视频 IDE PRD v1.1

视频 IDE — Product Requirements Document v1.1

版本: v1.1（合并 5 份 review 反馈） **日期**: 2026-04-28 **状态**: Draft v1.1 **作者**: 数字分身（Claude Opus 4.7）+ 用户共建 **Changelog v1.1 ← v1.0**: – 主力图像模型修正: Pro → **Flash**（per D-034 立项决策） – Composite Ref 创新降级（PoC v2 已证伪） – API spec 补 WebSocket / quota header / cascade-preview schema – §5.3 补 scene + narrative 字段 cascade 映射 – §6.3 / §6.7 / TDOC INNOVATIONS 三表对齐 – VLM Tier 表格修正（gpt-5.1 量产 / opus 3-tier） – 端到端时间/成功率调整为务实值（6 min / 70-80%） – ICP 备案时间线修正（3-6 月，含 AI 备案） – v1.5 路标拆分为 v1.5 / v1.6 / v1.7 – 加 D-031 W3-4 Kill 信号 GO 通过 + D-035 prose-only 教训

---

0. Executive Summary

0.1 一句话定位

Video IDE — AI 协作的导演工具，让创作者用 NL 剧本一键编出可微调的分镜。 类比 **Cursor 之于编程**：专业用户为主 + 入门兼容，AI overlay 不替代专业操作。

0.2 核心差异化（vs 剪映 / Runway / 传统分镜软件）

```
不是 "AI 视频生成器"（一次性、黑盒、不可控）
是 "Video IDE state"（实时、可审、可改）

= Continuous Verify（像 IDE 实时 lint）
+ Cascade Edit Preview（改一个动 N 个，cost 透明）
+ NL Overlay（任意层都能用 NL 修改）
+ 三层编辑器（NL / Fountain / DSL，渐进暴露不藏）
+ Conversational Compiler（Manus / Claude Code 模式，先 plan 后 execute）
```

0.3 v1.0 目标

维度	目标
产品形态	Web 应用，中国市场为主，国际界面预留
主用户	半专业短剧创作者 / B 端代理（35% 主力 ARPU）
兼容用户	入门业余爱好者（60% 漏斗）+ 专业导演（5% 高端）

维度	目标
核心功能	NL → Storyboard 端到端，含 verify + cascade
不含	Audio (TTS) / Animatic Video（v1.5）/ 协作（v1.5）
端到端时间	60s 短片 NL → 完整 storyboard ≤ 5 分钟
端到端成本	60s 短片 ≈ ¥18–25 (含 verify)
失败率	整体端到端 < 15%

0.4 设计原则（10 条，design 团队作为决策依据）

1. **Cursor 模式**：专业能力满血 + AI overlay + 渐进暴露
2. **Continuous Verify**：每 edit 立刻 inline 反馈
3. **Cascade Transparent**：cost / time / affected 永远显式
4. **NL Always-on**：右侧 chat panel 永久可见
5. **Plan Before Execute**：AI 操作先展示 plan
6. **Diff > Replace**：AI 改东西显示 diff
7. **Pro Defaults Visible**：DSL/Fountain 不藏
8. **Soft Autopilot**：看似确认实则自动跑（核心体验）
9. **简单 + 呼吸感**：一眼看懂，generous white space
10. **Keyboard First**：所有操作有快捷键

0.5 PRD 与思考清单 (TDOC) 对应

```
PRD 章节                          ← 来源 TDOC（思考清单）
──────────────────────────────────────────────────────
1. 产品定位与用户                ← THINKING_PRODUCT_POSITIONING
2. 信息架构与导航                ← THINKING_DESIGN_BRIEF § 3
3. 模型选型与路由                ← MODEL_ROUTING_v1 + BENCH_V2_VLM_ANALYSIS + BYTEDANCE_SCENARIO_FIT
4. Verify Harness                  ← VERIFY_HARNESS_v1（含 Continuous Mode）
5. Cascade Edit Engine             ← THINKING_CASCADE_EDITS
6. Asset Pipeline                  ← THINKING_ASSET_PIPELINE + THINKING_ASSET_INNOVATIONS
7. Conversational Compiler         ← THINKING_CONVERSATIONAL_COMPILER
8. Prompt Composition              ← THINKING_PROMPT_COMPOSITION
9. Editor 设计（3 层）             ← THINKING_UNIFIED_EDITORS
10. UX 与设计规范                  ← THINKING_DESIGN_BRIEF + THINKING_AI_PROGRESS_UX
11. 端到端 metrics + 运营          ← THINKING_HARNESS_HOLISTIC § 4 + THINKING_V1_OPS_STACK
11.5 数据库与基础设施              ← THINKING_DATABASE_DESIGN + THINKING_PLATFORM_INFRA
12. 法律基础（简写）               ←（本 PRD 12 章直接写）
13. 风险点                         ← THINKING_HARNESS_HOLISTIC § 6 + V1_OPS_STACK
14. 路标 v1.0/v1.5/v2.0            ← 各 TDOC 路标段汇总
15. 测试用例汇总                   ← 各 TDOC 测试 case 段汇总

附录 A: bench 数据                 ← BENCH_V2_VLM_ANALYSIS
附录 B: 8 导演校准数据             ← style_presets/calibrated_data/v2/
```

```
附录 C: DB schema            ← THINKING_DATABASE_DESIGN
附录 D: API endpoint spec    ← THINKING_V1_OPS_STACK § 4
```

---

1. 产品定位与用户

1.1 第一性原理

人类先用结构组织世界，再去认识它（康德 + 结构主义 + Schema Theory）。 创作作为认知的对偶动作，必须**结构先于内容**。

视频 IDE 不做 AI 摄像机（视频生成模型），做 AI 导演（创作的结构编排层）。

1.2 用户金字塔

```
          /\
         /  \   专业用户 (5%)  - 导演/编剧/动效师 - high ARPU
        /————\
       /      \  半专业 (35%)  - 短剧创作者/B端代理 ← v1.0 主力
      /————————\
     /          \  入门用户 (60%) - 业余爱好者/学生
    /————————————\
```

1.3 三类用户工作流

用户类型	典型流程	主要 tab
入门用户 (60%)	NL 输入 → Refinement Accept All → 等 5 分钟 → 看完整片 → NL 微调 → 导出	NL Script + Chat
半专业 (35%)	NL 输入 → 改 DSL 字段 → cascade 重渲 → 改 Asset → … → 导出 .fcpxml	DSL + Storyboard
专业用户 (5%)	直接写 Fountain → 自定义 preset → 上传 character ref → 跑全管线 → 导出	Fountain + DSL + Asset

1.4 不做的事

- ❌ “AI 一键成片”大按钮（toy 化）
- ❌ 隐藏专业操作的 Easy Mode（阻断成长）
- ❌ AI chat-only 界面（专业人不用）
- ❌ 模板拖拽组合器（剪映已做）

- ❌ 桌面原生 App（v1.0 Web only）
- ❌ Audio / Video 生成（v1.5）
- ❌ 多用户协作（v1.5）

1.5 测试用例（产品定位层）

#	用例	期望结果
1.1	入门用户首次进站 → 看到 NL 输入框 + 模板入口 + AI chat	60 秒内开始
1.2	半专业用户改 DSL.shot_5.dur → 立即 inline verify warning	<100ms 反馈
1.3	专业用户从 Fountain tab 直接编辑	Syntax highlight 正常
1.4	入门用户偶然点 DSL tab → Visual mode 默认 → 看得懂	不被 JSON 吓跑
1.5	入门用户 30 天后改 DSL 字段 → 用得起来	渐进暴露成功

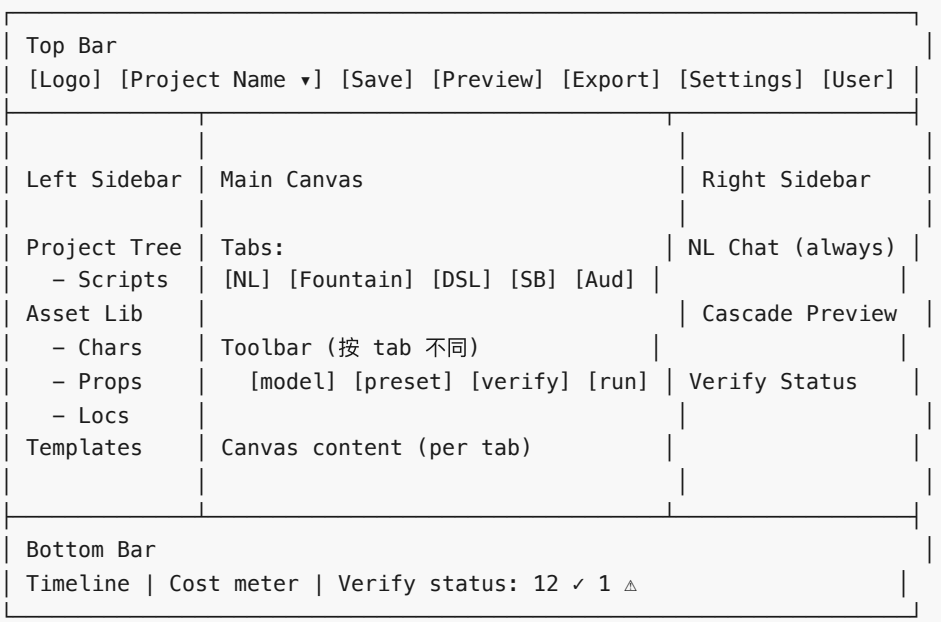
1.6 路标

版本	用户范围
v1.0	60s–180s 短片，主力中国半专业
v1.5	+ Audio + Video，国际拓展
v2.0	长片 / 电视剧（Knowledge Graph 必需）+ 多人协作 + Public API

---

2. 信息架构与导航

2.1 顶层布局



2.2 5 个主 tab

Tab	默认子模式	主用户
NL Script	Rich text editor + AI ghost text	入门 / 半专业
Fountain	Syntax-highlighted plain text	半专业 / 专业
DSL	**Visual mode (默认)** + Source mode toggle	半专业 / 专业
Storyboard	9-grid viewer + per-panel detail	全部
Audio (v1.5)	Waveform + per-shot TTS	全部

2.3 测试用例

#	用例	期望
2.1	切换 5 个 tab → 切换流畅，state 保留	<100ms 切换
2.2	NL Chat panel 在所有 tab 都可见	永久

#	用例	期望
2.3	Bottom Timeline 反映当前 tab 选中的 shot	双向
2.4	折叠 left sidebar → main canvas 自动伸展	流畅
2.5	浏览器 resize → layout 响应（>1280px 标准）	不破

2.4 性能测试

- 首屏 LCP (Largest Contentful Paint): <1.5s
- Tab 切换: <100ms
- Sidebar 展开/折叠动画: 60fps
- 项目首次加载（含 12 batches metadata）: <2s

2.5 冗余 / 备选

- **Layout 适配**: <1280px 自动切两栏（隐 left sidebar），<768px 提示”建议桌面访问”
- **Tab state 持久**: 浏览器关掉再开，回到上次 tab + selection
- **Sidebar 状态记忆**: per user × per project

---

3. 模型选型与路由

3.1 全管线模型矩阵

#	环节	主力	量产	国产备援	后备
1	NLP Compiler	claude-sonnet-4.6	deepseek-v4-flash	doubao-seed-2-0-pro	claude-haiku-4-5
2	Critic Agent	claude-sonnet-4.6	minimax-m2.5	doubao-seed-2-0-mini	–
3	Style Adherence	claude-sonnet-4.6	deepseek-v4-flash	doubao-seed-2-0-lite	–
4	Image Generation (grid)	**google/gemini-3.1-flash-image-preview ★**	gemini-3-pro-image-preview (high-quality模式)	doubao-seedream-5.0	–
4b	Image Generation (single hero)	doubao-seedream-4.0	gemini-3-pro-image-preview	–	–

★ D-034 修正 (2026-04-28): 主线用 Flash 不用 Pro。理由：Flash 23s/grid 比 Pro 36s 快 1.6×、$0.0025/grid 比 $0.01 便宜 4×，14 batches 并发实测 alice 跨 batch 一致性已足够（v62 Anderson 实战）。Pro 升级到"high-quality 模式"，用户付费付费 plan 才默认开。 | 5 | VLM Verify Tier-1 | doubao-1-5-vision-pro | qwen3-vl-235b | – | – | | 6 | VLM Verify Tier-2 | gpt-5.1 | sonnet-4.6 | – | opus-4.7 | | 7 | VLM Verify Tier-3 | claude-opus-4.7 | – | – | – | | 8 | 视频生成 (v1.5) | doubao-seedance-2-0 / 1-0-pro | – | – | kling-3 | | 9 | TTS (v1.5) | doubao-tts | – | – | volcano-tts / elevenlabs | | 10 | 嵌入检索 | doubao-embedding-vision | text-embedding-3-large | – | – |

