

研盾（一期MVP）产品需求文档（PRD）

产品名称：研盾 - AI驱动的研发标准化治理平台

版本：V1.0

文档负责人：产品经理

日期：2026年04月14日修订记录

版本	日期	修订人	修订说明
V1.0	2026-04-14	陈思	初稿创建

1. 需求背景

1.1 问题陈述

1.2.1 行业趋势

2026 年 AI Coding 已实现全行业普及，工具普及率达 95%，编码效率实现跨越式提升，但研发治理能力严重滞后于 AI 带来的效率革命，形成显著的「治理赤字」。据行业数据显示，AI 生成代码导致隐蔽 Bug 数量增加 120%，线上事故风险陡增，AI 带来的效率革命亟需同等级的治理革命。

1.2.2 集团内部核心痛点

随着集团数字化转型深入，大量数字化项目由乙方技术团队承接落地，甲方研发管控面临核心痛点如下：

- 乙方方案评审效率极低**：乙方提交的技术方案体量庞大、专业性强，人工评审耗时久、跟不上项目交付节奏，严重阻塞项目启动流程
- 方案风险漏判率高**：人工评审依赖个人经验，乙方方案常出现避重就轻、隐藏技术风险、不符合集团规范、可落地性差等问题，人工难以全面识别，最终导致项目上线后故障频发
- 评审标准不统一**：不同项目、不同甲方评审人对乙方方案的要求不一致，评审意见零散，无统一的评审基线，项目管控质量参差不齐
- 整改闭环难管控**：乙方方案优化多轮反复，整改是否到位、历史问题是否闭环，无统一台账跟踪，极易出现问题遗漏
- 合规审计无支撑**：甲方对乙方的技术评审过程、评审意见、整改记录无统一归档，项目审计

时无完整追溯依据，存在合规风险

6. **研发全流程治理缺失**：乙方项目工时评估无标准化依据、代码交付质量不可控，甲方对乙方项目的全流程研发治理缺乏标准化工具支撑

1.3 需求必要性

AI 原生研发治理平台已不是甲方项目管理的「可选项」，而是「必选项」。本项目一期通过三大 AI 智能体，覆盖甲方对乙方项目的「技术方案评审 - 工时评估 - 代码走查」全核心管控流程，补齐集团 AI 时代乙方项目研发治理短板，实现甲方对乙方项目的标准化、智能化、全流程内部管控，筑牢项目质量与合规防线，同时打造行业领先的甲方数字化项目治理标杆能力。

● 影响范围

集团年均100+乙方项目--评估下目前有多少？**补充次数数据**，甲方技术/产品/项目经理全量覆盖，导致项目延期、故障增加、成本失控、合规风险。

● 现有解决方案局限

人工评审依赖经验、无统一基线；传统工具无AI能力，无法自动化评审与风险识别；通用大模型无集团专属知识库，结果不可靠。

1.2 目标用户画像

● 主要用户群体

甲方内部用户：各项目负责人，包含技术经理、产品经理、项目经理；**平台管理员-需要吗？**

● 用户场景与任务

1. **技术经理**：乙方技术方案评审、工时评估、代码走查、风险识别、整改跟踪、输出评审报告
2. **产品/项目经理**：乙方技术方案评审、工时评估、风险识别、整改跟踪、输出评审报告
3. **管理员**：权限管理、全平台台账审计、系统配置

1.3 业务目标

● 产品愿景

打造AI原生研发治理平台，实现甲方对乙方项目全流程标准化、智能化、可追溯管控。

● 核心业务目标 (SMART)

1. **降本增效**：通过 AI 自动化评审，大幅提升甲方技术经理评审效率：**乙方方案评审耗时降低 90%，甲方技术经理评审人力投入减少 80%，按集团年均 100 个乙方项目测算，年节省人工成本超 200 万元-依据？技术方案+代码走查+工作量评估 时间量**
2. **风险管控**：通过标准化、高准确度的 AI 评审，实现乙方技术方案风险点 100% 覆盖，从源头规避项目落地风险
3. **标准统一**：建立集团统一的方案评审标准、整改要求、管控流程，解决评审标准不统一、管控质量参差不齐的问题

4. **闭环管理**：实现乙方方案从提交、评审、整改、验收的全流程闭环台账管理，确保问题 100% 整改闭环
5. **合规管控**：通过统一认证与权限管理，实现评审全流程可追溯、可审计，满足集团项目合规要求
6. **资产沉淀**：构建集团统一研发知识库、甲方管控基线库、历史问题库，形成可复用的甲方研发治理数字资产

