

翟羽京

AI 产品经理 · 作品集

用「先定义什么算好，再动手」的方式做 AI 产品

求职方向：AI 产品经理（杭州）

数字媒体技术 · 本科 · 应届

本作品集所有结论均可在对应文档中核验，不含无法溯源的数字

一句话定位

我不是最懂模型的人，但我是那种会先把「什么叫好」定义成可判定的标准，再动手的产品经理——并且会用实测去推翻自己的假设。

我证明三个能力

① 定义力 · 把模糊的「好」变成可执行的标准

从零搭评测体系：把「内容合格」拆成 6 条硬性约束，判定哪些能机器判、哪些必须人判，定出阈值与一票否决项。为什么只有两条配得上一票否决——因为它们的误放代价不可逆。

证据：30 条评测用例 · 可运行的判分脚本（4 项自测通过） · 判分口径一页速查

② 判断力 · 从证据反推产品决策

不看用户数据，而是从产品实现反推决策：为什么 AI 粗剪被放在免费额度里、这个商业约束如何倒推出技术架构的形状；并指出团队最大的数据缺口是没埋「结果是否被采纳」。

证据：全文划分「硬证据 / 无证据 / 推断」三档，附证据索引表并逐条标注强度

③ 落地力 · 真跑通过，不是画图

用 n8n + 飞书多维表格搭了三个工作流，以状态字段当队列、各管一段、独立失败，选题与脚本走人工审批闸门。另有一个打包成 APK 装到真机的 App。

证据：可导入的 n8n 工作流 JSON（WF1 已跑通） · 习惯 App 的 APK + 真机截图

三个项目是一条真实链路

03 · AI 内容生产 Agent（选题+脚本，文字） → 02 · 数字人内容流水线（n8n 编排 + 人工闸门） →

05 · AIGC 角色资产（画面与一致性） → 真实在运营的账号（发布 → 数据 → 归因）

建议的阅读顺序

1. 剪映 AI 粗剪产品分析（判断力的最强证据）
2. 判分口径速查（一页看懂评测怎么判）
3. 判分器校准记录（实测抓出并修掉 5 个判分问题）
4. 评测集设计说明（30 条用例怎么构造）

剪映 AI 自动粗剪 · 产品分析报告

剪映 AI 自动粗剪

产品分析报告（基于产品实现反推）

剪映 AI 自动粗剪 · 产品分析报告

阅读须知：本报告的证据边界

本报告不是基于用户数据或市场调研的产品分析，而是从产品实现反推产品决策的分析。

我能拿到的（硬证据）：埋点口径与参数枚举、功能规格与上限值、用户可见文案、付费权益名单、交互状态机、远端可调参数、真实运行日志。

我拿不到的（无证据）：留存率、转化率、用户访谈、市场规模、竞品内部数据。

因此：凡涉及功能规格与设计意图的结论，均有逐字出处；凡涉及用户画像、竞品对比、优先级建议的，均为基于产品逻辑的推断，已在文中逐处标注。

一、结论摘要

剪映的 AI 自动粗剪不是一个功能，而是一个底层能力 + 两种产品化形态 + 一条备用路线：

底层能力："AI 识别语音转文字，用户直接删文字来剪辑"（改字即改片）

自动版（产品一）：用户写文稿，AI 自动判断该删什么 → "智能解说粗剪"

手动版：用户自己看着文字稿删（这是用户本来就熟悉的功能）

备用路线（产品二）：不靠文字，靠 AI 看图 → Hub Agent 粗剪 rough_cut

关键澄清：这三者都不是"AI 生成视频"。它们做的是筛选与重排用户已有的素材——画面是用户拍的，声音是用户录的，AI 只负责挑、排、对齐。判据很简单：这类功能只会让视频变短，不会变长。

产品上做对的核心一件事是：

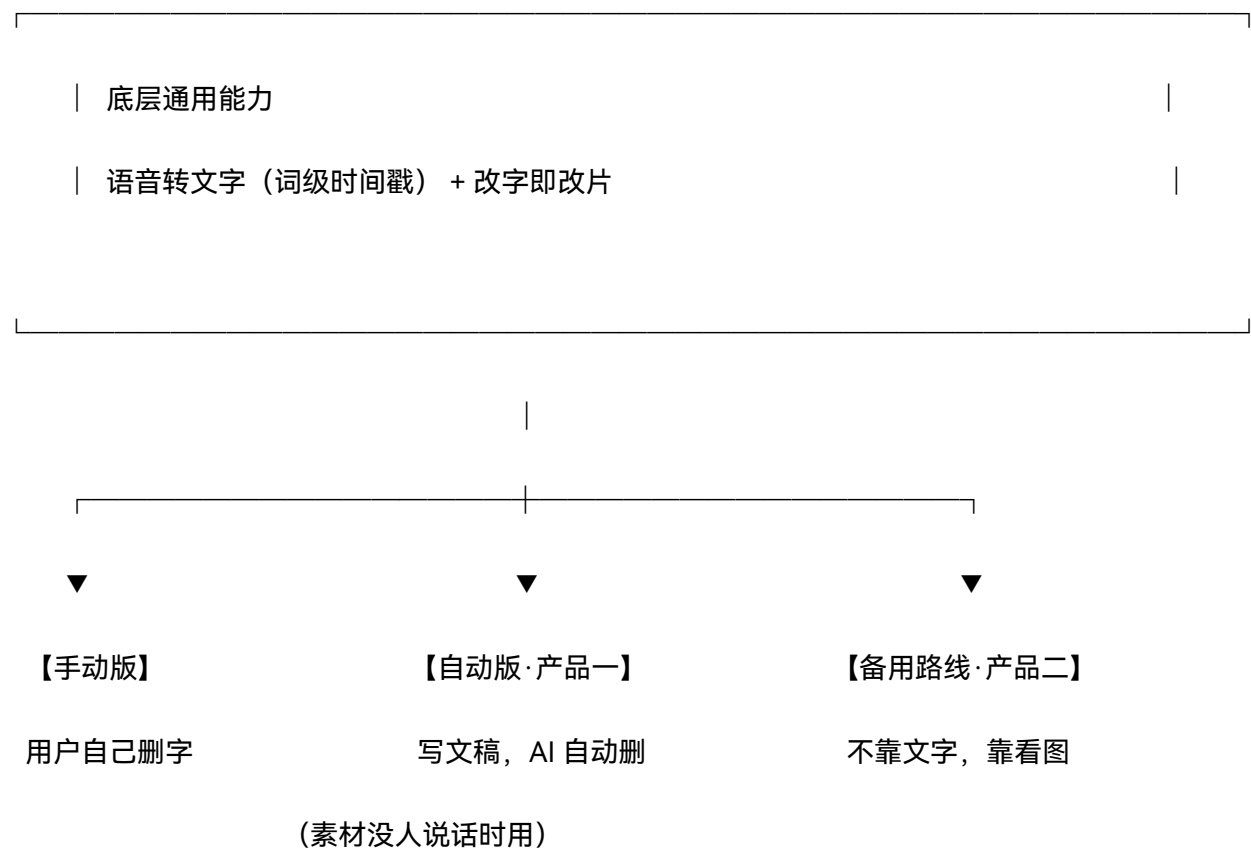
把 AI 定位成"做初稿的实习生"，而不是"替你做决定的导演"。

这个定位靠三条设计兑现：不发明事实、不破坏已有、不擅自定稿。而最大的商业约束是：核心能力被放在免费额度侧，因此必须在极低成本下跑得起来——这一条约束直接决定了技术架构的形状。

证据说明：功能规格、埋点口径、文案、上限值、付费名单均为逐字读到的硬证据；用户画像、竞品对比、优先级建议为基于产品逻辑的推断，已在文中逐处标注。

二、产品定义：同一能力的两种产品化形态，加一条备用路线

这是本次分析最重要的结构性发现。但先纠正一个说法："两个产品"是我的分析结论，不是官方口径。更准确的表述是这个结构——



产品二名称的实测证据：canvas_agent\app\index.html 的标题栏为 <title>剪映 Hub</title>。

产品一名称的实测证据：应用内帮助按钮指向的指南标题为《「智能解说粗剪」使用指南》；主按钮文案为"开始粗剪"。

证据强度提示：产品一有完整的页面文件、按钮文案、埋点与状态机支撑，可靠度较高；

产品二目前只见标题、后端路由与工具链，我没有见过它的实际交互界面——该行的"入口"与"发起方式"属推断，待界面验证。

产品判断：两者不是同一个功能的两个版本，而是两种不同的产品假设。

产品一赌的是：用户其实缺的是解说词，不是剪辑——给你文案，你改字，画面自动跟着走。

产品二赌的是：用户有大量素材，缺的是帮我挑——给意图，AI 帮你选片排序。

从成熟度看：产品一明显更完整（5 个专用埋点、完整状态机、引导文案、远端可调阈值）；产品二目前主要靠 Agent 的通用埋点覆盖，缺少产品级专用埋点。

第一个产品问题：这两个是要收敛成一个体验，还是显式分成两个入口？目前"两个都在做、但用户看不出区别"是最差的状态。

三、用户与场景

3.1 谁在用（从产品约束反推）

3.2 关键洞察：两种形态的输入门槛差异巨大

产品一的输入是文稿——所有受过教育的人都会的东西。

产品二的输入是视觉描述（例如"找主角大笑、背景虚化的片段"）——这是专业剪辑师的语言，普通用户描述不出来。

这是一个被低估的产品门槛。产品二如果直接面向小白用户，会卡在"用户不知道该描述什么"。

建议（推断）：产品二需要降低描述门槛——提供意图模板，或由 AI 主动提问澄清（"你是想要搞笑合集，还是干货总结？"），而不是要求用户自己写出画面描述。

四、用户任务（JTBD）

用一句话概括两个产品各自完成的任务：

产品一

当我有一条已经想好要说的话时，帮我把它变成有画面的视频。

产品二

当我有一大堆拍好的素材、但没时间一个个看时，帮我挑出该用的。

注意这两个任务是互斥的起点：一个有内容没画面，一个有画面没内容。这解释了为什么它们必须是两个产品——不存在谁替代谁。

五、功能规格拆解

5.0 先澄清：产品一不是"AI 生成模型"