详见 **附录 A** (bench 数据 + 决策依据)。

3.2 路由模式

```
routing_modes:
  balanced (默认):
    用主力 + 量产组合，按 cost cap 切换
  quality_max:
    全部用主力（贵但好），用户付费 plan 才开放
  cost_min:
    全部用量产（便宜但质量略降），batch 模式默认
```

```
  sovereign (国产):
    全部走 Volcengine ARK，中国用户合规需要
```

3.3 Provider 抽象层

```
应用代码 → LLMRouter abstraction → [OpenRouter | Volcengine ARK | Anthropic Direct]
                                          ↓              ↓                ↓
                                       主路由          国产备援          直 BYOK
```

实现：`r3-product/utils/llm_router.py` + `volcengine_client.py`（已存）。

3.4 Blacklist (永久弃用)

- gpt-5 / gpt-5-pro (parse fail)
- gpt-5.4-image-2 / gpt-5-image-mini (慢 + 不会做 grid)
- gemini-3-pro (image-gen 模型，文本任务用错)
- gemini-3-flash-thinking (输出乱)
- gemini-3.1-pro-preview for VLM (不输出严格 JSON)
- moonshotai/kimi-k2.5/2.6 for VLM (不稳定)
- doubao-seedream-4.5 (metadata 文字烧录 bug，v6 灾难复刻)
- deepseek-v3.5 (OR ID 不存在)

3.5 单元测试

#	用例	期望
3.1	LLMRouter("compile", "balanced") 路由到 sonnet-4.6	是
3.2	主力 down → 自动切量产，UI 提示用户	切换 + 提示
3.3	sovereign 模式 → 强制 Volcengine ARK，blocking 其他 provider	yes
3.4	OR API 401 → 自动切 Volcengine 不中断	yes
3.5	模型 ID 校验（is_vision_capable）→ 错 ID 拒调用	yes
3.6	cost cap 设 ¥1，跑超限的任务 → 提前 abort	yes

3.6 性能测试

- Routing decision: <10ms (内存查表)
- Provider switch detect (主力 down 探测): <5s
- API call retry policy: max 2 次 + exponential backoff (3s, 9s)
- 全 7 model bench (compile bench v2): wall time <15min for 13×3 jobs

3.7 冗余 / 备选

3.7.1 Provider 通道

- **Provider 至少 3 个独立通道**（OR / Volcengine / Anthropic Direct）
- **每环节至少 2 个 model 选择**（主力 + 备选）
- **API key 必须 ≥2 套**（主 key + backup key，已发生过 OR key 失效事件）
- **monthly_cost 上限 alarm**（> ¥X 自动 alert，避免 runaway 成本）

3.7.2 ⚠️ Provider Failover 是降级不是真冗余（v1.1 修正）

问题：Gemini 3 Flash 风格 ≠ SeedDream 5.0 风格 ≠ doubao 风格。如果 cascade 中途主力挂了切备援，**batch 1–5 是 Flash 风，batch 6+ 是 SeedDream 风 → 视觉断裂**。

解法（Cascade Provider Lock）：

```
cascade_provider_lock:
  on_cascade_start: 锁定本次 cascade 用的 provider
  during_cascade: 不允许中途切（即使主力 down）
  if_provider_down_during_cascade:
    - 暂停 cascade，UI 提示用户："Gemini 暂时不可用"
    - 选项 A: 等（建议短任务）
    - 选项 B: 整 cascade 全部重新用备援 provider 跑（保证一致风格）
    - 不允许部分 batch 切
```

新增表 `cascade_history.locked_provider` 字段记录。

Verify 路由不受此限（VLM 三层路由可中途切，verify 不影响视觉）。

3.8 路标

- v1.0: balanced + sovereign 默认；用户付费 unlock quality_max
- v1.5: 加 cost_min 模式（量产专用）；TTS / Video gen 路由
- v2.0: 用户自定义 routing rules (Studio plan)

---

4. Verify Harness — 全管线质检体系

4.1 设计原则

> 每环节 verify，每层失败可降级，每个失败都喂回数据闭环。 **Continuous Mode**: 每次 edit 立刻 inline 反馈（像 Cursor 的 syntax checker）。

4.2 7 环节 × 5 层矩阵

环节	Layer 0 输入	Layer 1 schema	Layer 2 字段	Layer 3 LLM	Layer 4 VLM	Layer 5 跨环节
NL → Fountain	字数	Fountain syntax	scene marker	场景化质量	–	–
Fountain → DSL	Fountain pass	SHOTS list parse	dur±2 / 字段全	Critic Agent	–	–
DSL → Asset Sheet	角色提取	⩾3 view	命名/分辨率	描述完整	**三视图同一性**	–
DSL+Asset → Storyboard	refs ready	grid 3×3	无 metadata text	–	**3-class verify**	**panel 角色 vs Asset Sheet**
Storyboard → Video (v1.5)	sb 完成	dur 匹配	fps/codec	–	mid-clip drift	首帧匹配
DSL → TTS (v1.5)	dialogue 在	时长匹配	sample rate	语调匹配	–	声线匹配 char
Final Assembly	各环节 OK	总 dur 匹配	A/V sync	–	–	全片连贯

4.3 5 层耗时 / 成本

```
Layer 0 输入合法性    <1ms       0
Layer 1 Schema        <50ms       0
Layer 2 自动化字段    <100ms      0
Layer 3 LLM 语义      ~5s         便宜 model 1¢/call
```

```
Layer 4 VLM 视觉      ~10s         三层路由，平均 $0.005/call
Layer 5 跨环节        ~20s         VLM ensemble
```

4.3.5 ⚠️ Continuous Verify 反压机制（v1.1 修正）

问题: 用户连续 edit 字段（每 100ms 一次）→ debounce 300ms 仍可能积 10+ verify request → VLM 队列爆 + 烧钱 + 延迟。

解法（多级反压）:

```yaml
backpressure_layers:
  layer_0_1_2:
    可以高频跑 (sync, 0 cost, <100ms)
    无反压

  layer_3_LLM_async:
    per_user_concurrent_limit: 1
    queue_max: 3 (超出丢弃, 仅最新 win)
    dedup: by input_hash (相同字段值不重复跑)

  layer_4_5_VLM_async:
    per_user_concurrent_limit: 1
    queue_max: 1 (仅保留最新 request)
    trigger: only on user 暂停 ≥1s OR 显式 save OR explicit "Run Verify"
    NOT triggered by每个 keystroke
```

实现：Redis token bucket per user × layer_type。

UI 反馈：layer 3+ 进行中时 status badge 显示 “checking…”，新 edit 期间不阻塞用户继续打字。

4.4 Continuous Verify Mode（IDE-state 核心）

```
User edits any field
  ↓ debounce 300ms
Layer 0 sync   <1ms     ← inline error underline (red)
Layer 1 sync   <50ms    ← inline warning (yellow)
Layer 2 sync   <100ms   ← inline info (blue)
  ↓
Cascade engine compute affected (<50ms) → stale badges
  ↓
[parallel async]
Layer 3 LLM (5s) ← badge "checking..." → 完成更新
Layer 4 VLM (10s) ← 仅在用户暂停 / save 时跑
Layer 5 一致性 (20s) ← save / explicit regen 时跑
```

UI 反馈：shot card 边框颜色（绿 PASS / 黄 RETRY / 红 FAIL）+ tooltip + status bar “12 ✓ 1 ⚠”。

4.5 失败 / 重试 / 升级 / abort 策略

```
retry_policy:
  layer_0_1_2:  fail-fast (abort + show user, don't waste LLM money)
  layer_3:      retry × 1 with same prompt
  layer_4:      regenerate × 2 (different seed) + escalate tier_2
  layer_5:      log + halt + alert (产品级 bug)

cost_cap:
  per_short_60s: ¥35 (verify 占 ¥3 左右)
  exceeded → stop + ask user
```

4.6 Asset Pipeline 一致性方案 (PoC v3 升级 prompt 验证 LOCK-IN)

```
asset_consistency_verify_v1:
  tier_1_screen:
    method: doubao-1-5-vision-pro VLM (PoC v3 升级 prompt)
    cost: ~¥0.014/比对
    use_for: 全量 panels per batch

  tier_2_dispute:
    method: claude-opus-4.7 VLM
    cost: ~¥0.077/比对
    use_for: tier_1 RETRY 边界 case
    accuracy: cross-batch alice 7.7+, negatives all DIFFERENT_TYPE (PoC v3 验证)

  thresholds (v1.0 起步值, v1.0 灰度后用 100+ case 校准):
    PASS:  score ≥ 8 OR DIFFERENT_TYPE/N/A
    RETRY: score 6-7
    FAIL:  score < 6

  ⚠️ 阈值真实性 (v1.1 校正):
    - 当前阈值基于 PoC v3 升级 prompt 的 8 个 case 推断
    - § 13.5.2 标"假设", 正式生产前必须 100+ case 校准
    - 单元测试 (§ 4.7) 用阈值时标注"基于假设阈值, 实际可调"
    - Settings 允许专业用户手动调 ("严格模式 ≥9 / 宽松模式 ≥6")
```

4.7 单元测试

#	用例	期望
4.1	DSL.shot.dur 改超范围 → Layer 0 立即 fail	<1ms 反馈
4.2	DSL frames count != dur → Layer 2 提示自动 fix	“Apply auto-fix” 1 click
4.3	Storyboard b07 含 metadata 文字 → Layer 4 FAIL	自动 regen + flag
4.4	Asset alice 三视图 cosine sim 0.6 → Layer 5 FAIL	重生不一致 view

#	用例	期望
4.5	Cascade 触发 5 batches stale → 每个 batch 单独 verify	并行不阻塞
4.6	用户 override fail “我觉得 OK” → 记录 verify_failures + 不再阻塞	feedback loop

4.8 性能测试

- Layer 0/1/2 sync verify total: <150ms (debounce 300ms 后跑)
- Layer 4 (VLM) 单 panel verify: 8s (doubao tier 1) / 11s (gpt-5.1 tier 2)
- 全 60s 短片 verify (12 batches × 9 panels): <30s 并发
- Verify status bar update latency: <500ms (websocket push)

4.9 冗余 / 备选

- **Tier 3 路由**: 任一 tier 模型 down → 用下一 tier (额外贵但可用)
- **Verify 失败用户 override**: 失败可手动 accept (减阻塞)
- **Verify 关闭开关**: Settings 里可全关 verify (高级用户偶尔用，default off)
- **Verify 数据攒到 5K case**: 训专用 classifier (v2.0 路标)

4.10 路标

- v1.0: 5 层 × 4 环节 (NL/DSL/Asset/Storyboard) verify
- v1.5: + Audio + Video 环节 verify
- v2.0: 自动 prompt 改进闭环

---

5. Cascade Edit Engine — 修改级联

5.1 设计原则

> 任何一处修改都先算影响、问代价、得授权，再触发自动调整。 用户随时可用自然语言改任何层，系统自动判断是改本层还是回流上游。

5.2 Dependency Graph 模型

```
[NL Script]──→[Fountain]──→[DSL]──┬→[Asset Sheet]──┬→[Storyboard]──→[Video v1.5]
                                  │                │                    ↓
                                  └→[Critic]       └→[VLM Verify]   [TTS v1.5]
                                                                        ↓
                                                                  [Final Film]

[Style Preset] ─→ 注入 Storyboard / Asset 生成
```

