PROJECT 03 / AGENT ENGINEERING

AI PR Review
代码审查 Agent Harness

让模型按需探索代码，交付能够进入 GitHub 评审流程的审查建议。

01 / 输入

指定待审变更

仓库、PR、head SHA
审查规则与模型配置

02 / 过程

运行代码审查

准备上下文、执行工具循环
约束资源、收集结构化结果

03 / 输出

进入研发协作

行内评论与审查总结
执行指标、人工判断与反馈

个人工作 · 大仓库缓存、多模型审查、上下文控制与评论位置校验。
代码基线：github-watchers master；Git 提交记录支持直接贡献。

Harness 负责模型执行任务的运行支撑

模型判断风险并选择工具；程序组织执行、限制和交付。

01

任务与代码环境

PR 入队、在途去重
仓库缓存、固定 head、diff

02

按需工具探索

读文件、搜索调用关系
查看 Git 历史并继续分析

03

资源与结束控制

上下文门槛、迭代上限
超时、输出截断、收集阶段

04

结构化结果

评论路径、行号、正文
总结与完成信号

05

GitHub 工作流

校验可评论的 diff 行
宿主发布 COMMENT review

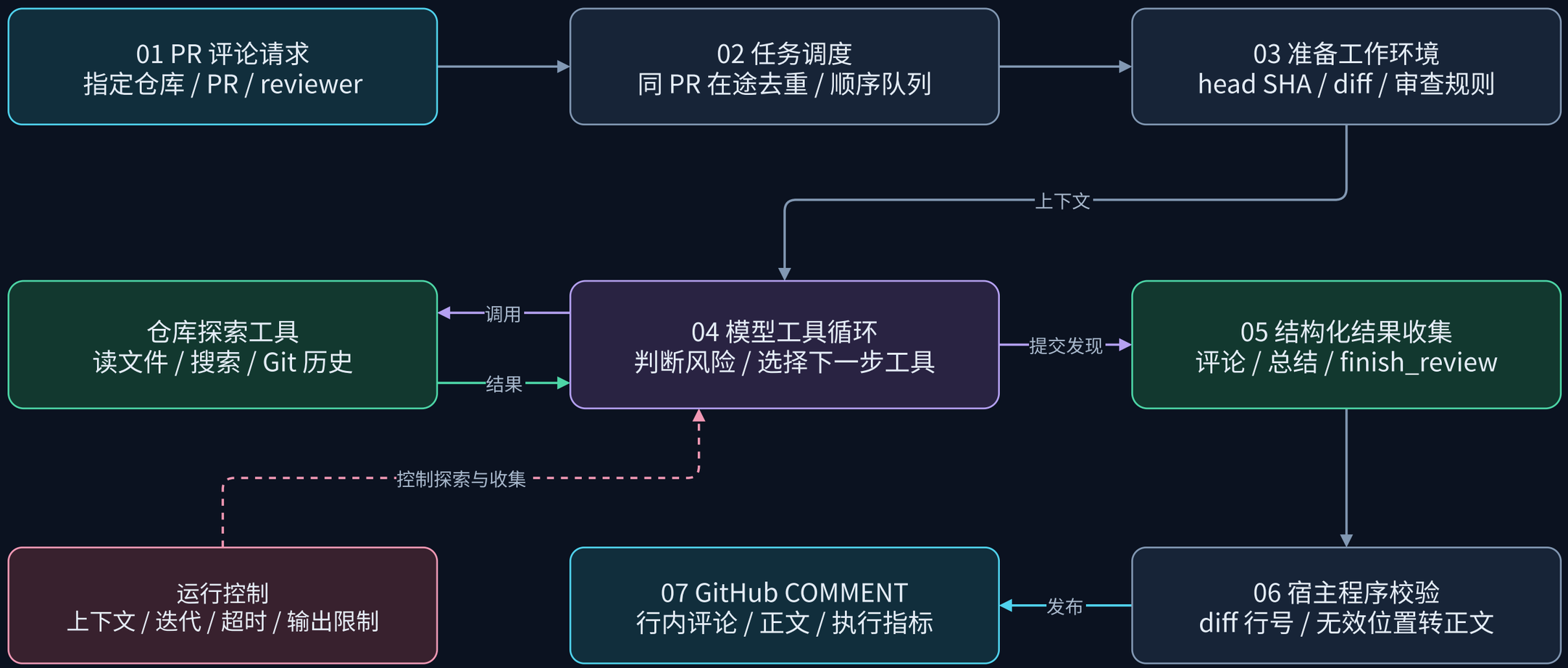
06

适配与可观测性

Anthropic / OpenAI 协议
模型、耗时、token 和错误

AI PR Review：模型分析，Harness 负责运行与交付

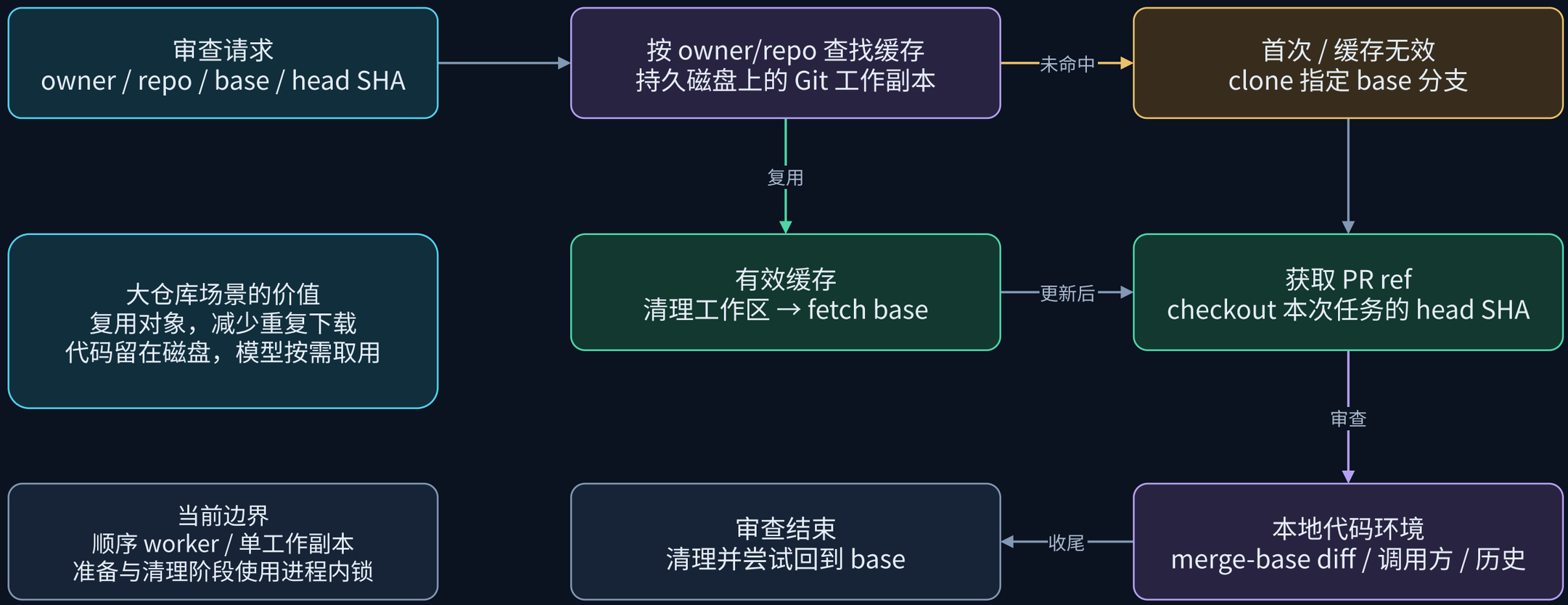
基于 github-watchers master：确定的 PR 工作流中嵌入模型自主选择工具的审查循环。