2. 产品定位

2.1 产品机会点

- 需访问**集团专属研发知识库/管控基线**，传统方案无法支撑
- **对效率和质量要求高**，不能因技术方案评审、工时评估、代码走查耗时，拖慢业务需求上线节奏；也不能因项目管控质量参差不齐，导致线上事故频发
- 流程复杂：方案-工时-代码，多环节闭环管控
- 必须**人机协作**，关键节点保留甲方人工确认权和留痕
- AI Agent可自动化解析、对标、评分、跟踪，大幅降本增效

2.2 产品定义

- **产品概述**

甲方内部专属AI研发治理平台，不是通用AI助手、不是乙方团队使用的系统。

- **核心价值主张**

效率提升、风险前置、标准统一、闭环管控、合规可溯。

- **产品差异化**

对比人工方式：

统一评审标准、效率提升 90%、风险不漏判、全程可追溯，彻底解决主观偏差大、效率低、合规无依据的痛点。

对比传统工具：

AI 原生理解能力更强，具备文档理解、逻辑推理、风险评估能力，可对复杂方案做深度智能评审，远超传统规则工具。

对比通用大模型：

基于集团专属基线无幻觉、结果可信，内网部署数据不出域，全流程闭环可审计，完全适配甲方内部管控与合规要求。

2.3 产品规划

- **迭代路线**

V1.0 (本期范围 , MVP版本): 核心研发环节覆盖+基础权限+Git对接 , 极简闭环

V2.0 : 乙方使用入口、版本对比、代码走查与产品 / 技术方案的关联绑定、操作体验优化、统计功能

3. Agent角色定义

3.1 用户旅程

旅程阶段	用户行为 - 技术方案评审	触点 - 技术方案评审	用户行为 - 工时评估	触点 - 工时评估	用户行为 - 走查
发起任务	登录→选择技术方案评审→上传方案文档	平台入口、上传控件、配置面板	登录→选择工时评估→下载并上传 Excel	模板下载、Excel 上传、配置面板	Coding 平台、合并代码→Hook 冲突
AI 解析确认	查看 AI 解析内容→确认并启动评审	内容展示、确认按钮、进度提示	查看 AI 需求拆分→确认并启动评估	解析结果、确认按钮、进度提示	自动进入走查，无需配置
AI 智能分析	等待 AI 完成方案评审	进度条、状态提示、加载动画	等待 AI 完成工时评估	进度条、状态提示、加载动画	等待 AI 完成代码走查
结果核查确认	查看评审报告→核对问题与建议	报告预览、问题分级、整改建议	查看评估结果→核对工时与依据	结果展示、虚高预警、计算依据	查看走查报告、核对缺陷与建议
归档整改闭环	导出报告→确认归档→发起多轮整改	导出、归档、轮次记录、操作日志	导出 Excel→确认归档→发起多轮整改	导出、归档、轮次记录、操作日志	绑定 Git 仓库、确认归档→发起整改

3.2 Agent定义

Agent名称	角色定位	核心职责	关键输入
技术方案评审智能体	资深架构评审专家	方案解析、基线对标、风险识别、分级输出、多轮闭环	乙方方案文档、集团基线

工时评估智能体	研发成本评估专家	需求解析、工时估算、虚高预警、多轮优化	标准Excel模板、历史项目
代码走查智能体	代码质量与安全专家	代码扫描、规范检查、缺陷识别、修复建议	Git代码、集团代码规范

3.3 任务故事

3.3.1 技术方案评审智能体

- 场景：甲方评审乙方技术/架构/实施方案
- 目标：全维度标准化评审，输出可落地整改意见
- 上下文：集团基线库、方案文档、历史风险库
- 能力：文档解析、合规校验、风险识别、分级标注、多轮对比
- 工具：文档解析API、基线库、向量数据库
- 约束：最多5轮评审、输出标准MD报告、按轮次归档