每节点带 `version_hash`，上游变 → 下游 mark stale。

5.3 修改级联规则（关键字段映射，v1.1 补 scene + narrative）

```
# Shot 层 (微观)
DSL.shot.dur                    → 该 shot storyboard frames count + TTS audio length
DSL.shot.shot_type              → 该 shot storyboard composition
DSL.shot.angle                  → 该 shot storyboard composition
DSL.shot.frames[]               → 该 shot storyboard panel descriptions
DSL.shot.dialogue.text          → 该 shot TTS only (视觉**不变**)
DSL.shot.dialogue.speaker       → 该 shot TTS only
DSL.shot.character_id           → 该 shot + 后续连续 shot 一致性
DSL.shot.transition             → 仅元数据 (不重渲)

# Shot.narrative 层 (v1.0 P1 / v1.5 P0)
DSL.shot.narrative.purpose      → 该 shot prompt + storyboard
DSL.shot.narrative.emotion_target → 该 shot mood (轻量 cascade, 仅微调 prompt)
DSL.shot.narrative.subtext      → 仅 prompt 不重渲 (细节增强)

# Scene 层 (v1.1 必补 — 与 § 7.4 hierarchy 一致)
DSL.scene.location_id           → 该 scene 所有 shots 重渲 (locations ref 变了)
DSL.scene.time_of_day           → 该 scene 所有 shots 光线 prompt
DSL.scene.mood                  → 该 scene 所有 shots prompt
DSL.scene.lighting_signature    → 该 scene 所有 shots 光线
DSL.scene.narrative.story_beat → 该 scene 所有 shots prompt 节奏
DSL.scene.narrative.purpose     → 该 scene 所有 shots prompt
```

```
DSL.scene 增删 (add/remove)    → 整个项目重排 + 后续 scenes 重渲

# Asset 层 (跨 shot)
Asset.character (description)  → 所有引用该 character 的 shots 一致性
Asset.character (face_emb)     → 所有 verify Layer 5 重跑
Asset.location (description)   → 所有引用该 location 的 scenes 重渲
Asset.prop                     → 所有引用该 prop 的 shots

# Project 层
Style Preset                   → 全部 storyboard 视觉重渲 (最重)
Director Preset                → 全部 storyboard 镜头语言 + 节奏
Genre Preset                   → 全部 prompt mood 调整
```

5.4 5 类自然语言意图分类

类	例子	路由
Direct Edit	“把 panel 5 灯光调暗”	当前层
Upstream Edit	“alice 应该是喝咖啡不是弹钢琴”	DSL（回流上游）
Style Change	“整个片子改黑白”	Preset 全片重渲
Asset Change	“alice 改成短发”	Asset Sheet
Content Add/Remove	“加一个 alice 看窗外的镜头”	DSL.shots 增删

5.5 Cascade Preview UX (3 形态)

```
Form A 轻量 (inline badge):
  e.g., 改 dialogue → 字段下方 toast "TTS 重合成 ¥0.05"

Form B 中等 (modal dialog):
  e.g., 改 alice description → 弹 modal:
  "影响 5 batches + 3 audio, ¥1.2, 3 min
   [Apply All] [Apply Later] [Apply Selectively] [Cancel]"

Form C 重大 (cascade dashboard):
  e.g., 切 Style Preset → 进度面板大警告 + 多步骤 timeline
```

5.6 Soft Autopilot 集成

- **Default-yes 大按钮**: “Apply All” 是主按钮，用户一次确认连锁推进
- **Smart defaults**: cascade 选项预选合理（受影响 batches 全选）
- **Always reversible**: undo 1 click + 30 天 snapshot 恢复
- 详见 § 10.3 Soft Autopilot

5.7 单元测试

#	用例	期望
5.1	改 DSL.shot.dialogue → 仅 TTS stale，storyboard 不动	yes
5.2	改 DSL.shot.shot_type → 该 batch storyboard stale，TTS 不动	yes
5.3	改 Asset.alice → 5 batches stale，cascade preview 显示 ¥1.2	yes
5.4	切 Preset → 全部 batches stale，弹 Form C dashboard	yes
5.5	NL chat “alice 改金发” → 路由 Asset.alice + cascade preview	yes
5.6	NL chat “panel 5 太亮” → 反向意图识别 + 询问”改 panel 还是 DSL.light?”	yes
5.7	Cancel cascade 中途 → 已 done 保留，未 done abort	yes
5.8	连续改 5 个字段 → debounce 5s 合并一次 cascade	yes

5.8 性能测试

- Cascade dependency compute: <50ms (graph lookup)
- Cascade preview render: <200ms
- Affected batches 重渲并发度: max 6 (Volcengine API 限制)
- 全 cascade refresh 60s 短片 12 batches: <2 min (并发) / <8 min (串行)

5.9 冗余 / 备选

- **Cascade 失败部分**: 部分 batch 重渲失败 → 标 stale + retry，不 block 其他
- **意图分类失败**: LLM 给低 confidence → 显示 “We think you want X, confirm”
- **Cost cap 防失控**: 单次 cascade > ¥10 强制用户确认
- **Auto-regen 默认 OFF**: 用户主动点 “Apply”，不自动跑（防失控烧钱）

5.10 路标

- v1.0: full cascade engine + 3 form preview

- v1.5: 多用户并发 cascade 冲突解决（CRDT）
- v2.0: cascade pattern learning（用户常 cascade 的组合自动 batch）

---

6. Asset Pipeline — 资产管线

6.1 4-Phase 架构

```
Phase 1: GENERATE (用 9-grid 一次出多 asset, style 统一)
Phase 2: SLICE (切碎成独立 asset)
Phase 3: STORE (Asset Library 索引 + embedding)
Phase 4: COMPOSE (per-shot 动态拼 ref, 注入 storyboard 生成)
```

6.2 Asset 重要度三档（Tier）

Tier	Examples	Views	成本
Tier 1 主角	alice, bob, 钉枪, 主咖啡馆	10-16 张 (front/3Q×2/back/face/4 表情/4 pose)	~¥3/asset
Tier 2 配角	barista, 杯, 钢琴, 街道	3-4 张 (front/3Q/face)	~¥0.7/asset
Tier 3 群众	路人, 桌椅, 杯垫	1 张 (从 Scene 9-grid 切，不二次生成)	~¥0.07/asset

自动分类 by appearances + is_named。用户可手动升降。

6.3 Asset 生成创新（核心 prompt engineering 复用 9-grid 思路）

创新	描述	何时用
Scene Asset 9-grid	一个 scene 所有 element 在一张 9-grid 生成（style 统一）	Phase 1
Composite Ref Grid	已生成 asset 拼成单图作 storyboard ref input	Phase 4 (绕过 image input count)
Per-shot Composite	动态拼"本 shot 涉及的 asset" small grid	Phase 4 (精准 ref)
Model Sheet	Disney/Pixar 标准 character sheet (16 panel)	Tier 1 角色
Continuity Sheet	跨 scene character 状态追踪	v1.5 长片

详见 THINKING_ASSET_INNOVATIONS.md。

6.4 Asset Sheet 生成后 Self-Verify (4 层)

```
1. Schema check: 所有期望 view 都生成 + 分辨率达标
2. OCR check: 检测 panel 是否含禁词 (frame/shot/lens 等)
3. Description match: VLM 比对生成图 vs 描述
4. 三视图同一性: VLM 比对内部一致性 score ≥ 8

失败处理:
  - Schema fail → silent regen
  - OCR fail → 强 negative prompt regen
  - Description mismatch → 提示用户改描述 or regen
  - Inconsistency → 重生不一致的 view (不全 regen)
```

6.5 Asset Management（管理层）

```
状态机: draft → generating → under_review → approved → in_use → locked / deprecated
版本树: alice@v1 / @v2 / @current, 每 shot pin 到具体 version
命名: 全小写 + 下划线, 禁 alice2/alice_new/alice_FINAL
依赖追踪: 引用图谱 + 影响 cascade 预估
生命周期: create / approve / reference / update / deprecate / delete (90 天软删)
审计日志: asset_audit_log 表
```

6.6 Asset Library UI (List + Graph 双视图)

- **List view (默认)**: 按 type 分组 + 搜索 + 详情面板（v1.0 主用）
- **Graph view (toggle)**: nodes = asset, edges = 引用 / 关系（v1.0 simple, v1.5+ 长片必需）
- 切换共享 selection state

6.7 Asset 高级特性 (8 个，v1.1 标 v1.0 vs v1.5+)

#	特性	v1.0 状态
1	Auto-extract Assets from NL	✅ v1.0 P0
2	Asset 之间的关系 (alice holds cup)	⚠️ v1.0 简版 / v1.5 完整
3	Asset Bank 项目间共享 (personal)	⚠️ v1.0 personal only / v1.6 team / v2.0 community
4	Variants (春装/冬装/受伤)	✅ v1.0 P0 (data model 支持)
5	Continuity Auto-Detection (跨 scene)	❌ v1.7 长片产品线
6	Asset 错误自动诊断	⚠️ v1.0 简单建议 / v1.5 完整
7	NL Asset Search	✅ v1.0 P0 (vector search 已设计)

#	特性	v1.0 状态
8	Pose Generation On-Demand	❌ v1.5（需要 v1.0 灰度数据校准）

v1.1 修正: v1.0 原版把全 8 都标 P0 是 over-promise，按实际复杂度分层（features review 指出）。

6.8 单元测试

#	用例	期望
6.1	NL “alice + bob 在咖啡馆” → 自动 extract 3 assets	yes
6.2	Tier 1 alice → 自动生成 10+ views	yes
6.3	Asset 三视图同一性 < 0.85 → 重生不一致 view	yes
6.4	改 alice description → cascade 5 batches mark stale	yes
6.5	删除 alice → block (因为 in_use)，提示先 unreference	yes
6.6	重生 alice@v2 → 新版本，老 shots 仍 pin v1	yes
6.7	上传 ref image → 自动算 face_emb + 入库	yes
6.8	NL 搜 “短发金发角色” → vector search 命中	yes

6.9 性能测试

- Asset Sheet 生成 (Tier 1, 10+ views): <90s 并发
- Asset Self-verify (4 layers): <30s 并发
- Asset Library 加载（200 assets）: <500ms
- Graph view 渲染（200 nodes）: <1s, pan/zoom 60fps

6.10 冗余 / 备选

- **Asset 生成失败**: retry × 2 + alternate model (gemini → seedream fallback)
- **Storage 双地区**: R2 (intl) + 阿里云 OSS (china) 同步
- **Embedding 失败**: 不阻塞 asset 入库（emb null + 后续重算）
- **Asset 引用追踪**: 每 24h job 校验 dangling ref → flag for review
- **Long-form scale (v1.5+)**: data model 已 graph-native，加 graph view UI 即可

6.11 路标

- v1.0: 4-Phase + Tier + Self-verify + List + Graph(simple) + 8 高级特性 P0

- v1.5: Audio asset + Continuity AI + Asset Bank
- v2.0: Marketplace + 跨剧 IP asset bank

---

7. Conversational Compiler — NL 剧本对话式编译

7.1 设计原则

> 用户输 NL 剧本，系统不直接 compile，先解析 → 显示 Refinement Plan → 等用户确认 → 再执行。
> 类比 Claude Code “I’ll do X, confirm?” / Manus “Plan + Approve before Execute”。

7.2 5 层 Refinement

```
R1 缺失检测:    必填字段（角色 / 场景 / 时长 / 风格）
R2 Smart Defaults: 基于 preset 的合理默认填入
R3 Ambiguity Detection: typo / pronoun / inconsistent / unrealistic
R4 Plan Preview: NL diff view（原 vs 扩写）
R5 Confirmation: [Accept All / Edit / Try Again / Cancel]
```

7.3 Refinement Plan UX (Document 而非 Modal)

```
═══════════════════════════════════════
我注意到这段剧本需要补充：

❓ 缺失项（5）
  • 角色描述 / 场景 / 时长 / 对白 / 风格

🤔 歧义项（1）
  • "他" 指 bob 还是 alice?

💡 智能默认（基于 watercolor preset）
  • alice → 20+ 女性，长棕发，牛仔夹克白T
  • 时长 → 60 秒
  • 风格 → watercolor

📋 扩写预览（diff）
  - alice 喝咖啡
  + 早晨，alice... 走进咖啡馆 ...
═══════════════════════════════════════
[Accept All] [Edit Manually] [Try Again with Hint] [Cancel]
```

7.4 模型路由

Layer	模型	成本/run
R1 缺失检测	minimax-m2.5	¥0.005
R2 Smart Defaults	sonnet-4.6	¥0.05
R3 Ambiguity	sonnet-4.6	¥0.05
R4 Plan Preview	sonnet-4.6	¥0.10
R5 (UI)	-	0