这是最容易被误解的一点，但它决定了整个产品的承诺边界。

产品一由三个组件构成，没有一个是"生成"的：

它做的事是"挑选与重排已有素材"，不是凭空造画面。

一个可以用来验证的判据：产品一永远只会让视频变短。

因为它只能"丢素材"，不能"造素材"——删掉一个"嗯"，长度就少 0.4 秒。

"配音解说"的解说词哪来的？来自用户自己写的文稿。AI 只负责把画面按文稿排好。另外那个"生成解说词"按钮确实用了生成模型，但那是写文案，不是生成视频，而且是可选的——用户完全可以自己写文稿，一步都不碰生成模型。

5.1 底层机制：这套能力用户其实已经很熟悉

产品一建立在剪映已有的一个通用能力之上——即"AI 识别语音转文字，用户直接删文字来剪辑"。

导入视频

↓

语音识别 → 转成文字，每个字带时间点

↓

屏幕显示整篇文字稿

↓

用户删掉"嗯""那个""这段废话"

↓

画面自动跟着剪掉

底层实现是"删除标记", 不是真删:

AttachmentAiClipperTranscript

└ Chapter / Paragraph / Segment / Sentence / Word / DeletionSpan

↑ 删除标记在这

操作: Delete / Restore (恢复) / Split (拆分) / Trim (修剪) / Unmark (取消标记)

有 Restore 和 Unmark 存在, 说明删除是可逆的——对应界面上能撤销的体验。

代码里的文件名直接暴露了这层关系:

\service\ai_edit\rough_edit\ai_clipper_transcript_edit_helper.cpp

\lyra\service\transcript_edit\transcript_edit_service.cpp

rough_edit (粗剪)、transcript_edit (转写编辑)、ai_clipper (AI 剪辑) 出现在同一块代码里——官方自己也是把它们当一件事做的。

5.2 产品一 = 上面那套能力 + 一个"自动决策层"

用户素材

↓

【已有能力】语音转文字 + 改字即改片

↓

【新增一层】用户写文稿 → AI 判断素材里哪些话该留、哪些该删 → 自动划一遍

↓

用户拿到已粗剪的初稿, 再手动微调

这个结构解释了一个关键约束：产品一依赖语音转文字，所以素材里没人说话时它就失效——没有字可删，机制转不动。这正是产品二（视觉路线）存在的理由。

5.3 产品一：智能解说粗剪

核心交互：一个文稿编辑器 + 一次生成动作。

值得注意的细节：结果追加而非替换，且提示语明确告知这一行为。

这解决了 AI 产品的核心信任问题：用户怕"AI 把我的东西弄没了"。追加而不是覆盖，等于告诉用户"你的原文还在"。实现成本几乎为零，但极大降低尝试心理门槛。

字数限制的运营含义：10 / 8000 这两个值都是远端可调的，产品经理不需要等客户端发版就能调整。这是把产品调优从"发版"里解放出来。

5.4 产品二：Hub Agent 粗剪

核心交互：对话式。

重要的产品结构发现：包装被拆成独立阶段，且设计原则是非破坏性。产品规范原文要求：

保持基础时间线、素材顺序、源区间和内容主线稳定。

不使用 replaceChildrenXml 重写 main；不新增、删除、替换或重排 main 下的主素材。

产品解读：这是一个产品承诺，不是技术限制。

它的意思是："AI 帮你加字幕配乐，但绝不动你剪好的画面。"

为什么要承诺这个？因为用户对 AI 的第二大恐惧（第一是"弄丢我的东西"）是"AI 乱改我的作品"。

把"粗剪"和"包装"分开、并承诺包装不动主体，等于把 AI 的能力切成用户可以单独授权的模块。用户可以说"帮我加个字幕"，而不用担心画面被改。

六、商业模式与付费墙设计

这是报告中证据最硬的部分——服务端下发的权益名单直接暴露了产品策略。

6.1 付费墙的位置

核心判断：AI 粗剪被放在免费额度侧，说明它是获客与留存的钩子，不是利润中心。

这是很典型的产品策略：用高频基础能力做规模，用高级美感能力做变现。

6.2 变现链路

用户用免费粗剪快速出片（建立习惯）

↓

成片要好看 → 需要调色 / 智能裁剪 / 美颜 → 遇到 VIP 墙

↓

导出时 → 水印 / 高清 / 时长限制 → 二次变现

这就是为什么成本是整个架构的设计约束：一个放在免费侧的功能，如果单次调用成本很高，规模越大亏得越多。

所以“极低分辨率的分镜检测模型”、“每秒抽一帧”、“每批最多 6 项”、“深档限制 30 秒”——这些技术选择背后是同一个商业约束：它必须在免费额度下跑得起来。

产品视角的关键结论：技术方案不是技术团队自己选的，是被商业模型逼出来的。

七、关键产品决策与取舍

决策 1：AI 只做初稿，不做定稿

证据：产品规范反复强调“提交后的素材、顺序和源边界作为方案交接，不代表用户已确认”；主 Agent 侧明确“不自动增加确认门禁”。

点评：这是一个有克制的定位。好处是体验顺畅、不打断；风险是用户可能不知道自己拿到的是初稿。

建议（推断）：产出后应明确告知这是可修改的初稿，并给一个直接进入修改的入口，而不是停在结果页。

决策 2：结果追加而非覆盖

点评：教科书级的好决策。降低尝试成本，实现代价几乎为零。

决策 3：不设确认门禁

点评：双刃剑。

好处：流畅、少打断、符合“AI 帮我干活”的期待

风险：AI 理解偏了但用户没察觉，一路跑到导出

建议（推断）：考虑按"破坏性"分级确认——加字幕不确认，重排整个时间线给一次确认机会。

决策 4：包装与粗剪分离

点评：非常好的产品分层。让用户能"只授权一部分 AI 能力"，控制感强。

决策 5：埋点设计得细

从埋点参数能看出产品团队关心什么：

特别值得注意的是 `apply_time` 与 `expected_time` 这一对——它在测"用户等得烦不烦"。

这是成熟团队的标志：不只看"功能有没有被用"，还看"用的时候体感好不好"。

决策 6：关键参数远端可调

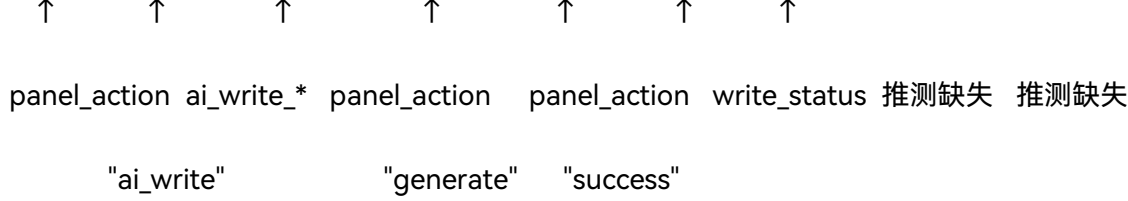
字数上下限、动态配置项、开关项均为远端下发。

点评：把"产品调优"从"发版"里解放出来，这是数据驱动迭代的基础设施。

八、数据体系分析（反推北极星指标）

从埋点可以还原出产品团队当前观测的漏斗：

看到面板 → 打开浮层 → 输入/生成文稿 → 点击"开始粗剪" → 生成成功 → 应用到草稿 → 用户保留



他们埋了：打开、生成文稿、发起粗剪、生成结果（成功 / 失败 / 取消）、耗时。

我没找到（可能缺失）：结果是否被用户保留。

这是我认为最大的数据缺口。

"生成成功"不等于"用户满意"。用户完全可能生成完、看了一眼、全删了。

真正的北极星应该是"AI 产出被采纳 / 保留的比例"，而不是"生成成功率"。

建议：补一个"AI 粗剪结果被采纳"的埋点（例如：生成后一定时间内未撤销，或用户在此基础上继续编辑）。

九、差异化定位（推断）

本节证据相对最弱，只能从产品设计反推差异化意图。

核心差异化判断：剪映没有走"一键生成爆款视频"的路线，而是定位在"帮你把初稿做出来"。

这其实是一个更聪明也更保守的定位：

不承诺"做出好视频"——做不到，会失望

承诺"省掉最枯燥的部分"——能做到，可交付

十、风险与待验证假设

10.1 产品风险

10.2 待验证假设

假设：用户更愿意"改文稿"而不是"拖时间线"

验证：对比两种路径的完成率与耗时

假设：用户能接受"AI 只给初稿"

验证：初稿采纳率、后续手动修改量

假设：免费粗剪能带动 VIP 转化

验证：使用粗剪用户的 VIP 转化率 vs 未使用用户

假设：包装功能的独立入口会被使用

验证：包装阶段的发起率

十一、行动建议（按优先级）

P0 — 补上"采纳率"埋点

现在的数据只能看到"生成成功"，看不到"用户满意"。没有这个指标，后续所有迭代都是盲的。

P1 — 明确两个形态的关系

产品一、产品二与底层的"文本剪辑"能力是什么关系，用户侧完全看不出来。当前的"两个都在、用户不知道区别"是最差状态。

P1.5 — 把"不是生成"这件事讲明白

用户很容易把"AI 剪辑"误解成"AI 生成视频"。产品一实际是挑选与重排已有素材（只会变短，不会变长）。这个边界不讲清，用户会期待"AI 变出好画面"，然后失望。主动设定期望，比让用户自己撞墙好。

P2 — 降低产品二的描述门槛

提供意图模板，或由 AI 主动提问澄清，而不是要求用户自己写出画面描述。

P3 — 给高成本操作加分级确认

按"破坏性"分级，而不是一刀切不确认。重排整个时间线值得一次确认，加字幕不值得。

P4 — 明确告知"这是初稿"

产出后告诉用户"这是初稿，可以改字 / 换镜头"，并给直接进入修改的入口。

十二、一句话总结

剪映的 AI 粗剪，产品上做对的核心一件事是：把 AI 定位成"做初稿的实习生"，而不是"替你做决定的导演"。

它靠三个设计兑现这个定位：

1. 不发明事实——时间点必须来自工具返回的真实数据，禁止 AI 编造
2. 不破坏已有——结果追加、包装非破坏、改动可撤销
3. 不擅自定稿——产出是方案，改稿权在用户