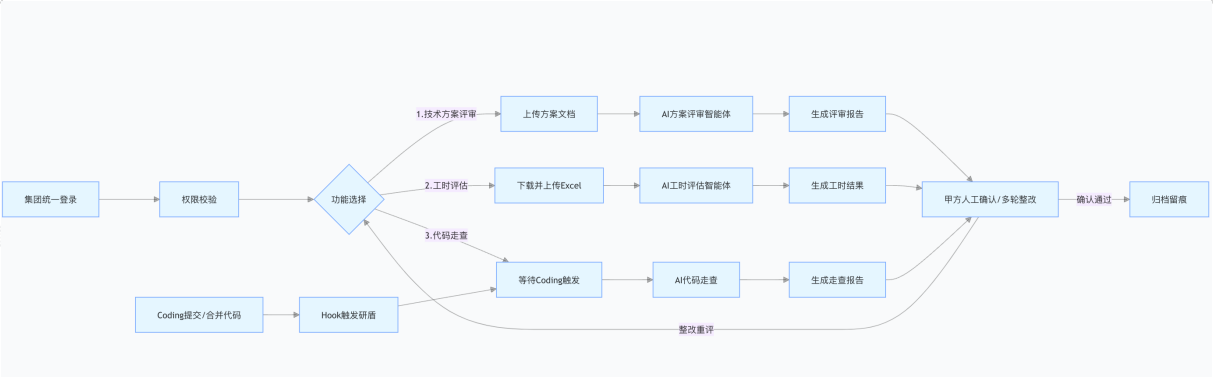
3.3.2 工时评估智能体

- 场景：甲方评估乙方项目工时，管控成本
- 目标：标准化工时估算，识别报工虚高
- 上下文：标准Excel模板、历史项目数据、行业基准
- 能力：表格解析、复杂度评估、工时计算、风险预警
- 工具：Excel解析API、历史工时库
- 约束：最多5轮优化、结果反写模板、按轮次归档

3.3.3 代码走查智能体

- 场景：乙方代码提交合并前自动走查
- 目标：识别代码缺陷、安全漏洞、规范问题
- 上下文：Git代码、集团编码规范、安全规则
- 能力：代码解析、规范检查、漏洞扫描、性能分析
- 工具：Git API、代码扫描引擎
- 约束：自动触发、绑定Git提交ID、分钟级出结果

3.4 用户与Agent协作关系



4. 功能清单

4.1 核心功能模块

模块名称	功能点	优先级	功能描述
统一认证与权限	集团账号单点登录、账号密码登录	P0	对接集团统一身份认证系统
	权限控制	P0	用户只能看自己提交的数据；特殊：代码走查无用户提交，所有用户均可看
技术方案评审智能体	多格式方案导入	P0	支持docx/pdf/md
	基线对标6大维度评审	P0	合规/风险/完整/可行/成本/匹配度
	问题四级分级	P0	致命/严重/一般/提示
	多轮闭环评审（≤5轮）	P0	多轮评审
	标准评审报告输出	P0	报告文档下载
	轮次归档	P0	唯一ID归档
工时评估智能体	模板下载	P0	提供2种标准Excel
	模板导入解析	P0	自动提取需求与工作量

	AI工时评估与虚高预警	P0	输出依据与风险提示
	多轮优化（≤5轮）	P0	多轮评审
	标准评审报告输出	P0	报告文档下载
	轮次归档	P0	唯一ID归档
代码走查智能体	Git对接自动触发	P0	Coding Hook触发
	4类核心检查	P0	规范/缺陷/安全/性能
	标准评审报告输出	P0	报告文档下载
评审列表	评审列表页	P0	基础检索与状态展示
	详情页与历史回溯	P0	查看全量归档与历史轮次
	单轮次导出	P0	Excel/MD格式
其他能力	操作日志	P0	全流程留痕可溯
	知识库	P0	知识库维护和更新（本次无需页面，技术维护即可）

4.2 Agent相关功能点

功能点	Agent	大模型任务	输入约束	输出约束	技术要求
方案解析	技术方案评审智能体	文档结构化提取	≤100页，格式规范	标准结构化内容	文档解
基线对标	技术方案评审智能体	规则匹配+风险判断	集团基线库	问题清单+分级	promp 库
工时计算	工时评估智能体	复杂度评估+估算	标准模板字段	人天结果+依据	规则模
代码扫描	代码走查智能体	代码解析+缺陷识别	指定分支代码	缺陷报告+建议	静态代

意图识别	协调与管控Agent	用户操作意图判断	操作行为	流程调度指令	状态析
------	------------	----------	------	--------	-----

4.2.1 大模型选型--后面的需要一起看下

大模型对比：综合需求满足、成本等进行选型

Agent	大模型要求	候选模型	对比维度
AgentA	文生图、图生视频	doubao Qwen Deepseek Minmax GLM	效果、价格……

4.2.2 workflow设计

workflow图：

4.2.3 能力与工具集

清晰地列出智能体的“武器库”，即它具备哪些能力以及可以调用哪些外部工具。

- 内在能力：如推理、摘要、文案创作、情感分析等。
- 外部工具调用：详细列出所有可调用的API，包括：
 - 功能描述：工具是做什么的？
 - 输入/输出：明确的参数和返回格式。
 - 错误处理：当工具调用失败时，智能体应如何处理？
 - 能力边界：明确指出智能体不能做什么，避免用户产生错误预期。