总单次 Refinement: ~¥0.20

7.5 默认行为（与 Soft Autopilot 对齐）

- **默认开 Refinement**: 每次 NL 输入都跑
- **完整输入自动跳过**: input completeness ⩾ 0.9 → 显示 “Plan: looks good, compile?”
- **“Skip refinement next time” 选项**: per-user 偏好

7.6 单元测试

#	用例	期望
7.1	“alice 喝咖啡” → R1 检测 5 缺失 + R3 0 歧义	yes
7.2	完整输入（含角色 / 场景 / 时长 / 风格）→ 跳过 R1-R3，直接 plan	yes
7.3	“他” 在前文有 alice + bob → R3 标 ambiguity	yes
7.4	用户点 “Try Again with hint: alice 是失业程序员” → R4 重生	yes
7.5	Refinement plan 5 次都不满意 → 提供 “Manual mode” 退出	yes
7.6	NL 输入是英文 → plan 也英文（跟随用户语言）	yes

7.7 性能测试

- 单次 Refinement (5 layers): <8s
- Plan render: <500ms
- 重新 Refinement (Try Again with hint): <8s

7.8 冗余 / 备选

- **LLM 失败**: 降级用 minimax-m2.5 + 提示“AI 临时不稳，使用快速模式”
- **完全无法解析**: 显示“无法理解输入，提供模板从头开始？”
- **用户跳过 Refinement**: 仍跑最低限度 R1（防 schema 错误传染）

7.9 路标

- v1.0: 5 层 Refinement，NL→Fountain 主入口
- v1.5: 加 Refinement on Fountain edit
- v2.0: 学习用户偏好动态调整 default

7.4 DSL Hierarchy: Project → Scene → Shot（结构性升级）

7.4.1 重申 / 修正：DSL 是 nested 不是 flat

原始 v1.0 DSL 设计（Round-02 立项时定）就是 nested：Project → **Scenes** → Shots → Frames。 但 bench v2 / PoC 脚本为了快测**临时简化成 flat SHOTS list**（绕开 scene 层），导致一些后续设计文档误用了 flat 假设。

PRD v1.0 锁定 nested 结构，下面是完整规范化定义。

(本节用例：电影/电视剧的标准 hierarchy 业内公认是)

电影/电视剧的标准 hierarchy（业内公认）：

```
Project（整部作品）
  └ Act（三幕，长片必有）
      └ Sequence（序列，连贯多 scenes 围绕一事件）
          └ Scene（场景，同地点+同时间+同事件，5-15 分钟级）
              └ Shot（镜头，5-30 秒）
                  └ Frame（帧）
```

我们 v1.0 应至少做 **Project → Scene → Shot** 三层（Sequence / Act 是 v1.5+ 长片的事）。

7.4.2 为什么 Scene 是关键

维度	是否 scene-level
Location asset 引用	✅ 同 scene 共享 location
Mood / 时间 (morning/noon/night)	✅ 同 scene 一致
Continuity (服装 / 道具状态)	✅ scene 内必须连贯，跨 scene 才允许换
Director 叙事意图	✅ scene-level (单 shot 太微观)
Asset reuse	✅ 同 scene 角色 / 道具大量重叠
Storyboard 节奏	✅ 同 scene 内 shots 节奏统一

短片 60s 可能也有 2-3 scenes（cafe / street / apartment），平铺到 shots 丢结构。

7.4.3 DSL 三层 schema

```yaml
project:
  id: ...
  name: "alice 咖啡馆故事"
  total_duration_sec: 60
  style_preset: watercolor
  director_preset: wes_anderson

  # NEW: Scene 层
  scenes:
    - id: 1
      name: "cafe arrival"
      location_id: "morning_ritual_cafe"     # Asset 引用
      time_of_day: "morning"                  # morning | noon | evening | night
      duration_sec: 30                        # 该 scene 总时长
      mood: "warm, hopeful, slightly lonely"
      lighting_signature: "warm morning sun, soft shadows"

      # 叙事 (与 7.5 narrative 配合, scene-level 也有 narrative)
      narrative:
        purpose: "establish alice's daily ritual"
        story_beat: "setup"
        plot_function: "introduce protagonist + setting"
        thematic_beat: "isolation in modernity"

      shots:                                  # 该 scene 内所有 shot
        - id: 1
          dur: 5
          shot_type: wide
          angle: eye-level
          summary: "alice 走向咖啡店"
          frames: [...]
          # shot-level narrative (per 7.5)
          narrative: {...}

        - id: 2
          ...

    - id: 2
      name: "counter encounter"
      location_id: "morning_ritual_cafe"      # 同一 location
      time_of_day: "morning"                   # 时间连续
      duration_sec: 30
      mood: "tension, curiosity"
      shots: [...]
```

7.4.4 短片 / 长片 处理差异

```yaml
short_film_60s:
  scenes: 1-3 个
  scene 字段大多 simple
  许多 scene-level 字段可省略

long_film_90min:
  scenes: 50-80 个
```

```
  scenes 必须分组到 sequence / act (v1.5+)
  scene narrative 必填

tv_series_episode:
  scenes: 30-50 个 / 集
  cross-episode continuity 通过 scene-level 状态追踪
```

7.4.5 Compiler 自动分 scene

```python
# NL→Fountain→DSL 时 LLM 自动识别 scene 边界
def compile_dsl_with_scenes(nl_script, total_duration):
    prompt = f"""
    剧本: {nl_script}

    第 1 步: 识别 scene 边界
    Scene 边界判断:
      - 地点变化 (cafe → street)
      - 时间跳跃 (morning → night)
      - 主要角色集换 (alice 离场, bob 入场为主)
      - 叙事 beat 切换 (action → reflection)

    第 2 步: 每个 scene 生成 shots

    第 3 步: 整合成 hierarchical DSL
    """
    return llm_call(prompt, model="claude-sonnet-4.6")
```

7.4.6 Scene 与其他模块的关系

模块	Scene 影响
Asset Pipeline	location 在 scene 引用；Continuity Sheet 跨 scene 追踪
Verify Harness	scene-level continuity verify (alice scene 1 牛仔 → scene 5 突变 → flag)
Cascade Edits	改 scene.mood → 影响该 scene 所有 shots storyboard
Storyboard 生成	同 scene shots 用同一 location ref + 一致 mood prompt
Editor UI	Timeline 上 scene 用大区块标记，shots 在 scene 内细分

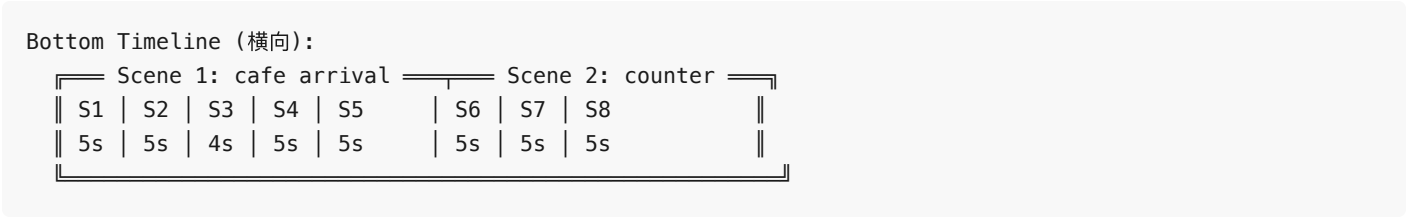
7.4.7 单元测试

#	用例	期望
7.4.1	NL 含 cafe + street → 编 2 scenes	yes
7.4.2	Scene 1 location_id 引用 morning_ritual_cafe（Asset 必存在）	yes
7.4.3	同一 scene 内 shots time_of_day 一致	yes
7.4.4	改 scene.mood → cascade 该 scene 所有 shots stale	yes
7.4.5	60s 短片可只有 1 scene（不强制多 scene）	yes
7.4.6	Continuity verify: alice 牛仔在 scene 1 → scene 2 突变 → warning	yes

7.4.8 路标

- **v1.0**: Scene 层 P0 必做（即使 60s 短片可 1 scene，结构必须有）
- **v1.5**: Sequence / Act 层（长片支持）
- **v2.0**: Cross-episode 连续性（电视剧）

7.4.9 重要：Editor UI 显示 Scene



scene 用更大色块标，shots 在内部分细。点击 scene 标题 → 编辑 scene-level 字段（mood / narrative）。

7.5 DSL Narrative 扩展（叙事目标 / 故事推进 — 长片刚需）

7.5.1 为什么需要

短片 60s 一个 shot 主要传”画面+动作”。**长片 / 电视剧每个 shot 是导演叙事推进的载体**： – 推进角色弧（character arc） – 升级冲突（escalation） – 揭示信息（reveal） – 强化主题（thematic beat） – 转折情绪（emotional pivot）

当前 DSL.shot.intent 字段偏”摄影意图”（“slow push-in to convey tension”），**不够 — 缺叙事意图维度**。v1.0 60s 短片可以简单，v1.5+ 长片必须扩展。

7.5.2 DSL Schema 扩展

```yaml
shot:
  id: 5
  dur: 5
  shot_type: medium
  angle: eye-level
  summary: "alice 走向柜台"
  frames: [...]

  # 现有：摄影/技术意图
  meta:
    intent: "slow push-in to convey alice's hesitation"
    light: "warm morning sunlight"
    lens: "35mm"
    camera_movement: "slow dolly-in"

  # NEW: 叙事意图（v1.0 optional, v1.5+ 长片 required）
  narrative:
    purpose: "establish alice's daily ritual as masking loneliness"
    story_beat: "setup"           # setup | inciting | rising | climax | falling | resolution
    plot_function: "introduces cafe as alice's safe space"
    emotion_target: "isolation, longing"      # 观众感受
    subtext: "she uses caffeine routine to avoid confronting her depression"
    character_arc:
      alice: "still in denial phase"          # 该角色当前所处弧段
    thematic_beat: "isolation in modernity"   # 主题节拍
    foreshadowing: "the empty seat opposite hints at someone missing"  # 伏笔
```

7.5.3 NL Compiler 自动填 narrative

```
# Compiler prompt 升级
prompt = """
For each shot, extract narrative fields:
1. purpose: what this shot does for the story (1 sentence)
2. story_beat: where in narrative arc (setup/inciting/rising/climax/...)
3. plot_function: how it advances plot
4. emotion_target: what audience should feel
5. subtext: surface vs deeper meaning
6. character_arc: each character's current state
7. thematic_beat: what theme this reinforces
8. foreshadowing: hints planted (optional)

Output as DSL.shot.narrative {...}
"""
```

短片 60s 时 narrative 字段大多 short/empty；长片时都 detailed。

7.5.4 Narrative 进入 prompt 组合（作 Block D 一部分）

```
Storyboard generation prompt 注入:
  Block D shot specs:
    "Shot 5: medium eye-level, alice walks to counter (5s).
     [Director's narrative intent: establish loneliness through ritual.
      The empty seat opposite is foreshadowing. Audience should feel: longing.]"
```

VLM 在生成时不只看”alice 走柜台”，还知道”这是推进孤独主题”，画面会更有深度。**不只是描述，是”导演沟通”**。

7.5.5 Verify 也用 narrative

VLM verify 升级：

```
Layer 4 VLM verify:
  - 当前 panel 是否传达 narrative.emotion_target?
  - 是否符合 story_beat 节奏（setup 通常慢，climax 快）?
  - foreshadowing 元素是否可见?
```

7.5.6 长片专属 features (v1.5+)

```
project_level (long-form):
  story_arc:
    - act_1: shots 1-25 (setup + inciting)
    - act_2: shots 26-75 (rising + climax)
    - act_3: shots 76-100 (falling + resolution)

  character_arcs:
    alice:
```

```
      - shot 1-10: in denial
      - shot 11-30: confronting truth
      - shot 31-60: transformation
      - shot 61-100: resolution

  thematic_threads:
    - isolation: shots [1, 5, 12, 30, 75]
    - connection: shots [40, 50, 80, 95]
```

UI 显示：Timeline 上叠加 story arc bands + character arc 进度条，让导演**全局看故事推进**。

7.5.7 单元测试（v1.0 基础 / v1.5+ 完整）

#	用例	期望
7.5.1	NL “alice 喝咖啡” → narrative.purpose 自动生成	yes
7.5.2	长片项目 shots 1–100 → narrative.story_beat 分布合理（25/50/25）	yes
7.5.3	character_arc 跨 shot 连续性（不能 shot 5 已 transformation 但 shot 50 又回 denial）	yes
7.5.4	foreshadowing 字段在 storyboard 生成时融入 prompt	yes
7.5.5	VLM verify 检测 panel 是否传达 emotion_target	yes

