备份与权限演练；决定是否扩张	保留、撤回、等待或停止都有证据；交接人能复现操作

若第 60 天仍没有足够搜索样本，就先检查可发现性、功能、内容及需求证据，不强行做显著性报告。若用户任务能完成但广告经济不成立，可以评估其他商业方向，也可以保留有价值的内容并停止扩张。没有义务为了让自动系统一直运行而继续付费。

每周看的是阻塞与用户问题，每月看的是队列、成本和证据，每次版本更新看的是风险与恢复。长期记录应保存已经确认的事实、有效失败样例、实验结论及其限制；新模型只读取这些材料提出建议，不能擅自改变经营目标、预算上限和审批规则。

10.9 交给下一位维护者的最小包

最终交接至少包含：资源清单与负责人、代码和配置版本、数据字典、当前页面及来源清单、未完成实验、批准与发布记录、费用口径、异常处理步骤、备份恢复记录，以及已知未验证项。秘密通过获准的安全渠道管理，不夹在这个包里。

让接手人独立完成一次本地复现、一次来源纠错、一项待审核差异和一次恢复演练。若必须靠原作者口头解释才能完成，补文档或简化流程。一个真正可运营的闭环，应当在模型失败、来源过期和负责人休息时仍能安全停住，并让下一次继续有据可查。

本章依据

- [D01] Python 3.12 `http.server`，仅作为本地检查工具，非生产推荐。
- [D02] Google Search Console 入门，访问 2026-10-14。
- [D03] GitHub GITHUB_TOKEN 最小权限、安全使用 Actions。
- [D04] GitHub schedule 触发规则，访问 2026-10-14。
- [D05] SQLite 备份 API、Python 3.12 `sqlite3`。
- 原型合同：`prototype/seo_loop/cli.py`、`store.py`、`policy.py`、`engine.py`、`models.py`；测试：`prototype/tests/`。生产扩展与排期均为本书方案，未在真实账户实施。

附录 A：与前一本 SEO 教材的对应关系

此前的 SEO 教材提供原理与方法，本指南把它们落实为可执行合同、检查点和反馈记录。下面的对应关系便于回读；原理没有因自动化而失效，AI 也不能跳过资料质量与用户价值。

前一本教材主题	本指南的落点	可检查的产物
搜索流程、意图与质量	01、03、05	用户任务卡、抓取与索引分离状态
战略、找词、意图映射	03	候选池、地区设备口径、机会证据
竞对和搜索结果页分析	03	任务缺口与页面类型判断
长尾与程序化 SEO	03、04	独立页面价值检查、去重与合并规则
信息架构与站内优化	04、05	分类关系、内链、canonical、可读正文
内容战略与站点类型	04	事实版本、审核、工具任务验收
抓取、索引、渲染与体验	05	robots、sitemap、预览门槛、功能测试
结构化数据、迁移与审计	05、10	与可见内容一致的标记、发布回滚清单
国际化、其他引擎与本地搜索	03、04、05	真实地区差异、语言对应、通知不等于索引
链接与数字公关	03、10	基于有用资源的推广与许可，不自动群发或操纵互链
测量、实验与诊断	07、09	同口径窗口、修订记录、等待/保留/回滚
AI 搜索	01、03、07	结果形态记录、当前官方报告能力边界
商业化	08	RPM 分母、成本表、停止与扩张条件
AI 内容与工作流	02、04、06、09	结构化角色、证据引用、确定性执行器

需要随时间修订的两处知识

本次核验的官方材料对 Google 生成式 AI 可见性报告和 FAQ 富结果给出了较新的状态。相关说明已写入 01、05、07 章；不要把旧教材中的产品状态当作永久规则。原教材文件本次没有重编，阅读这些主题时优先采用本指南的日期限定与链接，并核对实际账号可用功能。

生产化仍需补齐的连接

最小程序把闭环控制、证据引用和版本恢复跑通；它没有实现所有教材技术的自动执行。以下工作要按第 10 章的阶段验收推进：真实关键词与搜索结果采集、允许使用的资料获取、语义核验与正文生产、真实账户指标、发布审批与部署、广告合规和真实实验设计。可以逐个接入，不能因为某个 SDK 能调用就默认整条链已经可靠。

附录 B：参考资料索引

本索引汇总正文直接引用的外部资料，核对日期为 2026-10-14。正文链接是具体主张的定位入口；索引不意味着同一来源支持书中所有工程建议。案例文章属于个人经验，其余资料尽量采用平台、标准组织和项目官方文档。接口实际可用性仍需真实账户验收。

逐项资料

1. 陈然的案例。引用章节：01。
2. Google 垃圾内容政策。引用章节：01, 08。
3. GA4 互动时间。引用章节：01。
4. 参与率与跳出率。引用章节：01。
5. Google 内容自评指南。引用章节：01, 04。
6. AdSense RPM 定义。引用章节：01。
7. 申请条件。引用章节：01, 08。
8. 实行政策。引用章节：01。
9. 无效流量。引用章节：01, 08。
10. 同意管理要求。引用章节：01, 08。
11. 美国版权局说明。引用章节：01。
12. 美国版权法第 105 条。引用章节：01。
13. ICO 个人信息指南。引用章节：01。
14. Google Indexing API 范围。引用章节：01, 07。
15. IndexNow FAQ。引用章节：01。
16. 新版指南。引用章节：01。
17. 参与控制。引用章节：01。
18. 报表口径。引用章节：01。
19. GitHub 大文件说明。引用章节：01。
20. Hobby 规则。引用章节：01。
21. Vercel 限制。引用章节：01。
22. SQLite：适用场景与并发边界。引用章节：02。
23. OWASP：LLM 提示注入防护。引用章节：02。
24. Google：规范 URL。引用章节：02, 05。
25. 美国版权局：保护范围。引用章节：02。
26. Search Analytics query。引用章节：02, 06, 07。