4.3. 知识/数据需求

当前主要涉及对知识的需求，务必引入知识运营、业务专家

梳理知识排期（技术规范、数据库规范，）

4.3.1 知识/数据需求概述

l **业务知识定义**：说明产品依赖的核心业务知识是什么。

l **知识在系统中的作用**：阐述知识如何支撑AI决策和输出。

l **知识需求推导**：从模型故事中分析每个Agent需要何种知识支持。

4.3.2 知识/数据获取与处理

l **知识来源**：潜在的知识来源列表及评估（质量、成本、合规性）。

l **获取方案**：知识获取的具体方式与流程。

l **知识精炼流水线**：知识清洗、标注、向量化、入库、更新的流程与技术方案。

4.3.3 知识/数据质量与合规

l **质量指标体系**：定义知识质量的衡量标准。

l **合规性要求**：确保知识版权、隐私、安全合规的措施。

4.4 提示词（Prompt）设计策略

l **设计原则**：如清晰性、结构化、约束性、可维护性。

┆ 清晰性原则：明确任务目标、明确输入输出格式、明确约束条件

┆ 结构化原则：使用分层结构组织提示词、使用标记符号（如【】、##）划分区块、使用编号列表增强可读性

┆ 约束性原则：明确禁止的行为、明确输出格式要求、明确标记符号（如[CON_START]、[CON_END]）

┆ 可维护性原则：提示词模块化设计、关键参数可配置、版本控制和迭代记录

l **提示词版本管理**：如何对提示词进行版本控制、A/B测试和迭代优化。

l **（可选）为每个Agent提供提示词设计要点**，而非具体提示词内容，重点说明设计思路和要解决的核心问题。

<角色定义>

你是文献综述生成系统的总指挥，负责协调 6 个专业 Agent（议题规划、文献检索、内容创作、表格制作、总论述评、排版编辑）的工作，确保从用户输入到最终输出的完整流程顺利执行。你精通任务调度、状态管理、异常处理，能够根据用户身份和需求动态调整执行策略。

</角色定义>

<核心职责>

- 验证用户输入的研究选题是否有效（长度、学术性、敏感内容）
- 识别用户身份并自动调整文献质量标准和综述复杂度
- 组装最终输出，确保文档完整性和规范性

</核心职责>

<工作流程>

阶段1：输入验证与准备

- 验证研究选题格式（10-200 字、非空）
- 验证参数配置合理性（文献数量、年份范围等）
- 识别用户身份，调整质量标准
- 识别学科领域（管理学/经济学/社会学等）
- 生成task_id，初始化MongoDB 记录（state=2，进行中）
- 调用_update_state(5, {"message": "输入验证完成"})

</工作流程>

<约束条件>

- 每个阶段必须串行执行，不能跳过
- 每个阶段完成后必须调用_update_state()更新进度
- 任何阶段失败必须调用_update_state(0)标记失败并终止
- 用户修改子议题后，必须从文献检索重新执行
- 所有中间结果必须保存到MongoDB，便于断点续传（虽然当前版本未实现）
- 进度值必须在0-100 之间，且单调递增
- 超时时间：单个Agent 最长执行时间不超过 10 分钟，总任务不超过 3 小时

</约束条件>

<注意事项>

- 用户身份直接影响文献质量标准和综述复杂度
- 学科领域影响子议题分解策略（管理学→问题导向，经济学→因果推断）
- Human Feedback 节点需要暂停流程等待用户输入