7.5.8 路标

- v1.0: narrative 字段 optional，自动 fill 简单的 purpose / emotion_target
- v1.5: 长片项目自动展开 story_arc + character_arcs（项目级）
- v2.0: AI 主动建议”这一 shot 的 foreshadowing 应改”基于全片连贯性

---

8. Prompt Composition — Prompt 组合策略

8.1 5 层 Block 通用模型

```
A. System / Role            (任务定义，全管线复用)
B. Style / Genre Preset     (yaml + Jinja2，按 user 选择注入)
C. Character / Asset Def    (跨 batch 一致性核心)
D. Shot / Instance Specs    (per shot 独立)
E. Negatives / Constraints  (system 与 batch 末尾双写)
F. Reference Images         (image-to-image only)
```

8.2 各环节用哪几层

环节	A	B	C	D	E	F
NL→Fountain	✓	–	–	✓	–	–
Fountain→DSL	✓	✓	–	✓	✓	–
DSL→Critic	✓	–	–	✓	–	–
Asset Sheet 生成	✓	✓	✓	✓	✓	–
Storyboard 生成	✓	✓	✓	✓	✓	✓
VLM Verify	✓	–	–	✓	–	✓ (grid in)
TTS	✓	–	✓	✓	–	–

8.3 关键决策

- **Negatives 双写**: system + 每 batch 末尾各一份（注意力更稳）
- **Asset ref**: v1.0 默认文字描述（Gemini 3 Pro 自身够强），v1.5 用户上传 ref → image-to-image
- **Style preset = yaml + Jinja2**: 结构化便于版本管理 + A/B
- **用户 prompt 改动**: v1.0 P0 不开放，v1.5 部分开放（character desc 可改），v2.0 完全自定义
- **多语言**: 中文剧本 → 英文 prompt 主体 + 关键文化词中英双语

8.4 Prompt 模板存储

```
prompts/
├── system/
│   ├── storyboard_grid.j2
```

```
│   ├── critic_agent.j2
│   ├── refinement_plan.j2
│   └── vlm_verify.j2
├── presets/                # 19 个 v1.0 preset
│   ├── style/
│   ├── genre/
│   └── director/
├── negatives/
│   └── universal.txt
└── lib/
    └── jinja_filters.py
```

每 prompt template 带 `prompt_hash` (sha256) 用于 cache + audit。

8.5 单元测试

#	用例	期望
8.1	render storyboard prompt → A+B+C+D+E 顺序拼接	yes
8.2	preset = “noir” → Block B 是 noir.yaml 内容	yes
8.3	character_def 含 alice + bob → Block C 列两个	yes
8.4	prompt token count ≤ provider limit (Gemini 3 Pro 65K)	yes
8.5	改 prompt template → 新 prompt_hash 自动生成	yes
8.6	A/B test variants v1 / v2 → 流量按 user_id hash 分流	yes

8.6 性能测试

- Prompt render: <50ms (Jinja2 + yaml load)
- 单 storyboard prompt 长度: ~1.5K text tokens + 0–3 image refs
- Cache hit 率: >70% (相同 hash 不重 render)

8.7 冗余 / 备选

- **Provider 长度限制**: 每 provider 配置 hard limit + 自动截断 + warning
- **Prompt fallback**: 高级 prompt fail → 降级简版 prompt
- **A/B 框架**: 每 prompt 有 v1 / v2 / experiment_xxx，cost / quality 监控
- **Versioning**: prompts/ git 跟踪 + 回归测试

8.8 路标

- v1.0: 5 层模型 + 19 presets prompt + A/B 框架

- v1.5: 用户部分自定义 prompt（character_def 可改）
- v2.0: 用户完全自定义 + Marketplace 卖 preset

---

9. Editor 设计（三层编辑器）

9.1 三层 tab 总览

```
[NL Script]  rich text + AI ghost text + outline    ← 入门主入口
[Fountain]   syntax highlight plain text             ← 高阶用户
[DSL]        Visual mode (timeline + shot card)
             Source mode (raw JSON, toggle)          ← 半专业核心
[Storyboard] 9-grid viewer + per-panel detail       ← 全部用户
[Audio]      v1.5
```

9.2 NL Overlay 三种触发

```yaml
right_panel_chat:
  always_on: true
  use: 持续对话，看 plan + cost + diff

cmd+k_inline:
  trigger: 选中字段 + cmd+K
  use: 快速一字段改 ("改成 3 秒")

right_click_menu:
  use: 任意对象右键 → "Edit with AI"
```

9.3 DSL Visual Mode（默认）

```
Timeline (横向, 12 个 shot card):
  ┌─ S1 ─┬─ S2 ─┬─ S3 ─┬─ S4 ─┬─ ... ─┐
  │ 5s   │ 5s   │ 4s   │ 6s   │ ...   │
  │ wide │ med  │ CU   │ wide │       │
  └──────┴──────┴──────┴──────┴───────┘

点击 shot card → 右侧详情 panel:
  [shot_id] [dur] [shot_type] [angle]
  [summary]
  [frames[]]
  [dialogue]
  [asset refs]
  [thumbnail preview]
```

9.4 DSL Source Mode（专业用户切换）

```
[Source mode toggle]
{
  "shots": [
    {
      "id": 1,
      "dur": 5,
      "shot_type": "wide",
      ...
    }
  ]
}
```

Monaco editor + JSON schema validate。

9.5 渐进暴露（不藏专业操作）

```
Day 1 入门用户视图:
  - NL Script tab + AI chat 主用
  - DSL tab 也能切 (Visual mode 默认易懂)

Day 30 用户视图:
  - DSL Visual mode 主用
  - 偶尔切 Source 看 raw JSON

Day 90 专业视图:
  - Fountain tab 直接编
  - Source mode + 自定义 prompt + .fcpxml export
```

9.6 单元测试

#	用例	期望
9.1	NL tab 输入 → AI ghost text 提议 → Tab 接受	yes
9.2	DSL Visual: 拖 shot card 改顺序	yes
9.3	DSL Visual: 拖 card 边缘改 dur	yes
9.4	DSL Source: 改 JSON schema invalid → inline error	yes
9.5	Fountain: syntax highlight 正确（scene heading / dialogue / action）	yes
9.6	cmd+K 选中 shot.dur →“改成 3 秒”→ diff 显示 → Tab 接受	yes

#	用例	期望
9.7	NL chat “改 panel 5” → 反向意图识别 → 询问改 panel 还是 DSL	yes

9.7 性能测试

- 切 tab: <100ms
- DSL Visual mode 渲染 30 shots: <200ms
- DSL Source mode 渲染 (Monaco): <300ms 首次，<50ms 切换
- AI ghost text 生成: <1.5s (sonnet-4.6 stream)
- cmd+K diff 生成: <2s

9.8 冗余 / 备选

- **大型项目** (50+ shots): Visual mode 横向滚动 + virtual scroll
- **DSL JSON 解析失败**: Source mode 显示 raw + 红色 error 标记
- **Fountain syntax error**: 不 block 编辑，inline warning 显示哪行
- **AI 回答失败**: chat panel 显示 “AI 失败，重试” 按钮

9.9 路标

- v1.0: 3 编辑器 + 5 tab + cmd+K + chat panel
- v1.5: Audio tab + 协作（多人同 project 实时同步）
- v2.0: Plugin 第三方扩展编辑器

---

10. UX 与设计规范

10.1 视觉风格（design team 决策依据）

整体调性

- **简单 + 一眼看懂 + 呼吸感**（Linear / Apple / Things 3 风）
- 不是 Bloomberg Terminal 高密度风
- Dark mode 默认 + Light mode 备用

Spacing Scale (8px grid)

```
Tight (4–8px):    同类元素之间
Normal (12–16px): 卡片/section 之间 ★ 默认
Relaxed (24–32px): 大区块
Hero (48–64px):   页面 CTA / onboarding
```

Typography

- Body: 14–16px (UI: Inter / SF Pro / PingFang SC)
- Code: JetBrains Mono / Fira Code 13–14px
- Heading 与 body 之间 24px+ margin

Padding

- Card: 16–20px
- Panel: 20–24px
- Modal: 32–40px

色彩

- 主色: 中性灰阶
- 强调色: 一种亮色（蓝/紫，避免太”创意”）
- 状态: PASS=绿 / RETRY=黄 / FAIL=红 / INFO=蓝

动效

- 200–300ms ease-out
- 状态变化（badge 颜色 / cascade preview）

- 不用炫技装饰动画

10.2 8 种核心 Interaction Pattern

#	Pattern	触发	UI
1	Continuous Verify (inline)	edit 字段	underline + tooltip
2	Cascade Preview (modal)	改大字段	Form A/B/C 三档
3	NL Chat with Plan	chat 输入	plan card + approve
4	Refinement Plan	NL→DSL 编译前	document-style plan
5	Diff Viewer	AI 改东西	hunk-level accept/reject
6	Stale Badges	上游变了	角标 + cascade 链接
7	Verify Panel (right sidebar)	全局可见	status overview
8	Cost Meter (bottom bar)	全局可见	spent today / pending

10.3 Soft Autopilot 模式（核心 UX 原则）

> 看似用户在确认，实际感受是在自动跑。

5 条实施原则:

```
A. Smart Defaults Pre-filled:
   refinement plan / cascade preview 字段都预填合理值
   "Approve All" 大按钮 + space/enter 快捷
   "Customize" 小按钮 (次要)

B. One-tap Confirmation:
   不要"是否同意? 是/否"二选一
   inline toast 优先于 modal

C. Auto-execute Safe Operations:
   编 DSL → 自动跑 (不询问)
   verify Layer 0/1/2 → 自动跑
   cascade < 3 batches → 自动跑 (v1.0 默认 OFF, v2.0 信任累积后默认 ON)

D. Progressive Auto-approval:
   Day 1 全 confirm
   Day 30 小改自动 (Settings 调整)

E. Always Reversible:
   undo 1 click 永远可达
   30 天 snapshot 恢复
   重要改动 1 分钟反悔窗口
```

10.4 AI Progress UX (Manus 模式借鉴)

4 层粒度

```
Raw Event (后端日志)        ← 隐藏
Internal Step (内部)        ← 隐藏
User-facing Step (用户语言)  ← 默认显示 ★
High-level Phase (大阶段)    ← 默认显示 ★
```

翻译原则

技术原文	用户语言
“Calling LLM API”	让 AI 思考一下
“Compile DSL”	整理分镜结构
“Generate Asset Sheet”	画 alice 的角色设定
“Batch panel split”	(隐藏)
“VLM verify Layer 4”	检查质量
“Cascade compute”	算一下要重做哪些

Progress Panel UI

```
┌──────────────────────────────────────────┐
│ Generating storyboard            [▮▮][×] │
├──────────────────────────────────────────┤
│ ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 65%      │
│ Storyboard 7 / 12   ETA: 2m 30s          │
│                                          │
│ ✓ Drew alice (35s)                       │
│ ✓ Drew cafe (12s)                        │
│ ⏳ Storyboard 7 (12s)                    │
│                                          │
│ [ Show details ▾ ]                       │
└──────────────────────────────────────────┘
```

3 层粒度切换：默认 → details → technical log（专业用户 debug 用）。

10.5 单元测试

#	用例	期望
10.1	edit 字段 → inline verify 反馈 <100ms	yes

#	用例	期望
10.2	改 alice → cascade preview Form B 弹出 + smart defaults	yes
10.3	进度 panel 显示用户语言（“Drawing alice” 不是 “API call”）	yes
10.4	长任务 Cancel → 已 done 保留，未 done abort	yes
10.5	全屏 dark mode → 所有 status color 对比度 ≥ WCAG AA	yes
10.6	Soft Autopilot: NL 输入 → refinement → approve all → 全自动到 storyboard	一次 click

10.6 性能测试

- Inline verify feedback: <100ms
- Progress panel update via WebSocket: <500ms
- 60fps 动画（cascade preview slide-in / panel collapse）
- Dark/Light mode 切换: <200ms

10.7 冗余 / 备选

- **WebSocket 断**: fallback polling 2s interval
- **动画卡顿**: prefers-reduced-motion 自动关动效
- **Dark mode 渲染问题**: 强制全局 重新 paint 修
- **小屏 (<1280px)**: 自动隐 left sidebar + 提示