附录 A：技术底座（只列与产品判断直接相关的部分）

A.1 三个独立引擎

注意：安装目录下有个名为 smart_cut.ofx 的插件，名字含 "cut"，但它实际是智能裁剪（画面重构），不是粗剪。这是本次分析中最容易误判的一处。

A.2 三个时间单位——为什么分层

产品一能精确到词，基础在于时间单位被严格分层：

产品含义：剪辑要做到"一帧都不能差"（一帧约 33 毫秒），所以存储必须用微秒保证不丢精度；而对 AI 和用户，微秒不可读，所以对外用毫秒。

A.3 成本约束如何塑造架构

这一节是"商业模型决定技术架构"的直接证据。

附录 B：证据索引

维度	产品一：智能解说粗剪	产品二：Hub Agent 粗剪
承接页	H5 卡片（推断）	剪映 Hub（标题栏实测）
名称	"智能解说粗剪"（应用内自证）	内部代号 rough_cut
主 CTA	"开始粗剪"（实测文案）	Agent 对话式发起
输入	一段文稿	素材池 + 意图描述
本质	文案驱动的解说生成	素材驱动的语义剪辑
技术底座	端侧语音识别 + 转写编辑	云端多模态大模型
联网要求	生成解说词需联网，剪辑在本机	全流程必须联网
失效场景	没有口播的素材	云端不可用；素材无语音且无语义

用户类型	核心需求	产品一	产品二
口播 / 知识博主	一稿多剪、快速出片	✅ 主场	⚠️ 可用
访谈 / 播客	去口水词、精修	✅ 主场（词级精度）	⚠️ 可用
电商 / 带货	快速切卖点	⚠️ 弱（需要文稿）	✅ 主场
vlog / 旅拍	从大量素材挑精华	❌ 不可用（无主导语音）	✅ 主场
企业宣传 / 课程	长视频切短视频	✅ 可用	✅ 可用

组件	类型	做什么
语音识别（ASR）	端侧小模型	把素材的声音转成文字，每个字带时间点
文本匹配 / 对齐	算法	把你的文稿与素材的语音对应起来（像做连线题）
视频剪切与编码	纯技术	把片段抠出、拼接、重新编码

对比项	产品一	AI 生成视频
画面哪来的	用户拍的	AI 造的
声音哪来的	用户录的	AI 造的
AI 在干什么	挑、排、对齐	无中生有
输出长度	只会变短	可以任意长
能否编出用户没拍的东西	绝对不能	这正是它的本事

对比项	手动版（用户熟悉的文本剪辑）	自动版（产品一）
文字稿从哪来	AI 把素材转成文字	用户自己写一份文稿
谁决定删什么	用户看着删	AI 自动决定
输出	用户手动剪好的片子	一份粗剪初稿
用户要做什么	通读全文、逐句判断	写个文稿，等结果

产品要素	实现细节	产品解读
主输入	文稿，最少 10 字、最多 8000 字	下限防空提交，上限是成本与质量平衡
双模式	「配音解说」（素材静音配解说）/「原声剪辑」（保留原声）	两种完全不同的成片形态，切换成本极低
生成方式	① 自己写文稿 ② 让 AI 写解说词	降低冷启动门槛
AI 写解说的输入	解说角度（画外 / 出镜）× 写作风格（5 种）× 口播时长（4 档）	60 种组合，但用户只需点 3 下
过程反馈	实时进度（主标题 + 副标题两行动态文案）	长任务必须有过程反馈，否则用户以为卡死
中止	取消按钮，10 秒兜底超时	防止取消失败导致界面卡死
结果处理	追加到原文案后面（不是覆盖）	不破坏用户已有内容，降低尝试成本
再生成提示	提示"不满意可重新生成，会拼接到最后"	一次性提示（本地存标记位，只弹一次），避免骚扰
辅助编辑	润色 / 扩写 / 缩写	把 AI 能力嵌进编辑动作，而非做成独立功能

产品要素	实现细节	产品解读
------	------	------

产品要素	实现细节	产品解读
发起	自然语言描述需求	门槛高（见 §3.2）
过程	阶段式推进 + 进度回报	长任务，需要可观测
产出	一个"粗剪方案"（镜头序列）	产出是草稿，不是成片
确认	明确不设确认门禁	追求"少打断"
素材约束	用户提供的素材必须真实存在	防幻觉
包装	独立的第二阶段（字幕 / 配乐 / 转场 / 贴纸）	见下

层级	能力	产品意图
免费额度	smart_cut_pint_detection（智能粗剪检测）	让用户用得上、用上瘾
VIP 付费	intelligent_crop（智能裁剪）、color_match（色彩匹配）、relight（智能补光）、intelligent_matting（智能抠像）、aigc_* 系列	变现
免费但带变现钩子	subtitle_recognition（字幕识别）、lyric_recognition（歌词识别）	识别免费，导出 / 去水印付费

埋点参数	说明他们在关心
action（5 种取值）	每一步的流失——用户是打开了不用，还是生成了不满意
cut_model（3 种）	哪条路径更受欢迎（配音解说 / 原声剪辑）
is_has_text / text_cnt	冷启动用户 vs 有稿用户的占比
voice_change_speed	音色偏好
apply_time / expected_time	实际耗时 vs 预期耗时——体感是否达标
status（成功 / 取消 / 失败）+ error_code	失败归因

维度	剪映的做法	常见竞品做法	剪映的差异化意图
输入	文稿（产品一）	直接"一键成片"	让用户保留创作主导权
精度	词级（改字即改片）	段落级	可精修，不是黑箱
定位	粗剪（初稿）	一键出成片	定位在"省时间"而非"替代人"
包装	独立阶段、非破坏	一并生成	模块化授权

风险	依据	影响
两个产品认知混乱	存在两个独立引擎、名称不同、入口不同	用户不知道什么时候用哪个
强依赖云端	实测云端不可达时产品二完全失效	断网或服务波动时体验崩塌
产品二输入门槛高	需要用户"描述画面"	小白用户用不起来
静默失败	不设确认门禁	AI 理解偏了用户不察觉
采纳率未度量	缺"结果被保留"的埋点	无法判断真实满意度

引擎	技术特征	对应的产品能力
本地转写编辑流水线	端侧语音识别 + 词级时间戳 + 删除区间标记	产品一（智能解说粗剪）
Agent 编排引擎	云端多模态大模型 + 工具调用	产品二（Hub 粗剪）
智能裁剪	端侧人脸检测 + 图像显著性	独立的画面重构能力（VIP）

在哪一层	单位	举例
AI 与工具之间	毫秒	1500
中间编辑指令	毫秒，且禁止写单位	只写 3000，写 3000ms 直接报错
最终草稿文件	微秒	3000000

技术选择	商业动因
分镜检测用 96×96 极低分辨率输入	必须能免费跑
深档分析限制在 30 秒内	深度分析成本高约 5 倍，只准用在关键处
每批召回推荐不超过 6 项、上限 12 项	每批都是一次付费调用
已分析结果必须缓存复用	避免重复付费
先产出"内容目录"再细看	用便宜的粗看筛掉大部分数据

结论	证据来源	强度
产品二名称为"剪映 Hub"	canvas_agent\app\index.html 标题栏	强
产品一不是生成模型，是筛选与重排	组件构成（语音识别 / 文本对齐 / 剪切编码）	强
底层为"改字即改片"，用删除标记而非真删	AttachmentAiClipperTranscript + Delete/Restore/Split/Trim/Unmark	强

结论	证据来源	强度
产品一依赖语音转文字，无口播素材即失效	机制推导（无线索可删）	强
产品一与"文本剪辑"共用底层	rough_edit / transcript_edit / ai_clipper 同源代码路径	强
同一底层衍生出手动版与自动版	文档名与工具链对照	中
产品二的实际交互界面	未见界面，仅有标题与后端路由	缺口
两个独立引擎与两套 skill 绑定	两套埋点体系、两套 skill 绑定	强
功能名"智能解说粗剪"、"开始粗剪"	应用内帮助链接标题、按钮文案	强
字数上下限 10 / 8000 且远端可调	前端默认值 + 远端配置键	强
取消有 10 秒兜底超时	交互代码	强
结果追加而非覆盖	交互代码	强
再生成提示只弹一次	本地存储标记位	强
5 个埋点及参数枚举	埋点定义与调用点	强
免费额度含粗剪检测	服务端权益名单	强
包装非破坏原则	产品规范原文	强
不设确认门禁	Agent 指令原文	强
三个时间单位分层	编辑指令规范 + 草稿文件字段	强
云端不可达时产品二失效	实测云服务连接失败	强
用户画像、竞品对比、优先级建议	无用户数据 / 市场数据支撑	推断
未经用户数据验证	无留存 / 转化 / 访谈数据	缺口

内容生产 Agent · 判分口径速查

评测集 v1 · 判分口径（一页速查）

用途：把 PRD 里的 C1-C6 六条硬性约束，转成**可执行、可复现、可解释**的判定规则。

本文是《评测集设计说明》第 2 章的速查版，面试时可直接展示这一页。

状态：**口径已定稿**；首轮跑分结果待原始素材迁移后补入。

>

v1.1 修正（据生产实际形态，由作者确认）：

- ① 真实结构是「**一条主线 + 6 个小论点**」逐步推进，不是 7 段骨架顺序展开 → C4 判定方式与权重已改；
- ② 生产失败模式中**不存在“缺段”** → C4 降为结构回归检查，**不再是一票否决项**；
- ③ 一票否决项收窄为 **C1（确认违规）+ C3**；
- ④ C3 校验基准确定为**只比对知识库检索原文**。

判定方式三分法

判定方式	用在哪	为什么
代码判	C1 / C3 / C4（机器部分）	稳定、可复现、可解释，能明确指出违反哪一条，能定向重试
LLM 判	C6 / C5 / C4（论点是否服务主线）	主观项放量初筛，但必须先人工标 10 条验证一致率 ≥80%
人工判	C2 / C5 / C6 复核	需专业理解；LLM 仅做初筛，人工抽检复核