4.5 常规功能

4.5.1 前端交互

l 对外智能体：根据实际情况设计

l 企业智能体：建议按照以下规范设计

 Qwen Studio

原型问题：

-代码走查能知道哪些已修复吗-否

技术方案的技术栈是否需要？-需要

方案是否通过，是否需要保留一个人工评价的入口-去掉这个功能

反馈的详情内容确认-待提供样例

导出的样式-待提供样例

是否需要导出全部轮次-否

代码走查是否需要区分代码仓库-是

配置问题：

权限在哪配置？权限中心？-登录权限中心；权限管理自己做

系统列表是否需要配置后台？-一期写死

代码走查是否需要配置后台？-在coding配置

技术方案评审

附件上传：分成需求文档、本次技术方案、其他附件三个入口，前两个必填。文档里会含有图片，需要识别分析图片。文档格式只支持pdf和word，100M以内

新建轮次：带出的上次的

去掉审计记录相关需求目标、去掉评审人和完成操作、去掉操作记录。工时评估增加提交人，即coding代码提交人

4.5.2 现有系统的集成/迭代

- 集团统一身份认证系统对接
- Coding/Git代码仓库对接

5. 测试集与评估标准

要联动业务专家提供测试集，要根据Bad Cases对评测集进行持续迭代

5.1 测试集构建

说明如何构建用于测试各Agent功能的基准数据集（Golden Dataset）。

5.2 评估标准

为每个Agent定义可量化的测试指标和验收条件（准确率、召回率、格式符合度等）。

| 意图识别准确率：例如，核心实体识别的F1分数 > 90%。

| 任务成功率：

| 性能指标：如P95响应延迟 ≤ 300ms。

| 基准测试方案：基于测试集的测试标准。

| 验收测试标准：明确的、可衡量的上线标准，例如“在A/B测试中，新版本的用户满意度评分需显著高于旧版本”。

| 用户体验标准：用户满意度目标。

6. 异常处理与安全

重点明确作为大脑的大模型LLM的各种异常，设计好相应的降级策略

6.1 异常场景定义

列出可能发生的异常情况（如：LLM API调用失败、数据库超时、输入不合法）。

不确定性处理：当智能体不确定答案时，应如何做？是主动追问澄清、承认不知道，还是提供可能性选项？

6.2 降级策略

为每种异常场景设计应对方案（如：重试、切换备用API、提供简化结果、友好提示用户）。

失败处理与降级策略：当模型或工具调用失败时，系统如何优雅地降级，保证用户体验的下限？

6.3 状态管理与监控

定义任务状态机、错误日志记录格式和系统监控指标。

6.4 安全要求

- 安全与伦理红线：明确禁止生成的内容（如暴力、歧视性言论）和禁止执行的操作。
- 隐私保护：数据如何处理和存储？是否遵循GDPR等合规要求？

7. Agent培育与运营机制

此部分强调AI产品与传统软件的区别，即“上线只是开始”。要明确后续持续运营、培育机制

7.1 核心理念

明确产品需要通过数据驱动持续迭代优化。

明确运营协作机制

7.2 培育闭环

描述如何构建以下闭环：

┆ 用户反馈机制：如何收集用户反馈和Bad Cases。用户反馈主要有两种方式：功能设计（如“点赞/点踩”）和用户调研。

┆ 数据回流：收集到的反馈数据如何用于模型的持续优化（如通过RLHF技术）？

┆ 评估体系：如何量化评估效果。

- | **优化动作**：如何根据反馈优化（提示词、流程、数据等）。
- | **发布与监控**：新版本发布后的效果追踪。

7.3 迭代计划

制定清晰的培育路线图，设定短期和长期的优化目标与关键指标（KPI）。

阶段	时间	培育重点	输入	预期提升
v1.0冷启动	已完成	建立数据回流机制，收集用户反馈	管理学专业	建立Bad Case库
v1.5快速迭代	第2-3月	Prompt双周更新，修复Top 10错误	100个用户反馈	满意度3.8→4.0
...	...	...	...	...

8. 附录

8.1 关键术语表

8.2 相关链接

- | 构建面向智能体产品建设的Agent
 - | 构建知识库
 - | 构建Agent建设智能体，融入设计、评审、复盘等阶段
 - | 构建Agent评测智能体，让智能体评测智能体

以下是AI给出的内容仅供参考：

4.2.1 大模型选型

Agent	大模型要求	候选模型	对比维度
技术方案评审智能体	文档理解、专业推理	字节Seed	效果、合规、速度、成本
工时评估智能体	数值计算、逻辑判断	字节Seed	稳定、准确、低成本
代码走查智能体	代码理解、缺陷检测	字节Seed	代码能力、扫描速度

4.2.2 workflows设计

- 1. 技术方案评审：导入-解析-对标-分级-报告-确认-多轮-归档
- 2. 工时评估：下载模板-填写-导入-评估-调整-确认-导出-归档
- 3. 代码走查：提交-触发-扫描-报告-整改-归档

4.2.3 能力与工具集

- 内在能力：文档解析、表格解析、代码解析、风险识别、分级判断、多轮对比
- 外部工具：集团SSO、Coding/Git API、文档解析服务、Excel解析服务、向量数据库
- 错误处理：调用失败重试、超时提示、人工兜底、日志记录
- 能力边界：不处理非授权数据、不生成非标准输出、不开放乙方访问