27. SQLite 备份 API。引用章节：02, 10。
28. Python 3.12 sqlite3。引用章节：02, 10。
29. PostgreSQL：备份与恢复。引用章节：02。
30. RFC 9309。引用章节：04。
31. W3C PROV-DM。引用章节：04。
32. OpenAI Responses API。引用章节：06。
33. Structured Outputs。引用章节：06。
34. GA4 runReport。引用章节：06, 07。
35. AdSense reports.generate。引用章节：06, 07。
36. URL Inspection。引用章节：06, 07。
37. Search volume Live。引用章节：06, 03。
38. SERP Live Advanced。引用章节：06, 03。
39. AdSense Page RPM。引用章节：08。
40. 隐私说明要求。引用章节：08。
41. W3C WCAG 概览。引用章节：08。
42. Google 多地区与多语言网站。引用章节：08。
43. AdSense CPM 定义。引用章节：08。
44. Python 3.12 http.server。引用章节：10。
45. Google Search Console 入门。引用章节：10。
46. GitHub GITHUB_TOKEN 最小权限。引用章节：10。
47. 安全使用 Actions。引用章节：10。
48. GitHub schedule 触发规则。引用章节：10。
49. K01 Google, Search Console performance data filtering and limits。引用章节：03, 07。
50. K03 Google Ads, About Keyword Planner forecasts。引用章节：03。
51. K04 Google Trends, FAQ about Google Trends data。引用章节：03。
52. K06 Google Search Central, AI features and your website。引用章节：03。
53. K07 Google, Introducing Search Generative AI performance reports in Search Console。引用章节：03, 07。
54. T01 Astro, On-demand rendering。引用章节：05。
55. T02 Next.js, Static Exports。引用章节：05。
56. T03 Google, JavaScript SEO basics。引用章节：05。
57. T04 Google, Block indexing with noindex。引用章节：05。
58. T05 Google, Introduction to robots.txt。引用章节：05。
59. T07 Google, Build and submit a sitemap。引用章节：05, 07。

- 60. T08 Google, [Pagination and incremental page loading](#)。引用章节：05。
- 61. T09 Google, [Managing crawling of faceted navigation](#)。引用章节：05。
- 62. T10 Google, [HTTP status codes](#)。引用章节：05。
- 63. T11 MDN, [304 Not Modified](#)。引用章节：05。
- 64. T12 Google, [Localized versions](#)。引用章节：05。
- 65. T13 Google, [Documentation updates, 2026 年 5 月与 6 月 FAQ 更新](#)。引用章节：05。
- 66. T14 web.dev, [Web Vitals](#)。引用章节：05。
- 67. T15 Google, [Indexing API quickstart](#)。引用章节：05。
- 68. T16 IndexNow, [Documentation](#)。引用章节：05, 07。
- 69. F01 Google, [Using Search Console and Google Analytics data for SEO](#)。引用章节：07。
- 70. F02 Google Analytics, [Avoid sending personally identifiable information](#)。引用章节：07。
- 71. F03 Google Analytics, [API dimensions and metrics](#)。引用章节：07。
- 72. F04 Google Analytics, [Behavioral modeling for consent mode](#)。引用章节：07。
- 73. F08 Google Search Console, [Usage limits](#)。引用章节：07。
- 74. F11 Google Analytics, [ResponseMetaData](#)。引用章节：07。
- 75. F12 Google Analytics, [Data freshness](#)。引用章节：07。
- 76. F13 Google Analytics, [Data API limits and quotas](#)。引用章节：07。
- 77. F15 Google AdSense, [Metrics and dimensions](#)。引用章节：07。
- 78. F16 NIST, [Completely randomized designs](#)。引用章节：07。
- 79. F17 Google Search Central, [Minimize A/B testing impact in Google Search](#)。引用章节：07。
- 80. F18 Google Search Console, [Sitemaps submit](#)。引用章节：07。

随书代码与设计材料

- `prototype/seo_loop/`：本次编写的实现，包括状态、策略、反馈、站点生成、资料与接口模块。
- `prototype/tests/`：作者测试和独立复核测试；运行命令见第 09 章。
- `prototype/examples/`：明确标注的合成数据、模式与资料例子。
- `prototype/seo_loop/prompts/`：角色约束与结构化输出格式。
- `research/interfaces.json`：接口口径、已知限制与未知项。
- `readable-writing/`：用户提供的写作技能副本，保留原文。

引用与验证边界

书中用链接支持定义、接口和平台规则，设计取舍由正文解释。样例测试支持代码在列明环境中的行为，不支持真实排名或收益承诺。没有把搜索摘要、匿名经验帖或未经验证的收入截图当作规则依据。