C1-C6 判分表

编号	约束（原文）	来源	判定方式	可执行判据	违规等级
C1	类比必须来自知识库，禁止库外自创	实际	代码判（三档）	① 精确命中库内条目 → 通过；② 未精确命中但相似度 ≥ 0.25 → 疑似改写，待人工复核 （不否决）；③ 低于阈值 → 确认违规	仅"确认违规"一票否决
C2	类比与概念关键结构一一对应	设计还原	人工 + LLM 双评	类比是否覆盖概念的 3 个关键属性 ：全 3 / 部分 2 / 错配 0；两侧评分差 >1 分转人工	扣分项
C3	不引入未经验证的数据 / 事实	设计还原	代码判 + 人工复核	抽取全部数字·年份·专名 → 与知识库检索原文逐字比对 ；无出处即违规	 一票否决
C4	结构完整性（一条主线 + 6 个小论点）	设计还原	混合：机器抽取 + 人工/LLM 判定	路径 A（显式 7 段，历史格式）：段数/缺段/顺序；路径 B（真实格式）：机器抽取小论点数量， 各论点是否服务主线由人工/LLM 判	回归检查（非否决）：缺段 -10 / 变序 -5
C5	前后话题一致，不跑题	设计还原	人工 + LLM 双评	前/中/后各取 1 句算组间相似度； < 0.50 判跑题（阈值待校准）	扣分（严重转人工）
C6	口语通俗、术语必配解释、 信息密度足够	设计还原	人工 + LLM 双评	① 未解释术语数 ② 长句 (>40 字) 占比 ③ 信息密度。 "过度口语化"与"密度不足"需同时约束	扣分项

评分与分流

两层并用，**硬性规则优先**：

层	用途	规则
规则分（用于版本对	同一批用例上比较两版 Prompt	基础分 100 - 扣分项分值； C1 确认违规或 C3 违规 → 封

层	用途	规则
比)	/ 两份脚本	顶 <60
业务判定（用于上线决策）	决定这一稿去哪	命中一票否决 → 转人工；规则分 ≥85 → 通过；60-84 → 重试（≤2 次）；<60 或话题不一致 → 转人工

三条铁律

- 5. 能代码判的绝不用模型判 —— C1/C3/C4 走规则，保证可复现与可解释。
- 6. LLM 判必须先验证 —— 人工标 10 条，一致率 ≥80% 才放量；低于则回头修判据。
- 7. 主观项写 rubric，不写形容词 —— "类比要贴切"不可判；"覆盖概念 3 个关键属性"可判。

已实测的判分器边界（面试可信度来源）

边界	说明	影响
字符相似度识别不了重度改写	字符二元组 Jaccard 实测：轻度改写 0.74 / 核心片段 0.53 / 中度改写 0.29 / 重度改写 0.27；无关类比 0.00-0.02。取阈值 0.25 时，无关类比 4/4 正确否决、改写样本 4/5 正确放行	换词不换义的改写仍可能被误判，后续需引入语义匹配或"一条类比 + 多别名"登记
C3 只覆盖"带量数字"	不含数字的事实性断言（如"睡眠周期被打断"） 机器判不出	纯事实性错误需人工或 LLM 复核
出处"有没有" ≠ "不可信"	机器只判是否标注出处	输出中已注明出处本身仍需人工复核
C1 依赖类比库	未提供类比库时判"无法判定"，不会默认通过	避免无依据放行

尚未闭合的口径（需你确认）

- [] A1 ★ 类比库真实条目：以哪个清单为准？（未提供时 C1 无法判定）
- [] A4 小论点是否允许合并、是否固定 6 条？（影响 C4 路径 B 抽取规则）
- [] C5 相似度阈值 0.50、C6 长句阈值 40 字：均为初始参数，需按首轮数据校准。

评测集 v1 · 设计说明

评测集 v1：内容脚本合规与质量

项目：AI 内容生产 Agent 系统（通俗化讲解类口播脚本）

版本：v1.0（口径与用例就绪版）

作者：翟羽京

日期：2026-10-04

状态：判分口径与 30 条用例已定稿；首轮跑分结果待原始素材迁移后补入

阅读须知：本文的证据边界

沿用《剪映 AI 自动粗剪产品分析报告》同一套纪律，先划清本文里什么能被核实：

类别	内容
已确定（可复现）	判分口径（C1-C6 判定方式与判据）、30 条用例的输入与预期判定、执行流程、发布闸门、版本管理规则
待补入	首轮跑分的分维度通过率、各失败模式占比、判分器与人工的一致率
无证据（不编造）	涉及真实样本分布、真实留存/采纳数据的结论。凡未跑出的分数，本文一律留空并标注"待补入"

关于规模数据的可核实范围：PRD 中写的「知识库脚本原文 1,284 篇 / 类比技法 367 条」属**设计还原项**，

无法用本仓库核实。本作品集内放的是 **31 篇示范样本**（完整语料在另一台设备）（知识库语料/ 目录，12,445 句，

抽出 30 条类比候选）。为避免与 PRD 设计值冲突，本文所有涉及语料规模的表述一律以 31 篇为准。

>

为什么先交付“没有分数的评测集”：分数依赖尚未迁移的原始样本；口径与用例不依赖任何外部文件。

把能定的定死、把不能定的标清楚，比填一个无法核实的数字更有价值。

1. 评测目标与对象

1.1 评什么

评测对象：AI 内容生产 Agent 输出的「通俗化讲解类口播脚本初稿」。

不评什么（避免范围漂移）：

- 不评视频制作（PRD 已明确非目标）；
- 不评私域成交环节（系统边界止于公域内容与引流钩子）；
- 不评模型本身的能力（本评测评的是**产品链路 + Prompt + 知识库 + 评分规则的组合结果**）。

1.2 为什么必须建评测集

大模型输出是概率性的：同一 Prompt 换一批输入，效果可能天差地别。没有稳定可复现的测试集，就无法回答三个问题——**这次改动是变好了还是变差了、两个版本哪个更好、这稿能不能发。**

本项目的具体触发点：v1.0 存在“约 50% 初稿无法直接使用”的问题，但当时没有口径去回答“不合格”具体指违反了什么，问题只能靠人工通读发现。

1.3 评测要回答的三个决策问题

8. **能不能上线**：这版 Prompt / 知识库 / 评分规则能否放行？
9. **哪一版更好**：A 版与 B 版在同一批用例上的差异是否显著？
10. **哪里坏了**：不合格稿具体卡在哪个环节（召回不足 / Prompt 约束失效 / 评分漏判）？

2. 判分口径

完整速查表见 01-判分口径速查.md。本章说明设计理由。

2.1 为什么不采用“模型自由打分”

v1.0 曾考虑让模型对初稿打分。放弃理由（PRD 4.5 已记录）：

- 硬性约束（尤其 C1 类比必须来自库内）可判定、可解释、可复现，且能明确“违反哪一条”，从而定向重试；
- 模型自评不客观，同一稿多次评分会漂移，也难以稳定识别“类比是否库外”。

结论：客观项一律代码判；模型只用于主观项初筛，且必须先验证一致率。

2.2 判定方式与违规等级

v1.1 修正说明：最初设计时把“7 段骨架严格顺序”当作 C4 的判定对象，并把缺段列为一票否决项。据作者确认的生产事实，真实结构是「一条主线 + 6 个小论点」逐步推进（既避免过于专业化，又保证信息密度），且“缺段”并未出现在实际失败模式中。因此本表已按下述方式修正——这次修正本身也说明评测集必须建立在真实形态上，而不是设计文档的假设上。

编号	约束	判定方式	违规等级
C1	类比必须来自知识库	代码判（三档：命中 / 疑似改写待复核 / 确认违规）	🚫 仅“确认违规”一票否决
C2	类比与概念关键结构一一对应	人工 + LLM	扣分
C3	不引入未经验证的数据/事实	代码判 + 人工复核（基准：只比对知识库检索原文）	🚫 一票否决
C4	结构完整性（一条主线 + 6 个小论点）	机器抽取结构 + 人工/LLM 判“论点是否服务主线”	回归检查（非否决）：缺段 -10 / 变序 -5
C5	前后话题一致	LLM + 人工抽检	扣分（严重转人工）
C6	口语通俗、术语必配解释、信息密度足够	LLM 初筛 + 人工抽检	扣分

2.3 评分与分流

层	规则
规则分（版本对比用）	基础分 100 - 命中项分值；一票否决命中即封顶 <60

层	规则
业务判定（上线决策用）	一票否决 → 转人工；≥85 通过；60-84 重试（≤2 次）；<60 / 话题不一致 → 转人工

硬性规则优先：只要命中一票否决项，规则分再高也转人工——准确率不能越过安全与事实红线。

3. 用例规格（30 条）

3.1 来源与配比

来源	条数	构造方式	说明
真实 badcase 回流	12	按生产确认的失败模式重写为抽象用例（隐去真实文案，保留"选题类型 + 受众 + 违反约束 + 症状"）	主样本，最贴近真实问题
典型场景	8	从赛道（杭州 citywalk / 咖啡甜品 / 老杭州小吃）构造常见选题 × 三类受众	保证基本盘不崩
边界场景	6	从 PRD 第 9 章《异常与边界场景》1:1 派生	暴露薄弱环节
对抗场景	4	人工注入违规（库外类比 / 未验证数字 / 论点缺失 / 跨话题拼接）	守住底线

badcase 12 条的失败模式分布（由作者确认为真实生产问题）：

约束	失败模式	条数	等级
C5	前后话题不一致、跑题	3	扣分（严重转人工）· 最高频
C6	术语没解释、偏书面语	3	扣分项 · 最高频
C1	用了知识库外的自创类比	2	🚫 一票否决
C3	编造 / 记错数据与事实	2	🚫 一票否决
C2	类比和概念对不上	1	扣分项
C4	缺段 / 结构散乱	1	回归检查（非否决）

*重要修正：*据作者确认，生产失败模式中**不存在“缺段”**，且真实结构是「一条主线 + 6 个小论点」而非 7 段骨架。因此 C4 保留一条**回归用例**（防旧问题复发），但不再作为一票否决项。这也意味着 v1.0 到 v1.1 的判分权重从“结构优先”调整为“事实与来源优先、内容一致性次之”。

3.2 覆盖度原则

70% 典型 / 20% 边界 / 10% 对抗是通用起点；本项目为内容合规场景，对抗样本占比上调至约 13%（4/30），因 C1/C3 属一票否决项，误放代价高。