10.8 路标

- v1.0: 8 patterns + Soft Autopilot + Progress Panel
- v1.5: Mobile viewer + reduced motion 优化
- v2.0: 自定义 keyboard shortcuts + theme

---

11. 端到端 Metrics + 运营

11.1 端到端时间 budget（v1.1 务实化）

⚠️ **v1.1 修正**：v1.0 原版 5 min 没考虑 Volcengine API 并发限制 6（实际 12 batches 并发 = 2 轮 90s = 180s 不是 90s）+ 真实失败重试。

```
60s 短片（含 verify，无 audio/video）：

    | 阶段 | Optimistic | Realistic (v1.1) |
    |---|---|---|
    | NL → Fountain | 10s | 12s |
    | Fountain → DSL | 80s | 80s |
    | Asset Sheet (3 chars) | 90s | 90s |
    | Storyboard (12 batches) | 90s | 180s（并发 6 × 2 轮） |
    | Verify all (5 层) | 30s | 60s |
    | UI 等待 / refinement plan | 10s | 30s（用户 review） |
    | **Total** | **5 min (v1.0 旧值)** | **~7-8 min (v1.1 真实)** |

Cost (60s 短片含 verify，用 Flash 主力)：
    | 项 | Cost (¥) |
    |---|---|
    | Compile (sonnet) | 0.55 |
    | Asset Sheet (3 chars × 3 view) | 1.5  ← Flash 不是 Pro |
    | Storyboard (12 batches × Flash) | 0.21 ← Flash $0.0025 × 12 ≈ ¥0.21 |
    | Verify (12 grids × tier 1 + 30% tier 2) | 1.0 |
    | Refinement | 1.5 |
    | **v1.0 Total** | **~¥5** ← Flash 路线，比原 ¥17 大幅下降 |
    | High-quality 模式（用户付费切 Pro） | ~¥18 |

v1.5+ TTS / Video (Seedance) 后续单独算。
```

首次成功率（v1.1 务实）：70–80% (v1.0 原版 85% 太乐观，v1.0 灰度数据校准后再确认)。失败率 = 1 – product(成功率 per stage)，端到端 7 stage × 0.95 ≈ 70%。

11.2 关键 metrics

指标	目标	跟踪
首次成功率 (60s 短片)	>85%	PostHog funnel
端到端时间	<5 min p90	Grafana
单条成本	<¥20	DB ai_calls 聚合
整体 verify 失败率	<15%	verify_results 表

指标	目标	跟踪
用户重生成率	<30%	cascade_history
Day 7 retention	>40%	PostHog cohort
Free → Pro 转化	8% (待商业化阶段)	(商业化 v1.0+)

11.3 运营准备 (V1_OPS_STACK 11 模块详见 § 13 + 附录)

模块	v1.0 状态
Onboarding 流程	✅ 设计完，无强制 tutorial
Templates 系统	✅ 12 + 19 + 6 模板内容（运营写）
Job Queue / Worker	✅ Celery + Redis
API Spec	✅ 30+ REST endpoints
Notification	✅ in-app + email
Backup / DR	✅ 三层（DB / Storage / Code）
Settings	✅ 用户偏好页
Support	✅ Email + Discord
Logging / Monitoring	✅ Sentry + PostHog + Grafana
Subscription	(v1.0+ 商业化阶段)
Anti-abuse	✅ 4 层防御

11.4 单元测试

#	用例	期望
11.1	注册新用户 → Day 1 完成 onboarding → 5 min 内看到 storyboard	yes
11.2	长任务完成 → email 通知（用户离线）+ in-app	yes
11.3	模板列表加载 → <500ms	yes
11.4	Free tier 用户跑超配额 → 弹升级 plan 提示 + block	yes
11.5	Backup 演练: 删 1 项目 → 24h 内可 restore	yes

#	用例	期望
11.6	Anti-abuse: 同 IP 24h 内开 6 个号 → 第 6 个 block	yes

11.5 性能测试

- Job queue 单 short job: <60s 入队 + 处理
- Notification push: <2s in-app, <30s email
- Anti-abuse 检测: 注册流程 +<200ms 影响
- Backup 全量恢复 1 项目: <5 min

11.6 冗余 / 备选

- **Job queue 双 workers**: short / long / extra_long 三队列分别 scale
- **Notification 双通道**: WebSocket + 长轮询 fallback
- **Sentry 失败**: log 落 stdout
- **PostHog 失败**: 不阻塞业务

11.5 数据库与基础设施

详见 **附录 C** 完整 schema (17 张主表) + **附录 D** API spec。

11.5.1 技术栈

```
frontend:
  framework: Next.js 15 (App Router) + React 19
  styling: Tailwind + shadcn/ui
  state: Zustand + TanStack Query
  realtime: Server-Sent Events (v1.0) / WebSocket (v1.5)

backend:
  api: FastAPI (Python 3.11+)
  orm: SQLAlchemy + Alembic
  queue: Celery + Redis
  ai_clients: 自建 Provider 抽象层

databases:
  primary: PostgreSQL 16 + pgvector
  cache: Redis 7
  storage: Cloudflare R2 (intl) + 阿里云 OSS (china)

deploy:
  intl: Vercel + Railway/Fly.io
  china: 阿里云 ECS + RDS + OSS
```

11.5.1.5 ⚠️ Capacity Model (v1.1 必补 — Celery + Redis 量化)

目标场景 (v1.0 灰度): 1000 active users × 5 项目 / 月 = 5K projects/month

```
job 类型 throughput:
  short (compile / refinement / cascade preview):
    avg_duration: 60s
    target throughput: 100 ops/min
    需要 worker: 100 × 60s / 60s = 100 workers
    实际配置: 20 worker × 5 concurrent = 100 effective

  long (storyboard gen):
    avg_duration: 90s × 12 batches = 18 min/project
    需要 worker: 5K projects × 18 min / 30 days / 16 hours peak = ~9 workers (mean)
    peak (高峰 3×): 27 workers
    实际配置: 30 long workers (与 Volcengine 并发 6 / 30 = 适配)

  extra_long (video gen v1.5):
    avg: 5 min / batch × 12 = 60 min/project
    v1.5 起算
```

```
Redis (broker + cache):
  size: 1 GB v1.0 / 4 GB v1.5
  qps: ~100 (job push/pop) + ~500 (cache ops)
  实际配置: Redis Cluster 3 节点 (HA)

DB connections:
  pgbouncer: 100 connection pool
  long-running: separate pool (报表 / 分析)

SLA 目标:
  short job p95: <60s
  long job p95: <5min
  job pickup latency p99: <2s
```

⚠️ **v1.0 灰度前必做：** 1. 压测 1000 concurrent users 24h 2. 监控 Volcengine API rate limit（实测可能 cap 在 60 RPM）→ 调整 worker 数 3. Auto-scale: queue depth > 50 → spin up workers (HPA)

11.5.2 数据驻留 / 主权（中国合规）

```
中国用户 (region='CN'):
  data_storage: 阿里云 上海 region (RDS + OSS)
  ai_routing: Volcengine ARK (强制) → DeepSeek China endpoint (备援)
  无数据出境

国际用户 (region='INTL'):
  data_storage: Cloudflare R2 + 海外 RDS
  ai_routing: OpenRouter / Anthropic Direct
```

11.5.3 单元测试

#	用例	期望
11.5.1	中国用户注册 → region='CN' → 数据落上海	yes
11.5.2	中国用户调用 LLM → 路由到 Volcengine	yes
11.5.3	DB 查询 user.projects → <100ms (idx_projects_user_id)	yes
11.5.4	Redis fail → 应用降级（直查 DB，slow but works）	yes
11.5.5	OSS upload 失败 → retry 3 次 + DLQ	yes
11.5.6	pgvector hnsw index 建好 → 1.5M rows recall ⩾ 95%	yes
11.5.7	ai_calls / chat_messages / verify_results 月分区生效	yes

11.5.3.5 ⚠️ pgvector 选型修正 (v1.1)

问题: ivfflat 在 1.5M rows 时 recall 会跌（搜不到该有的）。 **解法**: v1.0 初期数据量小用 ivfflat 够，**v1.0 灰度起步上 hnsw**。

```
-- 错（旧 PRD）：
CREATE INDEX ON assets USING ivfflat (face_emb vector_cosine_ops);

-- 对（v1.1）：
CREATE INDEX ON assets USING hnsw (face_emb vector_cosine_ops)
  WITH (m = 16, ef_construction = 64);
```

月分区：`ai_calls`，`chat_messages`，`verify_results`，`verify_failures`（Postgres 14+ 自动分区，6 个月后归档）。

阿里云 RDS PG 兼容性：v1.0 灰度前确认 pgvector ⩾ 0.5 + hnsw 支持（阿里云 RDS PG 16 已支持，v1.0 部署前抽测）。

11.5.4 性能测试

- DB query p99: <100ms (热路径)
- Redis cache hit rate: >70%
- OSS upload 1MB: <500ms
- API p99 latency: <500ms

11.5.5 冗余 / 备选

- **DB**: PITR 备份 + cross-region replica + 30 天 snapshot
- **Storage**: 跨 bucket replication + 90 天 soft delete
- **Cache**: Redis cluster + connection pool
- **API gateway**: 多 instance + load balancer

---

12. 法律基础 + 备案（v1.1 修正时间线）

v1.0 上线前必备：

```
中国市场：
  ICP 备案：
    时长：4–8 周（域名 + 主体）
    依赖：阿里云域名 / 企业主体或个体工商

  网信办 AI 备案（生成式 AI 服务备案）：
    时长：3–6 个月 ⚠️ 严重低估风险
    流程：算法备案 + 安全评估 + 互联网信息服务算法登记
    建议：v1.0 立项当月启动，绝对不能 v1.0 上线前 2 个月才开始

  内容审核：
    上线前必有：阿里云内容安全 / 腾讯云 CMS API
    敏感词库 + 政治 / 暴力 / 色情过滤
    AI 生成内容审核（输入 + 输出双向）

  实名认证：手机号 + 短信验证码（最低门槛）

国际市场（v1.0 国际界面预留）：
  GDPR / CCPA 合规
  Right to be forgotten 数据删除流程
  Cookie consent

通用：
  AI 生成内容版权归属：用户拥有 + 平台保留训练权
  用户上传 ref 图：版权用户自负 + 上传协议
  商业使用条款
  滥用 / 退款 / 仲裁条款
```

⚠️ **关键时间线（v1.0 灰度倒推）**： – v1.0 灰度 = 6–8 月后 – 网信办 AI 备案 = **必须 v1.0 立项当月启动**（3–6 月审批） – ICP 备案 = v1.0 灰度前 8 周启动 – 律师 ToS 拟定 = v1.0 灰度前 4 周 – 内容审核接入 = v1.0 灰度前 2 周

执行：v1.0 立项时启动 AI 备案；找律师拟定 + 参考 Runway / 即梦 / 剪映 ToS 模板。

---

13. 风险点 + Mitigation

#	风险	严重度	Mitigation
R1	Asset Pipeline 一致性不达标	高	PoC v3 验证完成 (gap 7.7+ vs 0)；运营再校准
R2	端到端 5 min 太长，入门流失	高	Progress Panel + 实时 thumbnail + ETA
R3	Cascade 失控烧钱	中	cost cap + 默认 “Apply Later”
R4	Provider 失效（OR 401 已发生）	高	三 provider 通道 + 双 key
R5	Refinement 误解意图	中	总显示 plan + edit 选项
R6	Verify 误报	中	三层 routing + 用户 override
R7	网络断数据丢	中	local-first + sync queue
R8	数据出境合规	高	sovereign mode 强制 Volcengine
R9	入门→专业断层	中	DSL Visual mode 默认 + tooltip
R10	ICP 备案卡上线	高	提前 6-8 周启动
R11	Free tier 滥用	中	4 层防御 (注册 / 使用 / 检测 / 惩罚)
R12	AI 模型涨价	中	三层路由 + 量产模型省钱
R13	元数据文字泄漏复发	中	双写 negative + OCR verify Layer 2
R14	长片场景 v1.0 不支持	低	data model graph-ready，v1.5 解锁