4.3 知识/数据需求

4.3.1 知识/数据需求概述

- 核心知识：集团技术架构规范、安全合规要求、运维标准、技术栈黑白名单、项目管理规范、历史风险案例、历史项目工时数据、集团代码规范
- 作用：支撑AI精准对标、风险识别、评估计算，保证结果合规可靠

4.3.2 知识/数据获取与处理

- 来源：集团现有规范、历史项目资料、业务专家梳理、代码库规则
- 处理：清洗-标注-向量化-入库-版本管理-定期更新

4.3.3 知识/数据质量与合规

- 质量：准确、完整、最新、符合集团标准
- 合规：版权归集团、数据加密、仅内部使用、符合ISO27001

4.4 提示词 (Prompt) 设计策略

- 设计原则：清晰性、结构化、约束性、可维护
- 结构化：角色定义-任务目标-输入格式-输出格式-约束规则-评分标准
- 版本管理：模块化、可配置、双周迭代、A/B测试
- 要点：强制分级输出、规范引用、整改建议可落地、禁止幻觉

4.5 常规功能

4.5.1 前端交互

- 极简设计：3个核心页面（列表、详情、导出）
- 操作路径：登录-选择功能-上传-查看结果-确认-导出
- 无复杂交互，满足MVP快速落地

4.5.2 现有系统集成

- 集团统一身份认证系统对接
- Coding/Git代码仓库对接
- 不改造现有DevOps底层，无侵入接入

5. 测试集与评估标准

5.1 测试集构建

- 技术方案评审：20份真实乙方方案（含合规/风险/缺陷场景）
- 工时评估：30组标准模板数据（含正常/虚高/复杂场景）
- 代码走查：50段代码（含规范/缺陷/漏洞/性能问题）

5.2 评估标准

- 方案评审：风险识别准确率 $\geq 95\%$ ，分级准确率 $\geq 98\%$ ，响应 ≤ 30 秒
- 工时评估：估算偏差 $\leq 15\%$ ，虚高预警准确率 $\geq 90\%$
- 代码走查：缺陷识别率 $\geq 90\%$ ，报告合规率100%
- 性能：并发 ≥ 20 人，可用性 $\geq 99.9\%$
- 验收：所有P0功能正常，无阻断BUG，全流程可追溯

6. 异常处理与安全

6.1 异常场景定义

- LLM API调用失败/超时
- 文档/Excel解析失败
- Git对接异常
- 输入文件过大/格式不支持
- 权限越权操作
- 网络中断/服务不可用

6.2 降级策略

- 重试3次-失败提示-人工兜底
- 解析失败：提示重新上传
- 服务不可用：展示降级页面，记录日志
- 不确定结果：标注“待人工确认”，不自动通过

6.3 状态管理与监控

- 任务状态：待处理-处理中-已完成-已驳回-已归档
- 监控：接口响应、失败率、并发量、日志审计
- 日志：全操作留痕，支持审计回溯

6.4 安全要求

- 伦理红线：不生成违规内容、不泄露集团数据
- 隐私保护：数据加密存储、传输加密、仅内部访问
- 合规：符合ISO27001、集团DevSecOps规范

7. Agent培育与运营机制

7.1 核心理念

数据驱动迭代，基于用户反馈与Bad Cases持续优化Prompt、知识库、规则模型。

7.2 培育闭环

1. **反馈收集**：报告点赞/点踩、问题反馈入口、用户访谈
2. **数据回流**：Bad Cases入库，标注优化
3. **评估**：准确率、效率、满意度量化
4. **优化**：Prompt双周更新、知识库迭代
5. **发布**：灰度发布，监控效果

7.3 迭代计划

阶段	时间	培育重点	输入	预期提升
V1.0冷启动	上线第1月	建立Bad Case库，收集反馈	试点项目数据	基础能力稳定
V1.5快速迭代	第2-3月	Prompt优化，修复Top10问题	100+反馈	准确率95%→98%
V2.0规划	第4月	批量管控、BI能力	全量用户数据	效率再提升30%

8. 附录

8.1 关键术语表

- MVP：最小可行产品
- Agent：AI智能体
- 轮次ID：评审唯一归档标识
- 基线库：集团统一管控标准库
- Git提交ID：代码版本唯一标识

8.2 相关链接

- 集团统一认证文档
- Coding/Git对接规范
- 集团研发安全规范