3.3 字段规范

每条用例固定 12 个字段（详见 02-评测集用例表.xlsx）：

用例编号 | 类型 | 来源/失败模式 | 输入选题 | 目标受众 | 知识库命中 | 预期判定 | 期望违规项 | 实际结果 | 实际违规项 | 是否一致 | 备注

关键要求：每条用例必须先写"预期判定"。没有预期的用例不是测试，只是素材。

3.4 用例清单（规格已定，实际结果待补）

- 30 条完整规格见 02-评测集用例表.xlsx。其中：
- 边界 6 条已按 PRD 第 9 章定稿（可直接跑）；
 - 对抗 4 条已按 C1/C3/C4/C5 定稿（可直接跑）；
 - badcase 12 条已按作者确认的失败模式定稿（输入选题为抽象描述，待补真实措辞）；
 - 典型 8 条选题骨架已定，待补真实选题措辞。

4. 执行流程

步骤	内容	产出
1. 校准判分器	人工标 10 条 → 与代码判 / LLM 判比对 → 一致率 ≥80% 才放量	判分器校准记录
2. 跑测	30 条逐条跑，记录实际结果与违规项	评测记录表
3. 归因	每条失败定位到「约束编号 → 根因 → 修复动作」	badcase 归因记录
4. 回归	修复后重跑该条，确认是否解决且未引入新问题	回归记录
5. 出报告	分维度通过率 + 结论 + 下一步	首轮评测报告

自动判分：run_eval.py 已实现 C1 / C3 / C4 的确定性判定，可批量跑并输出违规明细。C2 / C5 / C6 需人工或 LLM 判。

5. 发布闸门

条件	规则
达标线	合格率 ≥ 90%（初稿可直接使用比例）
一票否决项	出现 C1 确认违规 或 C3 违规 → 不得放行 ，无论总分多高
结构回归	C4 缺段 / 变序不触发否决，但需记录并回归（防旧问题复发）
版本放行	新版本在金标集上的分数 不得低于 当前版本
上线后追踪	上线后 1-2 周监控 badcase 增量与人工接管率

一票否决线必须在评测设计阶段写清楚，而不是出问题后补救。

一票否决项的取舍依据：只保留“误放代价不可逆”的两项——C1（库外自创类比，等于对外输出未经验证的知识）与 C3（编造事实）。C5/C6/C2 属质量问题，可通过重试与人工兜底收敛，因此设为扣分项而非否决项。

6. 版本管理与回流

- **像代码一样管理：**每次新增/修改用例记录变更原因与日期；
- **明确 owner：**本文档与用例表由产品负责人维护，避免集子随迭代过时；
- **开放贡献通道：**运营/编导发现的 badcase 可提交进入集子，不只靠算法侧维护；
- **警惕评测饱和：**某能力项长期 100% 时，降级为**回归集**（防旧问题复发），另开更难的新题；
- **回流节奏：**每周把线上发现的 badcase 按“约束编号 + 环节”归因后回流。

7. 当前状态与下一步

7.1 已完成

- [x] 判分口径定稿（C1-C6 判定方式、判据、违规等级）
- [x] 评分与分流规则（规则分 + 业务判定两层）

- [x] 30 条用例规格（边界 6 条、对抗 4 条定稿；典型 8 条骨架已定）
- [x] 执行流程、发布闸门、版本管理规则
- [x] 自动判分脚本（C1/C3/C4）

7.2 待补入（依赖原始素材迁移）

- [] 典型 8 条用例的真实选题措辞
- [] badcase 12 条的真实选题措辞（失败模式与受众已定稿）
- [] 首轮跑分：分维度通过率、失败模式占比
- [] 判分器与人工的一致率
- [] 阈值再校准（C5 相似度 0.50、C6 长句 40 字；C1 阈值 0.25 已实测标定）

7.3 下一步

11. 补入 30 条用例的真实选题措辞 → 用例表可跑；
12. 原始素材迁移后跑首轮 → 填第 7.2 各项；
13. 按首轮数据校准阈值 → 出 v1.2。

附录 A：与其他文档的对应关系

本文	对应文档	关系
C1-C6 判分口径	PRD 5.4《硬性约束》	本文把"约束"转成"判据"
评分与分流	PRD 5.6《动态评分校验》	本文的规则分与业务判定与之一致
用例来源	PRD 第 9 章《异常与边界场景》	边界 6 条 1:1 派生
验收方式	PRD 第 10 章《验收标准》	本文把"给定一当一则"转成可执行用例
阈值参数	PRD 附录 B6	本文附录列明待校准阈值
失败模式	PRD 第 12 章《风险与依赖》	风险项转化为可测用例
待同步	PRD 5.3《骨架生产（7 段式）》与附录 B3	⚠ PRD 现写"7 段骨架"且标注为"设计还原"，与真实结构「一条主线 + 6 个小论点」不一致，需回改 PRD

附录 B：术语

术语	说明
golden set（金标集）	用于发版闸门、“分数不退步才放行”的那一层用例
回归集	已饱和的旧题降级后的集合，用于防旧问题复发
badcase 回流	把线上发现的错误案例沉淀进评测集并归因修复的闭环
一票否决项	无论总分多高都不得放行的红线项
LLM-as-a-Judge	用模型对输出打分的方法，需先验证与人工的一致率

判分器校准记录（5 个实测问题）

判分器校准记录 v1（C1 / C3 / C4）

用途：记录自动判分逻辑在开发中实际暴露的错误，以及修正方式。

为什么值得留档：这件事同时证明两件事——① 判分器不是"写完就算"，是需要被校准的；② 我自己会发现并修掉假阳性/假阴性，而不是让它污染分数。

数据说明：本记录中的样本均为构造样本（用于验证规则），不代表真实生产数据。

背景：为什么客观项必须代码判、且代码判也必须被校准

C1（类比必须来自知识库）、C3（不引入未经验证的数据/事实）、C4（7 段骨架完整）被定义为一票否决项。一票否决项的特征是**误判代价不对称**：

- **假阴性**（真违规判成通过）→ 违规内容流出，是事故；
- **假阳性**（合规内容判成违规）→ 正常稿件被拦，增加人工成本，久了团队会绕过评测。

因此判分器必须双向验证。以下两个问题都是在这个验证过程中被发现的。

问题 1：出处识别把「段落标题」当成了引用（假阴性）

现象：脚本

【原理展开】

南宋时期杭州人口超过一百万，是全国最大的城市。

一百万 属无出处数字，应判 C3 违规；实测却判为"通过"。

根因：出处正则最初把任意 **【...】** 都视为引用标记，于是 7 段骨架的段落标题 **【原理展开】** 被识别为"有出处"。

危险性：这是系统性的假阴性——只要脚本按规范写段落标题，所有无出处数字都会被自动豁免，等于 C3 这条一票否决项在生产环境里完全失效。

修正：只承认真正的引用写法，覆盖三类真实写法：

14. 段内注释：（来源：省文旅厅）、（出处：内部活动数据）

15. 前置引用：据《睡眠医学》2023 年的研究、引自 XX 报告

16. 脚注编号：[1]

回归验证：

输入	期望	实测
据《睡眠医学》2023 年的研究...	豁免	✅ 豁免
南宋时期杭州人口超过一百万。	违规	✅ 违规
浙江省 2024 年接待游客超过 3 亿人次（来源：省文旅厅）。	豁免	✅ 豁免
南宋时期杭州人口超过一百万[1]。	豁免	✅ 豁免
【原理展开】南宋时期杭州人口超过一百万。	违规	✅ 违规

问题 2：数字识别把口语量词当成数据（假阳性）

现象：修正问题 1 后，"合规样本"里

就像手机后台开了一堆 App...

被 C3 判为违规，抓出的"数字"是 一个。

根因：数字正则匹配"中文数字 + 量词"时过于宽松，无法区分事实性数据（"一百万""3 亿""五十条"）与虚指量词（"一个""两个""一堆"）。

危险性：假阳性会淹没真问题——如果每篇稿子都因"一个"报违规，C3 的告警就没人看了。

修正：

- 阿拉伯数字一律视为需核验数据（可带单位）；
- 中文数字仅在数量级表达（万/亿/千/百/倍）或带单位且非虚指量词时才计入；
- 明确排除"一个""两个"这类口语量词。

回归验证：

输入	期望	实测
南宋时期杭州人口超过一百万	命中	命中
提升了 300% / 连续打卡 66 天 / 每天走 3 公里 / 阅读了 1284 篇 / 涨粉一万 / 复盘了五十条视频	命中	全部命中
开了一堆 App，电量看着满	不命中	不命中
第一个做法是先固定起床时间	不命中	不命中
这两个任务互斥	不命中	不命中
拿一个号排队	不命中	不命中

已知边界（当前版本不做，写清以免被当成 bug）

边界	说明	影响
无数字的事实断言判不出	"睡眠周期被打断"这类 不含数字 的断言，C3 无法判	C3 只能覆盖"带量数字"，纯事实性错误需人工或 LLM 复核
口播正文提取是启发式的	提取依赖"制作元信息行"的特征词匹配（时长/画幅/镜头/标签...）。 如果输入是策划稿 （含选店清单、路线、发布时间），仍有漏网数字进入判定	已加低置信度护栏（见问题 5）：违规比例 >60% 时判"结果不可用"，不产生违规结论、不触发否决
出处"有没有"不等于"不可信"	机器只判是否标注出处，不判出处本身是否真实	输出中已注明"出处本身仍需人工复核"
C1 依赖类比库	未提供类比库时 C1 判为"无法判定"，不会默认通过	避免在无依据时给出通过结论
C2 / C5 / C6 未实现	需人工 + LLM 双评，且 LLM 判需先验证与人工一致率 ≥80%	见《判分口径速查》判定方式三分法
字符相似度识别不了重度改写	见问题 3；中度改写相似度 0.293 贴近阈值 0.25	需后续引入语义匹配或"一条类比 + 多别名"登记