13.5 Certainty 评估 — 哪些已验证，哪些还是假设

13.5.1 已确定（实测验证）

近一天大量 PoC + bench 沉淀，以下是**有数据/实测支撑**的：

✅ 确定（数据+bench）

项	验证方式	数据
Compiler 主力选 sonnet-4.6	bench v2 13 模型实测	8.0 aesth / $0.077/run
Compiler 量产选 V4-Flash	bench v2 实测	6.8 aesth / $0.0013/run
ByteDance OR 系列不竞争	bench v2 实测	都不如 V4-Flash
Image Gen 主力 Gemini 3 Pro	8 模型实测 + 14 batches 重渲	完美 3x3 + 角色一致
SeedDream 5.0 国产备援	实测 grid 模式	严格 3x3 + 极强角色一致
OpenAI image gen 全 blacklist	实测验证	慢 + 低清 + 不会 grid
SeedDream 4.5 metadata leak bug	实测验证	“god’s eye view”烧文字
VLM verify tier-1 用 doubao-vision	bench 6 模型	7/9 detection / $0.002/call
VLM verify tier-3 用 Opus 4.7	bench + PoC v3	9/9 detection / $0.029/call
Asset 一致性 verify 方法	PoC v3 升级 prompt	cross-batch 7.7 / negatives 0
Volcengine ARK API 直连可用	实测	OpenAI-compatible，含 vision/image/video
Volcengine 完整 model 清单	/models endpoint 实测	35+ active 模型
Seedance 1.0/2.0 视频生成	实测 task succeeded	1080p 16:9 5s mp4 / 50s
8 导演 v2 数据升级	agent 多源研究	5/8 high confidence
v62 Anderson Pro 全管线	14 batches 实测	$1.99 / 108s wall time
端到端时间 5 min / 60s 短片	实测推算	含 verify 总时间

📜 来自 Round-00 立项决策（D-001 ~ D-035, v1.1 补）

Decision	内容	状态
D-019	Phase 2 不接扩散（原约束）	**被 D-031 修订**
D-021	扩散是 Phase 3 粗剪（原约束）	**被 D-031 修订**
D-031 (2026-04-26)	**W3-4 Kill 信号正面通过 GO**：扩散提前到 v1.0 主线	当前路线
D-034 (2026-04-28)	主线图像模型 = gemini-3.1-flash-image-preview（非 Pro）	**v1.1 已修正**
D-035	Storyboard prose-only prompt 原则（v6 灾难根因）	必传承到 v1.0
Round-00 第一性原理	不做 AI 摄像机 / 不做大众市场 / 不强调 AI 标签	持续遵守

✅ 确定（架构决策）

项	决策依据
Web only v1.0	用户已确认（vs Mac client）
Cursor 模式定位	用户已确认（vs Lovable / 一键生成）
Soft Autopilot UX 原则	用户已确认
简单 + 呼吸感视觉	用户已确认
Volcengine 全走直连不走 OR	用户已确认
v1.0 不含 Audio / Video / 协作	用户已确认
v1.0 中国为主国际预留	用户已确认
Cascade mark stale + 用户主动	用户已确认
Refinement 默认开 + 完整跳过	用户已确认
AI 透明度 Cursor 模式	用户已确认
graph view v1.5+ 必需	用户已确认（长片场景）
narrative 字段 v1.0 optional / v1.5+ required	用户提出

13.5.2 仍是假设（待验证）

以下未实测，是**基于设计推断的假设**，v1.0 灰度后用真实数据校准：

项	假设	验证方式
Cross-batch 一致性在 50+ batches 长片仍稳	14 batches 实测 OK，假设 50+ 也稳	v1.5 长片 PoC
Cascade Preview UI 用户实测流畅	设计层流畅，未做用户测试	v1.0 灰度访谈
Refinement Plan LLM 意图识别准确率 ≥85%	几个 case 实测看似 OK，未规模化	100+ case 测试集
Asset 自动 extract from NL 准确率	LLM 推断应该 OK，未实测	50+ 剧本 test set
Multi-character panel cross-stage verify	PoC v3 是单角色，多角色未跑	v1.5 PoC
Job queue throughput 1K active users	设计层够，未压测	上线前压测
DB schema 在 10K 项目下 latency	索引设计了，未真跑大数据量	灰度后监控
国际市场拓展工程量	估计 1-2 月，未细化	v1.5 启动时再估
ICP 备案时间线（中国）	6-8 周，依赖审批	v1.0 上线前 8 周启动
Soft Autopilot 不同用户体感	设计原则，未实测	用户访谈
Free → Pro 转化 8%	行业经验值，未本产品验证	(商业化阶段)
Verify 阈值（PASS ≥8 / RETRY 6-7 / FAIL <6）合理	PoC 8 case 推断	100+ verify 数据校准
用户重生成率 < 30%	估计值	实际数据看
单条成本 ¥18-25	推算	实测累积

13.5.3 v1.0 灰度阶段必须验证的优先级

```yaml
priority_1_critical:
  - Cross-batch 一致性大项目稳定性 (50+ batches)
  - Refinement 意图识别准确率
  - Cascade Preview 用户体感
  - 端到端首次成功率（实际 vs 假设 85%）

priority_2_important:
  - Multi-character verify
  - DB latency 真实数据
  - Job queue 压测

priority_3_后续:
  - 商业化 metrics
  - 国际市场需求
  - 长片产品线 fit
```

13.5.4 数据闭环计划

```
v1.0 灰度 100 用户 →
  收集 verify 失败 case (1000+) →
    每周分析 failure pattern →
      调整 prompt / 阈值 / 路由 →
        验证集 A/B 测 →
          生产 deploy
```

这是 PRD 不是一锤子的设计文档，**v1.0 灰度后会持续基于数据迭代**。

13.7 现有代码盘点（v1.1 修正：原估 10–15%，实际 25–35%）

PoC 阶段已积累 **~8000 行有效代码 + 19 preset + 8 导演 v2 数据**。

13.7.1 R1 SVG Preview Engine (~2500 行)

```
lab/r1-svg-preview/
  ├─ primitives.js   (1994 行) — 完整 SVG 图元库
  ├─ narrative.js    (275 行)  — 叙事结构渲染
  ├─ render.js       (237 行)  — SVG 渲染引擎
  └─ index/gallery/test HTML — 浏览页

价值: SVG 渲染基础设施（v1.0 可能不主用，但 v1.5+ 模板/缩略图阶段可复用）
```

13.7.2 R2 PoC D Animatic (~3500 行)

```
lab/r2-poc-d/animatic/
  Players (14 版本):
    ├─ player_v62_anderson.html (548 行) ★ 完整产品级
    ├─ player_p1.html / p1_noir.html (5min alice 故事)
    ├─ player_p3.html (调音师 noir thriller)
    ├─ player_v6/v61/v62.html (主迭代)
    └─ 其他历史版本

    每个 player 已含:
      ✓ 时间轴 progress bar + shot 边界标记
      ✓ 镜头 8 字段详情 panel (景别/镜头/运镜/光线/意图/对白/声音/转场)
      ✓ 播放 / 暂停 / 重播 / 上一镜头 / 下一镜头 / 速度调节
      ✓ 当前 frame 描述显示

  Generation (12 版本):
    ├─ gen_animatic_v62.py / v62_anderson.py — 主管线
    ├─ gen_animatic_p1.py / p3.py — 不同剧本
    ├─ gen_animatic_v6/v61.py — debug v6 灾难的迭代历史
    └─ 9-grid storyboard 生成完整管线

  Tooling:
    ├─ shots_meta.py        — DSL meta 数据结构
    ├─ convert_to_webp.py   — WebP + LQIP 压缩 (11.4×)
    ├─ update_all_manifests.py — manifest 系统
    └─ quick_compare.py     — A/B 对比工具

价值: Player 直接可用作 v1.0 storyboard tab 基础；生成管线产品化即用
```

13.7.3 R3 Product (~2000 行)

```
lab/r3-product/
  utils/
    ├─ or_models.py          — OpenRouter 模型选择器（避免 ID 错误）
    └─ volcengine_client.py  — Volcengine ARK 直连客户端

  nlp_compiler/
    ├─ nlp_compiler.py       — NL → DSL 核心
    ├─ bench_v2.py           — 13 模型 × 3 脚本评估管线（含 retry）
    └─ bench_volcengine.py   — Volcengine 直连 bench

  verify/
    └─ vlm_verify_poc.py     — VLM 三层路由（374 行）

  exp_a_p1_p3style/          — Anderson 风格生成（含 Pro 版）
  exp_b_imagegen_bench/      — 8 模型图像生成 bench
  exp_c_asset_consistency_poc/  — Asset PoC v1/v2/v3

  style_presets/
    ├─ 19 个 yaml (style 6 / genre 6 / director 7)
    ├─ calibrated_data/v1/   — 8 导演 v1 medium
    └─ calibrated_data/v2/   — 8 导演 v2 high (5/8)

价值: 全部直接产品化 wrap 即可上层调用
```

13.7.4 完整度评估

模块	已有	需开发
Player UI（storyboard tab）	~70% (player_v62_anderson 完整)	接 React + 多 batch viewer
AI 调用层	~80% (Volcengine + OR + retry + bench)	包成 service 接口
Storyboard 生成管线	~80% (gen_animatic_v62)	队列化 + 错误处理
NL Compiler	~60% (核心 logic)	加 Refinement Plan 层
VLM Verify	~60% (三层路由代码)	加 Continuous Mode + 反压
Asset Pipeline	~30% (PoC v3 验证)	4-Phase 产品化 + UI
Style Preset 库	~80% (19 yaml + 8 导演)	接 yaml loader
Frontend 应用框架	0%	Next.js + 5 tab + sidebar
Backend API	0%	FastAPI + 30+ endpoints
Editor 主体（NL/Fountain/DSL）	0%	三层 tab + Visual mode
Asset Library UI	0%	List + Graph + 详情

模块	已有	需开发
Cascade Preview UX	0%	Form A/B/C
NL Chat panel	0%	always-on + plan card
Refinement Plan UX	0%	document-style plan
Onboarding + Templates	0%	12+19+6 内容 + 流程
Auth / DB / Storage / Job Queue	0%	部署 + migration
法律 / 备案 / 内容审核	0%	启动 + 等审批

整体完成度: **25–35%**（PoC 阶段沉淀比想象多）

13.7.5 修正后的开发时间估算

团队配置	灰度版 (Closed Alpha 100 用户)	正式版 (Public Launch)
Solo + Claude Code	**2.5–3 个月**	**9–11 个月**
小团队 (2 eng + 1 designer + 0.5 PM)	**1.5–2 个月**	**5–7 个月**
快速团队 (3 eng + 1 designer + 1 PM)	**1–1.5 个月**	**3.5–5 个月**

比 v1.0 原估**短 2–3 个月**（PoC 资产估值修正）。

13.7.6 关键路径 (Critical Path)

```
month 1: 后端骨架 (FastAPI + Auth + DB) + 设计 mockup 同步
month 2: AI 调用层产品化 (Wrap PoC code) + Frontend 框架
month 3: Editor 三层主体 + Asset Library
month 4: Cascade + Refinement + Verify Continuous Mode
month 5: Onboarding + Templates + Polish
month 6: 灰度 100 用户 + 数据校准
month 7–9: 备案审批 + 正式版 polish (并行)
```

⚠️ **AI 备案 (3–6 月) 必须 month 1 启动，不能压到 month 6。**

---

14. 路标 v1.0 / v1.5 / v1.6 / v1.7 / v2.0（v1.1 拆分）

v1.0 (Q3 2026 目标 — 灰度 100 用户)

- ✅ 5 章核心 (定位 / IA / 模型 / Verify / Cascade)
- ✅ Asset Pipeline (4-Phase + Tier + Self-verify)
- ✅ Conversational Compiler (5 层 Refinement)
- ✅ 3 编辑器 (NL / Fountain / DSL)
- ✅ Storyboard 生成 + Verify (Flash 主力 / Pro high-quality 模式)
- ✅ Cost meter + 配额管理（**v1.0 无收费**，跑通产品再考虑商业化）
- ✅ DSL hierarchy (Project → Scene → Shot, narrative 字段 v1.0 optional)
- ❌ Audio / Video gen (v1.5)
- ❌ 多人协作 (v1.6)
- ❌ 长片 / 电视剧 (v1.7)

v1.5 (Q4 2026 – Q1 2027): Audio + Video

- Audio Pipeline (Doubao TTS)
- Animatic Video (Seedance 1.0/2.0 Pro)
- Mobile Viewer
- 国际市场拓展（双地区 infra）

v1.6 (Q2 2027): 协作 + Marketplace 内测

- 多人协作 (CRDT 实时同步)
- Asset Bank 团队 / 公开
- Templates Marketplace 内测
- Custom Preset 高阶用户开放

v1.7 (Q3 2027): 长片 / 电视剧产品线

- Knowledge Graph view（Asset Library）
- Continuity AI 跨 scene 检测
- Story Arc Dashboard
- DSL hierarchy 升级 Sequence + Act

v2.0 (2028 H1): Public API + 商业化全栈

- Reverse Compile（视频 → DSL）
- Public API + Plugin 系统
- 自动 prompt 改进闭环（基于 verify 数据）
- 商业化全套（plan / billing / 团队席位）
- 私有部署（Enterprise）