修正后的自测结果

SELF TEST 1: 合规样本 → rule_score 100 / veto false / route 通过

SELF TEST 2: 违规样本 → rule_score 59 / veto true / route 转人工（一票否决）

C1 不通过 1 项（库外类比）

C3 不通过 1 项（无出处数字：一百万）

C4 不通过 1 项（缺段，仅扣分不否决）

SELF TEST 3: C1 三档判定 → 通过 / 待人工复核 / 不通过 三档均按预期

SELF TEST 4: C4 双路径 → 路径 A（显式 7 段）/ 路径 B（主线+论点）均正确识别

SELF TEST RESULT: PASS

复现命令：

```
python run_eval.py --self-test
```

问题 3：严格子串匹配把「合规改写」判成违规（假阳性）

现象：评测集设计前假设"类比句必须能在库内精确命中"。实测发现脚本会使用**知识库类比的改写版**（例如库内是"就像手机后台开了一堆 App，电量看着满，用起来掉得飞快"，脚本写成"就好比手机后台开了一堆 App，电量看着挺满，用起来掉得飞快"）。按严格匹配会判违规，触发一票否决——**把合规内容拦下**。

根因：把"类比必须来自知识库"（语义要求）实现成了"字符串必须一致"（字面要求）。

修正：C1 改为**三档判定**，并计算字符二元组 Jaccard 相似度：

档位	条件	处理	是否触发否决
命中	含库内条目的关键片段	通过	否
疑似改写	相似度 ≥ 0.25	送人工复核，不计入否决	否
确认违规	相似度低于阈值	判违规	 是

阈值实测标定（不是拍脑袋定的）：

样本类型	相似度	目标判定
原文	1.000	命中
轻度改写	0.742	疑似改写
保留核心片段	0.533	疑似改写

样本类型	相似度	目标判定
中度改写（换词不换义）	0.293	⚠ 已知边界
重度改写	0.268	疑似改写
无关类比 A/B/C/D	0.000 / 0.000 / 0.022 / 0.000	确认违规

阈值扫描结果：取 0.25 时，无关类比 4/4 正确否决、改写样本 4/5 正确放行；阈值抬高到 0.30 后改写样本只能放行 3/5，只会更差。故定为 0.25。

已知边界（如实记录，不掩盖）：字符相似度识别不了"换词不换义"的重度改写——中度改写 0.293 已贴近阈值，一旦低于 0.25 就会被误判为确认违规。正确解法是后续引入**语义匹配（向量检索）**，或把库内条目登记为"一条类比 + 多个别名"。

问题 4：C4 的判定对象搞错了（设计假设 vs 生产事实）

现象：最初把 C4 定义为"完整保留 7 段骨架，不缺段不调序"，并把缺段设为一票否决项。

根因：这个定义来自 PRD 5.3 的"7 段式骨架"，而该处在 PRD 附录 B3 中本就标注为「**设计还原**」（即作品集呈现所需、非原始实现）。作者确认的真实形态是：**一条主线 + 6 个小论点逐步推进**——目的是既避免过于专业化，又保证大多数人看得懂且信息密度足够。同时作者确认的生产失败模式里**根本没有"缺段"**。

危险性：评测集会去测一个生产中不存在的问题，同时把真正高频的失败（跑题 3 条、术语与书面语 3 条）权重压低。

修正：

17. C4 改为**双路径**：路径 A（显式 7 段，历史格式）走段数/顺序检查；路径 B（真实格式）机器只抽取小论点数量，"各论点是否服务主线"交人工/LLM 判定；

18. C4 从一票否决项降为**结构回归检查**（缺段 -10 / 变序 -5）；

19. 一票否决项收窄为 **C1（确认违规）+ C3**；

20. badcase 用例分布按真实失败模式重排：**C5 跑题 3 条、C6 术语与书面语 3 条、C1 两条、C3 两条、C2 一条，C4 仅保留 1 条回归用例。**

连带发现：PRD 5.3 的"7 段式骨架"与真实结构不一致，**PRD 需要回改**（已在《评测集设计说明》附录 A 标注待同步）。

问题 5：C3 的判定范围错了——把制作元信息当成事实性数据

现象：拿作品集里真实存在的脚本 02-首条脚本-杭州运河 Citywalk.md 试跑，C3 判出 **46 条违规 / 47 个数字**。

抓出来的"无出处事实"包括：**40 秒**（视频时长）、**9:16**（画幅）、**地铁 3 号线**（交通方式）、**步行约 15 分钟**（路线）、**1 甜品**（选店配置）。

根因：C3 的判据是"数字必须有出处"，但**没有限定判定范围**——脚本文件里除了口播正文，还有大量面向制作的元信息（时长、画幅、镜头、路线、标签、发布时间）。这些不是讲给受众听的内容，不需要出处。

危险性：如果直接用，**任何一份正常脚本都会被判一票否决**。实测中它把 summary 直接推成"转人工（一票否决）"。

修正：

21. 新增 `extract_narration()`：按制作元信息特征词（时长/画幅/竖版/镜头/标签/POI...）跳过非正文行，C3 默认只在**口播正文**上判定；输出中记录跳过了多少行；
22. 增加 `--no-narration-filter` 开关，用于在全文本上复核；
23. **增加低置信度护栏**：若违规比例超过 60%，判"**结果不可用（低置信度）**"，不产生违规结论、不触发否决，并提示"输入很可能不是纯口播稿"。

修正后实测：

C3：结果不可用（低置信度）

warning：违规比例 34/35 超过 60%：输入很可能不是纯口播稿

（含选店清单/路线/时长等非受众内容），请提供纯口播正文后再判

summary：rule_score 100 / veto_triggered false / route 通过

如实说明这个结果的局限：02-首条脚本-杭州运河 Citywalk.md 本质是**策划稿**（含选店清单与路线），不是纯口播稿，所以正文提取只能算启发式。护栏的作用不是"判对了"，而是**拦住一个会误导决策的错误分数**——真正的修法是评测用例统一使用纯口播正文格式，这已列入待办。

问题 6：知识库规模——设计值与实测值不一致

现象：PRD 中记录的知识库规模为「脚本原文 1,284 篇 / 类比技法 367 条」，而本作品集内放的是 **知识库语料/** 下的 **31 篇示范样本**（完整语料在另一台设备，未迁移）（12,445 句，抽出 30 条类比候选）。两个数字相差约 40 倍。

根因：PRD 的该处数据与「7 段式骨架」一样属于**设计还原项**（PRD 附录 B 已标注），不是对实际知识库的盘点；而评测集的 C1 判定必须建立在**真实**的类比库上。

为什么必须写出来：C1 是“类比必须来自知识库”这条一票否决项。**如果类比库不完整，C1 会把大量合法类比误判为“库外自创”**——库越小，假阳性越高。这是比判分器 bug 更根本的风险。

处理方式：

- 24. 本文与设计说明中所有语料规模表述改以 **31 篇实测值**为准，并显式标注 PRD 设计值不可核实；
- 25. `kb_analogies.txt` 定性为**初稿**（30 条自动抽取结果），需人工筛选后才能作为 C1 基线；
- 26. 在比对结论中增加前提说明：**评测结果的有效性以“类比库完整度”为前提**，类比库未补全前，C1 的“确认违规”结论需人工复核。

问题 7：库内类比被误判——整句与半句的匹配错位

现象：把筛选后的正式类比库接上判分器联调时发现，**库里真实存在的类比被判为「待人工复核」甚至「确认违规」**。

例：库内条目是整句「就像外卖系统给外卖员设置计件以及送达时间的 deadline 一样，我们都知道 deadline 可以批量生产马路神仙」，

而脚本里类比常写成半句「就像外卖系统给外卖员设置计件以及送达时间的 deadline 一样」——精确包含匹配失败。

根因：判分器用「子串包含」判定命中，但**库存整句、脚本抽半句**时两者互不包含。这会造成系统性假阳性：库里的类比越多，

脚本写法越灵活，误判越多——**等于把一票否决项变成了随机否决**。

修正（三档判定顺序）：

- 27. 精确包含 → 通过;
- 28. 最长公共子串 (LCS) ≥ 库条目的 60% → 视为同一类比, 通过;
- 29. 否则计算字符二元组相似度, ≥0.25 送人工复核, 低于阈值判确认违规。

实测 LCS 比率（标定依据）：

样本	LCS 比率	判定
库内原句	1.00	通过
轻度改写	0.97	通过
中度改写（换词不换义）	0.70	通过
保留核心片段	0.60	待人工复核
无关类比	≈0.00	确认违规

连带收益：此前记录的「已知边界——中度改写 0.293 会被误判为违规」已被修复：LCS 判定把它正确识别为同一类比。

同时修掉的次要问题：类比常横跨逗号，原按句号切分会切出与库对不上的片段；已改为「整句 + 逗号级分句」两级提取，

并对结论去重（整句已命中时，其分句不再重复报告）。

结论与下一步

30. 一票否决项的判分器**必须双向验证**（既测该拦的、也测不该拦的）。本次五个问题分别是：

- 问题 1 假阴性（最危险：会让 C3 完全失效）；
- 问题 2 假阳性（会让告警被淹没）；
- 问题 3 字面/语义混淆（会把合规改写拦下）；
- 问题 4 判定对象错误（测了生产中不存在的问题）；
- 问题 5 判定范围错误（会把任何正常脚本判死）。

五类都不是"写代码时的粗心"，而是**判据设计问题**——这正是评测集必须先做校准、不能直接跑分的原因。

另有问题 6 属**数据基础问题**（判分器正确性会被类比库完整度决定），问题 7 属**匹配算法问题**（库存整句、脚本抽半句导致系统性命失败）。

31. **判定对象与范围必须来自生产事实，不能来自设计文档的假设**——问题 4、5 是同一类错误的两面：一个是“测错了东西”，一个是“在错误的文本上测”；

32. 阈值类参数（C6 长句 40 字、C5 相似度 0.50）仍是初始值；C1 阈值 0.25 已实测标定，但受“字符相似度”方法限制；

33. 下一步：

- 评测用例统一改用**纯口播正文**格式，避免再触发低置信度护栏；
- 把双向验证思路扩到全部 30 条用例，统计判分器与人工判断的一致率，**低于 80% 的维度不得放量**。

成本与延迟预算（文字链路）

成本与延迟预算（选题与脚本生成 · 文字链路）

归属：本文核算 AI 内容生产 Agent 系统（选题 + 脚本生成）的成本，属文字链路。

与其它文档的边界：画面生成成本见 02-数字人内容流水线/成本核算-生视频.md，两者不重叠。

v1.2 修正说明： 本文件早期版本混入了视频生成成本（RunningHub / MiniMax H3），

属于归类错误——那部分不属于本系统。现已拆分，视频成本移出。

一、实测调用结构

依据 02-数字人内容流水线/04-脚本与台账/N8N-WF1-选题与脚本.json（16 节点）逐节点核对，只有 2 个节点调用 LLM：

用 LLM：

环节	每轮次数	输入 tokens	输出 tokens	备注
选题生成	1	140	~400	system 54 字 + user 157 字；输出 5 个选题 JSON
脚本生成	5	183	~600	每个选题 1 次；输入只注入选题四字段
合计	6 次	1,055	3,400	4,455 tokens/轮（5 条脚本）

三处与设计文档不一致（实测发现）

PRD 里的假设	workflow 实测	影响
有"动态评分 → 自动重试 ≤2 次"	WF1 没有评分与重试节点	成本模型不含重试倍数
走 RAG 召回知识库原文	脚本节点只注入选题四字段，知识库原文未进 prompt	无检索 token 成本
脚本可能按段多次调用	每选题恰好 1 次	调用次数可确定

二、单篇成本（按不同模型单价）

模型	输入（元/百万）	输出（元/百万）	单轮（5 篇）	单篇
Doubao 1.5 Lite 32k	0.3	0.6	¥0.0024	¥0.0005
Doubao 1.5 Pro 32k	0.8	2.0	¥0.0076	¥0.0015
Doubao Seed 1.6 Flash	0.15	1.5	¥0.0053	¥0.0011
DeepSeek flash（低谷）	1.07	4.26	¥0.0156	¥0.0031

结论：文字链路的成本可以忽略——单篇约 ¥0.0015，日更 10 篇月成本约 ¥0.5。

三、量级外推

日产量	月 LLM 成本
1 篇/天	≈ ¥0.05
3 篇/天	≈ ¥0.14
10 篇/天	≈ ¥0.46

四、延迟结构（待实测）

延迟结构与成本结构相反：LLM 只占 16 个节点中的少数，人工审批（Wait）才是耗时主体。

环节	类型	P50	P95
选题生成	LLM	☐	☐
脚本生成 ×5	LLM（可并行）	☐	☐
飞书写入/回填	HTTP	☐	☐
人工审批（Wait）	人工	☐	☐
端到端（含人工）	—	☐	☐

判断：所谓"周期压缩 70%"，压缩的是人工等待与返工，不是机器耗时。

五、诚实边界

项	说明
实测	调用次数、prompt 字数（直接解析工作流 JSON）
牌价	各模型单价来自公开价格页， 以你实际账单为准
未核实	端到端耗时、人工审批占比、重试是否真的未实现

不编造声明：本表的单位成本为“实测用量 × 公开单价”的推算，非实际账单数字。

数字人内容流水线 · n8n 搭建说明

N8N workflow 搭建说明（AI 数字人探店 Citywalk）

设计原则：用飞书多维表格「状态」做队列，3 个小 workflow 各管一段，独立失败、可单独开关。

- WF1 选题与脚本（已给 JSON）
- WF2 配音与数字人画面（待补）
- WF3 剪辑发布与复盘（待补）

一、N8N 环境（自托管，免费）

34. 本地启动（需 Docker）：

```
docker run -it --rm -p 5678:5678 -v n8n_data:/home/node/.n8n n8nio/n8n
```

35. 浏览器开 <http://localhost:5678> 注册本地账号。

36. 人工门的 Wait Webhook 要公网可达（手机点链接才能触发），用内网穿透：

- Cloudflare: `cloudflared tunnel --url http://localhost:5678`（给一个临时 https 域名）
- 或 ngrok: `ngrok http 5678`
- 把这个域名填到变量 `N8N_BASE`（不带结尾斜杠）。

37. workflow 测试可用「手动触发」；正式用要把右上角 Active 打开（Wait 节点在 Active 下走 production webhook 地址）。

二、配置 N8N Variables（Settings → Variables）

变量名	填什么
<code>FEISHU_APP_ID</code>	飞书自建应用 App ID
<code>FEISHU_APP_SECRET</code>	飞书自建应用 App Secret
<code>FEISHU_BITABLE_APP</code>	多维表格 app_token（表格 URL 中 <code>/base/</code> 后那段）

变量名	填什么
FEISHU_BITABLE_TABLE	数据表 table_id (表格 URL 中 table= 后那段)
FEISHU_BOT_WEBHOOK	飞书群自定义机器人的完整 Webhook 地址
LLM_API_KEY	豆包方舟 / DeepSeek 的 API Key
LLM_MODEL	模型/接入点 ID (豆包方舟为 endpoint id; DeepSeek 填 deepseek-chat)
N8N_BASE	内网穿透后的公网域名, 如 https://xxx.trycloudflare.com

JSON 里 LLM 默认用豆包方舟 `https://ark.cn-beijing.volces.com/api/v3/chat/completions`
(OpenAI 兼容)。用 DeepSeek 就把 URL 换成 `https://api.deepseek.com/chat/completions`。

三、飞书自建应用（多维表格读写）

38. 打开 `https://open.feishu.cn` → 「开发者后台」→ 「创建企业自建应用」, 拿到 App ID / Secret。
39. 「权限管理」开通:
- `bitable:app` (查看、评论、编辑多维表格)
 - (发消息若不走群机器人, 可加 `im:message`)
40. 「版本管理与发布」→ 创建版本并发布 (个人/企业内通过即可)。
41. 关键: 打开你的多维表格 → 右上角「...」→ 「添加文档应用 / 协作者」→ 把这个自建应用加进去并给可编辑权限, 否则接口报无权限。

群自定义机器人（发审批通知）

42. 建一个工作群 (可只有自己) → 群设置 → 群机器人 → 添加「自定义机器人」→ 复制 Webhook, 填到 `FEISHU_BOT_WEBHOOK`。

四、多维表格「探店内容台账」字段

字段名	类型	说明
路线/主题	文本	
起终点/交通	文本	

字段名	类型	说明
店铺方向	文本	
钩子	文本	
状态	单选	线索/待出稿/待确认/待配音/待画面/待剪辑/待审发/已发布
口播稿	多行文本	
分镜 B-roll	文本	
音频 URL	URL	
数字人视频 URL	URL	
成片 URL	URL	
发布链接	URL	
发布时间	日期	
播放/完播/点赞/评论/收藏	数字	各建一个
备注	文本	

状态流转：待出稿 → (WF1) → 待确认 → (人工①) → 待配音 → (WF2) → 待画面 → 待剪辑 → (WF3) → 待审发 → 已发布。

五、导入并测试 WF1

43. N8N → Workflows → Import from File，选 **N8N-WF1-选题与脚本.json**。

44. 确认第二节变量都已配置。

45. 点「手动触发」：

- 观察：拿到飞书 token → LLM 给 5 个选题 → 写入台账 → 拉待出稿 → 逐条出脚本 → 回填 → 群里收到审批卡片。

46. 手机点卡片「 通过」（此时需内网穿透在线）→ Wait 节点继续 → 状态变「待配音」；点「打回」则回「待出稿」。

47. 常见报错：

- 401/无权限 → 应用没加进多维表格协作者，或权限没发布；

- Wait 点了没反应 → N8N_BASE 域名不对/穿透没开;
- LLM 解析失败 → 模型没严格输出 JSON, 可在 LLM 节点加 `response_format: {"type": "json_object"}`。

六、WF2 / WF3 节点规划 (确认 WF1 跑通后我补 JSON)

WF2 配音与数字人 (定时, 扫描「待配音」)

48. 获取 token → search 待配音

49. Fish Audio TTS: POST <https://fishaudio.org/api/open/v3/speech/tts> (header Bearer, body {text, voiceId}) → 回填音频、状态待画面

50. 分支:

- **付费全自动**: 火山 OmniHuman CVSubmitTask (肖像图 URL + 音频 URL) → 轮询 CVGetTask → 回填视频、状态待剪辑

(火山 visual 接口要 AK/SK HMAC 签名, 我会在 WF2 里给一个签名 Code 节点)

- **免费人工门②**: 飞书通知「即梦上传肖像+音频」→ Wait → 人工回填视频 → 待剪辑

WF3 剪辑发布与复盘

51. 扫描待剪辑 → 飞书通知实拍+剪辑 (附镜头清单) → Wait 回填成片 → 待审发

52. 审核卡片 (AI 标注/广告标注/违禁词/版权) → Wait

53. 发布通知 (标题/话题/POI) → Wait 回填发布链接 → 已发布

54. 每周定时: 回收数据 → LLM 复盘爆款钩子 → 更新脚本模板与选题权重 (写回表格/文档)

七、风控与合规 (自动化边界)

- 发布环节**不建议硬全自动** (批量秒发触发风控), 保持人工错峰发布;
- 矩阵去重靠脚本钩子和素材组合, 不用镜像/变速;
- AI 数字人内容按平台标注「AI 生成」, 商单标注「广告」;
- 没到店不编体验, 实拍是流水线里不可删除的人工环节。

生视频成本核算

生视频成本核算（数字人流水线）

归属：本文核算画面上成的相关成本，属于数字人内容流水线。

与其它文档的边界：03-AI 内容生产 Agent 系统/评测集/04-成本与延迟预算.md 只核算选题与脚本生成（文字链路）的成本，两者不重叠。

一、已确认的实测单价

项	值	来源
MiniMax H3（RunningHub）	720P + 4 秒 = ¥1，按次扣款	作者实测

说明：H3 在 RunningHub 上有两种形态——ComfyUI 画布工作流（按算力时长）与标准模型 API / AI 应用（按次）。实际使用的是表单形态 → 按次计费。

二、成本公式

单条成片成本 = 片段数 × ¥1 × 重生成倍数

片段数 = 成片时长 ÷ 4（向上取整）

重生成倍数 = 1 ÷ 首次通过率

三、按实测成片时长（49 秒）计算

素材	时长
开场钩子-前 4 秒-驱动口型用.wav	4.0 秒
首条口播-运河 citywalk.wav	45.0 秒
合计	约 49 秒 → 13 个 4 秒片段

首次通过率	单条成片	月更 10 条
100%	¥13	¥130
50%	¥26	¥260
30%	¥43	¥430
20%	¥65	¥650
10%	¥130	¥1,300