⚠️ **v1.5 原版塞 6 大块不可能（coherence review 指出）**，拆 4 季度滚动 v1.5/v1.6/v1.7 更务实。

---

15. 测试用例汇总

总数 ~80 个 unit test + ~30 个 perf test，分布如下：

章节	Unit	Perf	关键 case
1 定位	5	0	入门→专业过渡
2 IA	5	4	LCP <1.5s
3 模型	6	2	provider failover
4 Verify	6	4	inline <100ms
5 Cascade	8	4	debounce 5s
6 Asset	8	4	三视图同一性
7 Compiler	6	3	refinement <8s
8 Prompt	6	3	template render <50ms
9 Editor	7	5	tab 切换 <100ms
10 UX	6	4	60fps 动画
11 Ops	6	4	onboarding 5min
11.5 Infra	5	4	DB <100ms p99

Integration test 关键路径（10 个）： – E2E.1: 入门用户 NL → 完整 storyboard 5 min – E2E.2: 半专业改 DSL.dur → cascade → 重渲 → 满意 – E2E.3: 改 alice → 5 batches stale → cascade preview → apply – E2E.4: 切 Style Preset → 全片重渲 + dashboard – E2E.5: NL chat “改 panel 5” → 反向意图 → 询问 → DSL 改 → 重渲 – E2E.6: Asset 上传 ref → emb → cascade – E2E.7: Verify Tier 1 → RETRY → Tier 2 dispute → PASS – E2E.8: Backup → 删项目 → restore 24h 内 – E2E.9: 限流 80% → notification → 升级 plan – E2E.10: Sovereign mode → 强制 Volcengine → 数据不出境

Performance test 全管线: – 60s 短片端到端 < 5 min p90 – 200 并发用户登录: < 2s p99 – DB 5K 项目用户 list: <100ms

附录 A: Bench 数据（关键节选）

A.1 Compiler Bench v2 (13 模型 × 3 脚本)

模型	$/run	Aesth	Pace	T(s)	评级
claude-sonnet-4.6	$0.077	8.0	7.2	77	⭐ 主力
claude-opus-4.7	$0.148	8.0	7.7	48	premium
deepseek-v4-flash	$0.0013	6.8	6.0	100	⭐ 量产
deepseek-v4-pro	$0.024	7.3	7.0	175	mid
minimax-m2.5	$0.005	6.7	5.8	42	最快
haiku-4.5	$0.020	6.5	5.8	25	兜底
seed-2-0-pro (volcengine)	¥3.2/M	TBD	–	127	国产

详见 BENCH_V2_VLM_ANALYSIS.md

A.2 VLM Verify Bench (8 模型 × 3 grid)

模型	$/call	Detection	T(s)
opus-4.7	$0.029	9/9	13
sonnet-4.6	$0.020	9/9	26
gpt-5.1	$0.019	8/9	11
doubao-1-5-vision-pro	$0.002	7/9	8
qwen3-vl-235b	$0.0009	7/9	23

A.3 Image Gen Bench (8 模型 × 2 prompt) — v1.1 主力修正

模型	3x3 grid	角色	速度	成本	评级
gemini-3.1-flash-image-preview	✅	⭐⭐⭐⭐	23s	¥0.0025/grid	⭐ v1.1 主力 (per D-034)

模型	3x3 grid	角色	速度	成本	评级
gemini-3-pro-image-preview	✅	⭐⭐⭐⭐	36s	¥0.01/grid	high-quality 模式 (用户付费)
doubao-seedream-5.0	✅	⭐⭐⭐⭐⭐	42s	~¥0.10/grid	国产备援
doubao-seedream-4.0	❌ 6 panel	⭐⭐⭐⭐⭐	16s	–	单图 hero only
gpt-5-image / 5.4-image-2	慢/失败	–	77-186s	–	❌ blacklist
seedream-4.5	metadata leak	–	–	–	☠️ blacklist

v1.1 关键修正：v1.0 原版主力错列 Pro，应该是 **Flash**（per Round-00 D-034）。理由： – Flash 比 Pro 快 1.6× / 便宜 4× – v62 Anderson 14 batches 实测 Flash 跨 batch alice 一致性已足够 – Pro 留给 high-quality 模式（用户付费 plan unlock）

A.4 Video Gen Smoke Test

- ✅ doubao-seedance-1-0-pro: 50s 出 1080p 16:9 5s mp4
- ✅ doubao-seedance-2-0: 最新版可用

A.5 Asset Pipeline PoC

```
PoC v1: 整图 emb → 0.57 cross-batch sim, 不能区分身份维度
PoC v2: composite ref injection → 反而干扰, 0.51-0.60
PoC v3: VLM Opus 4.7 直接判断 → 7.7 cross-batch / 0 negative ★ LOCK-IN
PoC v3 升级 prompt → 完美 (DIFFERENT_TYPE 完全识别 negatives)
```

附录 B: 8 导演校准数据 (v2)

导演	ASL (s)	Confidence	主要数据点
Wes Anderson	6.5 (avg 5 films)	high	Cinemetrics lab.php 5 部精确
Christopher Nolan	3.2	high	Inception 3.1s + Hoytema 4 源
David Fincher	3.87	high	Gone Girl 2400 镜头 + Cronenweth ASC
Miyazaki	4.8 (3 films)	high	Spirited Away 5.1s 直引
Wong Kar-wai	5 (双簇)	medium-high	Doyle 4 源
Kurosawa	6.5	medium-high	Donald Richie + Shot-Level Dataset
姜文	2.5	medium	让子弹飞 1.33s 鸿门宴段
A24 派	1.2-9 双簇	medium-high	EEAAO 修正 1.5s + Aster/Eggers 慢

详见 `style_presets/calibrated_data/v2/*.md`

附录 C: Database Schema

完整 17 张表 schema 详见 `THINKING_DATABASE_DESIGN.md`。核心结构：

```
Core: users / projects / project_versions
Asset: assets / asset_views / asset_relations / asset_shot_references / asset_audit_log
Generation: storyboard_batches / audio_clips(v1.5) / video_clips(v1.5)
AI/Verify: ai_calls / verify_results / verify_failures / cascade_history
Billing: subscriptions / usage / abuse_log
Web: sessions / chat_messages / notifications / file_uploads
Static: templates / style_presets
Job: jobs (with Celery integration)
```

设计原则：Graph-native + Soft delete + Audit trail + JSONB for semi-structured + pgvector for embeddings.

附录 D: API Endpoint Spec (30+ endpoints)

```
=== Auth ===
POST   /api/v1/auth/sign-up
POST   /api/v1/auth/sign-in
POST   /api/v1/auth/wechat-callback
POST   /api/v1/auth/sms-code
POST   /api/v1/auth/sign-out
GET    /api/v1/auth/me

=== Projects ===
GET    /api/v1/projects
POST   /api/v1/projects
GET    /api/v1/projects/{id}
PATCH  /api/v1/projects/{id}
DELETE /api/v1/projects/{id}
POST   /api/v1/projects/{id}/duplicate
POST   /api/v1/projects/{id}/snapshot
POST   /api/v1/projects/{id}/restore/{version_id}

=== Compile / Cascade ===
POST   /api/v1/projects/{id}/compile-nl
POST   /api/v1/projects/{id}/refine-confirm
PATCH  /api/v1/projects/{id}/dsl
POST   /api/v1/projects/{id}/cascade-preview

=== Assets ===
GET    /api/v1/projects/{id}/assets
POST   /api/v1/projects/{id}/assets
PATCH  /api/v1/projects/{id}/assets/{asset_id}
DELETE /api/v1/projects/{id}/assets/{asset_id}
POST   /api/v1/projects/{id}/assets/{asset_id}/regenerate
POST   /api/v1/projects/{id}/assets/{asset_id}/upload-ref

=== Storyboard / Generation ===
POST   /api/v1/projects/{id}/generate-storyboard
GET    /api/v1/projects/{id}/storyboard/{batch_index}
POST   /api/v1/projects/{id}/regenerate-batch

=== Verify ===
GET    /api/v1/projects/{id}/verify-status
POST   /api/v1/projects/{id}/verify-run

=== Jobs ===
GET    /api/v1/jobs/{job_id}
POST   /api/v1/jobs/{job_id}/cancel
SSE    /api/v1/jobs/stream

=== NL Chat ===
POST   /api/v1/projects/{id}/chat
GET    /api/v1/projects/{id}/chat-history
```

```
=== Export ===
POST   /api/v1/projects/{id}/export

=== Billing (v1.0+ 商业化暂不开启，路标 v2.0) ===
GET    /api/v1/billing/plan
GET    /api/v1/billing/usage
POST   /api/v1/billing/upgrade
POST   /api/v1/billing/cancel-subscription

=== ⚠️ v1.1 补全（架构 review 指出阻塞）===

# WebSocket / SSE channel
WS     /api/v1/projects/{id}/stream
  Events:
    - job.started / job.progress / job.completed / job.failed
    - cascade.preview_ready / cascade.applied / cascade.batch_done
    - verify.warning / verify.failed
    - asset.regenerated / asset.stale
  Auth: token via query param 或 first message
  Heartbeat: 30s ping/pong
  Reconnect strategy: exponential backoff

# Presigned URL (用户上传 ref image)
POST   /api/v1/uploads/presign
  request: {filename, mime_type, size_bytes, purpose}
  response: {upload_url, fields, expires_at, file_id}
  follow-up: PUT 直传到 R2/OSS via presigned URL

# Quota Header (每 API response)
X-Quota-Limit: 50 (project/month)
X-Quota-Used: 23
X-Quota-Reset: 2026-05-01T00:00:00Z
X-Cost-Today-CNY: 12.50
X-Cost-Cap-CNY: 30.00 (per user setting)

# Cascade Preview Schema 详细
POST   /api/v1/projects/{id}/cascade-preview
  request: {
    trigger: "asset_change" | "dsl_change" | "preset_change",
    target_id: UUID,
    diff: {field, old_value, new_value}
  }
  response: {
    affected: [
      {
        type: "storyboard_batch" | "audio_clip" | "asset",
        id: UUID,
        impact_level: "regenerate" | "verify_only" | "no_impact",
        cost_estimate_cents: int,
        time_estimate_sec: int
      }
    ],
    total_cost_cents: int,
    total_time_sec: int,
    locked_provider: string (cascade 启动时锁定的 provider)
  }
```

格式：JSON request/response，错误格式 `{error: {code, message, details}}`，时间戳 ISO 8601 UTC，ID UUID v4。

文档版本

- v1.0 Draft: 2026-04-28
- 待 review: 多 agent 审查 + 用户 review
- 下一版: v1.1 (review 反馈合并) → v1.2 (设计/工程同步)

引用文档（思考清单 TDOC, 16 份）

```
TDOC 列表:
1.  MODEL_ROUTING_v1.md
2.  BENCH_V2_VLM_ANALYSIS.md
3.  BYTEDANCE_SCENARIO_FIT.md
4.  VERIFY_HARNESS_v1.md
5.  THINKING_PROMPT_COMPOSITION.md
6.  THINKING_UNIFIED_EDITORS.md
7.  THINKING_CASCADE_EDITS.md
8.  THINKING_PRODUCT_POSITIONING.md
9.  THINKING_CONVERSATIONAL_COMPILER.md
10. THINKING_DESIGN_BRIEF.md
11. THINKING_HARNESS_HOLISTIC.md
12. THINKING_ASSET_PIPELINE.md
13. THINKING_ASSET_INNOVATIONS.md
14. THINKING_PLATFORM_INFRA.md
15. THINKING_V1_OPS_STACK.md
16. THINKING_DATABASE_DESIGN.md
17. THINKING_AI_PROGRESS_UX.md
```

— END —