通过率未实测，上表是按公式推算的区间，不是已发生的支出。

四、三个可执行动作

- 55. 补测首次通过率（唯一的键变量）：同一角色、同一套提示词连续跑 10 次，数出可用几条；
- 56. 优先用图生视频：拿已定稿的角色图当首帧（资产里有 角色-正面肖像-对口型用.png、角色-全身正面-参考用.png），一致性由输入图锁定；
- 57. 4 秒用满：4 秒是计价下限，生成 2 秒和 4 秒同价。

五、与另一条链路的关系

链路	成本主体	归属文档
选题与脚本生成	LLM token（极低，单篇约 ¥0.0015）	03-AI 内容生产 Agent 系统/评测集/04-成本与延迟预算.md
画面上成	生视频按次（¥1/4 秒）	本文

结论：两条链路的成本量级差三个数量级——文字侧几乎可忽略，成本压力全在画面上成。

六、诚实边界

- 单次单价（¥1/4 秒）= 作者实测；
- 成片时长 49 秒 = 由音频文件实测（wav 头部采样率不可靠，已用文件大小反推校正）；
- 首次通过率、失败请求是否扣费 = 未核实；
- 表中成本为公式推算，非实际支出。

习惯养成 App

06 · 习惯养成 App（脑科学 90 天计划）

这份材料证明：从内容设计到工程交付的完整闭环能力。

交付物

文件	说明
脑科学习惯养成 90 天计划.html	完整原型（含今日打卡、脑科学原理、四大习惯、每日例程、奖励系统、90 天路线、急救手册、周复盘）
_shots/习惯.apk	Android 安装包（约 4.7 MB）
_shots/*.jpg	桌面端与移动端真机截图

内容侧的判断（不只是个打卡工具）

原型把行为设计理论落成了可操作机制：

- 习惯回路 CUE → ROUTINE → REWARD：用"刷牙 / 坐下 / 换衣"当线索，微习惯约 8 分钟，庆祝 + 每日大奖做奖励；
- 引用研究支撑"66 天"而非"21 天"：伦敦大学学院 Lally 等（2010）追踪 96 人，行为自动化中位数 66 天，个体跨度 18-254 天，漏一天不影响最终结果——并明确标注曲线为按研究拟合的示意图；
- 行为控制权迁移：从"目标导向系统（前额叶主导、每次重新决定、费力）"迁到"习惯系统（基底核、线索触发即执行、自动省力）"。

已知局限（主动说明）

- 工程脚手架来自应用生成器模板（其 README 明确写明 `src/index.tsx`、`components/ui/*` 禁止修改），业务逻辑、页面与数据模型为该作品实现部分；

- **这是个人自用工具**，不以用户量为目标——所以「真实用户数据」不是它的缺失项，它的价值在于「从行为科学到可交付产品」的完整实现。

面试若被问“这是不是 AI 生成的”：直接承认脚手架来源，把重点放在你做了什么判断、否决了什么方案上——这比被追问出来强得多。

我的贡献边界与 AI 使用说明

我的贡献边界与 AI 使用说明

为什么要写这一页：这份作品集的完整度较高，面试官很自然会问"这些是不是你做的"。

与其被追问，不如**先把自己的边界说清楚**——这本身就是产品经理该有的诚实与自省。

本文只讲事实，不讲感受。

一、总览：我做了哪些判断，工具做了哪些执行

类别	我做的	工具/AI 做的
问题定义	判断什么问题值得做、边界在哪	—
方案取舍	选哪条路、放弃哪条、为什么	—
标准制定	什么叫"好"、什么叫"不合格"、什么算"通过"	—
判据设计	哪些能机器判、哪些必须人判、阈值定多少	—
代码/文档产出	逻辑与结构	排版、润色、代码初稿
素材生成	提示词策略、验收标准	图像/视频的实际生成
工程脚手架	业务逻辑、页面、数据模型	项目模板与 UI 组件库

二、逐作品说明

1. 剪映 AI 自动粗剪 · 产品分析报告

四栏	内容
我独立完成	证据边界的划分（什么是硬证据、什么是推断）；三个核心发现（商业约束倒推技术架构、追加而非覆盖的信任设计、采纳率才是北极星）；证据索引表与强度标注；P0-P4 建议的优先级排序
用 AI/工具后修改	文档排版、语句润色
我否决/放弃的	最初想写成"功能罗列 + 竞品对比"，后来否决——因为我没有用户数据和市场数据，写

四栏	内容
	出来的对比全是猜测。改成"从产品实现反推决策"后，每条结论才都有出处
我无法验证的	留存率、转化率、真实用户访谈、竞品内部数据（文中已逐处标注为推断）

2. AI 内容生产 Agent 系统 · PRD

四栏	内容
我独立完成	问题定义与非目标；"关键产品决策与取舍"表（v1 为什么先做单链路、评分为什么不让模型自由打分）；C1-C6 硬性约束；阈值 85/60 与重试 ≤ 2 的设定；附录 B 的设计还原项标注
用 AI/工具后修改	文档结构规范化、排版
我否决/放弃的	① 让模型自由打分——否决，理由是模型自评会漂移、不可复现；② v1 直接上多智能体——否决，理由是生产未稳时前，前后置智能体没有可靠输入，只会放大不确定性与成本
我无法验证的	知识库规模（完整语料在另一设备；作品集内放的 31 篇是示范样本，用于演示抽取方法）；原始 SOP 与界面未迁移；样本统计口径待补

3. 评测集 v1 与判分器 (**这部分是我最想让你看的地方**)

四栏	内容
我独立完成	判分口径（C1-C6 判定方式与判据）；评分与分流规则；30 条用例的构造；发布闸门与一票否决项；判分器脚本的逻辑设计；5 个判分 bug 的发现与修正
用 AI/工具后修改	代码复现与调试、文档排版
我否决/放弃的	① 让模型判所有项——否决，改成"能代码判的绝不用模型判"；② C4 把"7 段骨架"当判定对象——已否决并修正，因为真实结构是"一条主线 + 6 个小论点"，且生产中根本没有"缺段"问题
我无法验证的	首轮跑分结果（原始样本在另一设备）；判分器与人工的一致率（未跑）；类比库完整度——作品集内的 31 篇语料是示范样本，完整知识库在另一设备

判分器上实测发现并修正的 5 个问题（这是"我真的在动手"最硬的证据）：

#	问题	类型
1	引用识别把段落标题【原理展开】当成出处	假阴性（会让 C3 一票否决项完全失效）

#	问题	类型
2	数字识别把"开了一个 App"当成数据	假阳性
3	严格字符串匹配把类比的 合规改写 判成违规	字面/语义混淆
4	C4 拿"7 段骨架"当判定对象	判定对象错误
5	C3 把"时长 40 秒""竖版 9:16"当成事实性数据	判定范围错误

4. 数字人内容流水线 · n8n workflow

四栏	内容
我独立完成	用飞书多维表格状态字段做队列的设计；三 workflow 各管一段、独立失败的拆分；LLM 输出结构化解析 + try/catch 兜底；人工审批走 Wait 节点；发布环节刻意不做全自动（平台风控约束）
用 AI/工具后修改	节点 JSON 的写法、文档整理
我否决/放弃的	发布环节全自动——否决，批量秒发会触发平台风控，改为人工错峰发布
我无法验证的	WF2 / WF3 目前是节点规划，未产出 JSON（文档中已标注"待补"）；单篇的真实耗时与成本

5. AIGC 角色资产体系

四栏	内容
我独立完成	"固定锚点段 + 场景段 + 画质段"的提示词结构；角色一致性的锁定策略；AI 开场与实拍正片的"接缝处理"四原则（转场遮挡、色调统一、环境音贯穿、首尾呼应）；发布前检查清单
用 AI/工具后修改	图像与视频由平台生成（即梦 / RunningHub + MiniMax H3 等），我做提示词与筛选
我否决/放弃的	用大动作镜头——否决，SOP 里写明"动作越小越像真人，大转头、大挥手最容易暴露 AI"
我无法验证的	首次通过率未实测；各平台单价未逐一核对

6. 习惯养成 App

四栏	内容
我独立完成	内容设计（习惯回路 CUE→ROUTINE→REWARD 的落地、四习惯体系、90 天路线、

四栏	内容
	急救手册、奖励机制)；数据模型与统计逻辑；页面实现
用 AI/工具后修改	工程脚手架来自应用生成器模板（模板 README 明确写明 <code>src/index.tsx</code> 、 <code>components/ui/*</code> 禁止修改)；排版与样式调整借助工具
我否决/放弃的	纯打卡工具路线——否决，改为用行为科学解释"为什么是 66 天而不是 21 天"（引用 Lally 等 2010 研究，96 人追踪，自动化中位数 66 天，个体跨度 18-254 天）
我无法验证的	无（这是个人自用工具，不以用户量为目标，因此「真实用户数据」不是它的缺失项）

三、我是怎么用 AI 的（方法论，不是辩解）

用 AI 做什么	不用 AI 做什么
生成文档初稿与排版	定义问题与非目标
写代码初稿、调试报错	决定判据与阈值
生成图像/视频素材	判断素材能不能用、什么算通过
整理与归纳信息	做取舍、决定放弃什么

一条我自己的原则：AI 可以把执行做得很快，但"什么叫好"必须我自己定义。

所以我在这个项目里花时间最多的地方不是写代码，而是：

- 把"内容合格"拆成 6 条可判定的约束；
- 想清楚哪两条该设成一票否决（因为它们的误放代价不可逆）；
- 反复验证判分器在"该拦的"和"不该拦的"两个方向上是否都准。

四、这份说明本身也是一种能力证明

我没有把"用了 AI"藏起来，原因很简单：

- 58. 作品集的价值不在"全部手写"，而在"判断是否成立"；
- 59. AI 时代真正稀缺的是定义标准、识别边界、发现问题的能力——而这些恰好是我在上面各表"我独立完成"一栏里写的东西；
- 60. 与其让面试官猜，不如让他看到我怎么区分"我做的"和"工具做的"——这本身就是产品经理的基本功。

五、诚实边界

项	说明
本文不含任何虚构数据	所有"无法验证"项均如实标注
表格为自评	分栏依据是我的实际记忆与文件证据，欢迎对照作品集核验
会持续更新	随着首轮评测跑分、原始素材迁移，本文的"无法验证"栏会逐项减少