模型按需调查与提出建议；宿主管理工具、状态和发布契约。人工判断内容正确性与是否合并。

大仓库先缓存到本地，再按 PR 准备审查版本

公司部分仓库体积大；复用已有 Git 对象，减少每次审查都重新 clone 的传输与准备开销。



默认不限制历史深度；single-branch 不等于 shallow。缓存加速代码准备，context window 限制实际送入模型的内容。

一次变更的风险，可能在调用方才显现

以工厂返回值为例：接口本身允许空值，调用方可能默认它永远有效。

变更位置 / 机制示例

工厂函数的返回契约

```
Engine* engine = factory();
// 返回值是否可能为空?
```

仅看新增接口，未必知道下游如何使用。

关联位置 / 机制示例

调用方的隐含假设

```
Engine* engine = getEngine();
engine->init();
```

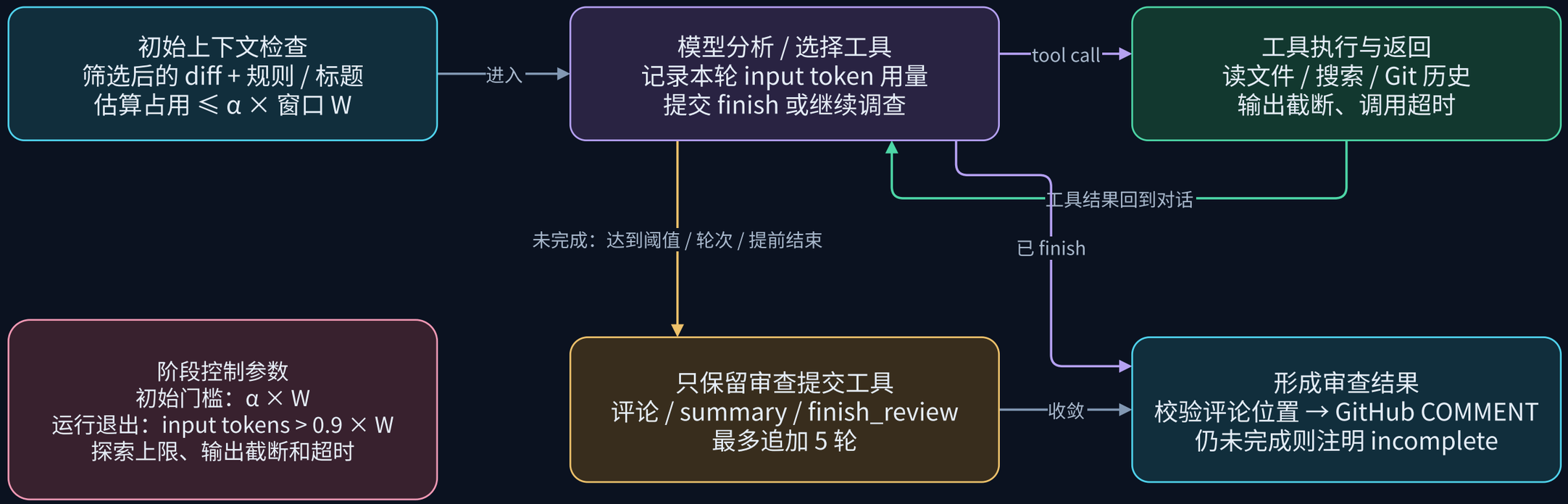
搜索调用方后，才能检查契约是否匹配。

> 固定 PR head → 读取相关文件 → 查找调用关系 → 形成有依据的审查建议。

> 工具提供按需探索能力；不代表每次读遍全仓库，也不自动保证判断正确。

Review loop：探索、反馈，再收敛到结构化结果

完整代码留在 repo cache；工具返回的片段随对话进入上下文，宿主控制每个阶段的工具集合与退出条件。



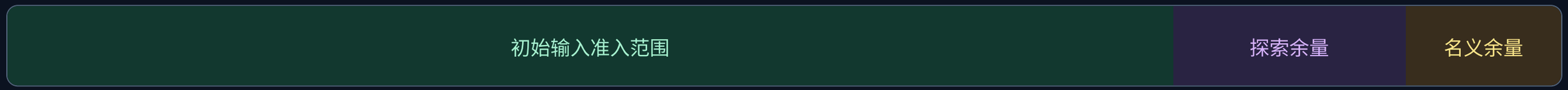
200k / α=75% 的配置示例：初始门槛约 150k；运行阈值 180k
阈值在 API 返回后检查；收集继续使用原对话，不代表严格保留了输出空间。

限制初始输入，为后续调查留出余量

完整代码留在缓存；筛选后的 diff、规则和按需工具结果进入模型上下文。

200k 窗口 / 75%

配置阈值示意，非实际用量或硬性预留分区



进入审查前

约 150k

初始门槛 = W × α
用字符数 / 4 估算，超限拒绝

运行中的退出阈值

180k

本轮 input tokens > W × 90%
结束探索，转入结果收集

窗口上限

200k

其余代码通过工具按需读取
结果截断，限制单次上下文增长

90% 检查发生在 API 返回后；收集沿用原对话。图中的名义余量不构成输出空间保证。

审查意见还要满足 GitHub 的发布契约

01 / Collector

模型提交结构化发现

path / line / body
summary / finish_review

02 / Validate

宿主检查评论位置

匹配本次 diff hunk
区间外的发现转到正文

03 / Deliver

发布并保留人工判断

宿主调用 GitHub API
AI 路径统一使用 COMMENT

一个具体工程问题：无效行号会影响整批提交

发布前校验位置，并为不能放在行内的发现提供正文入口。

> 模型负责提出观点，程序负责确定性的结构与位置检查。
> 语义正确性仍需要 reviewer 判断及相应验证。

把模型能力接入可运行的研发流程

本人工作

主要直接贡献

大仓库缓存、多 LLM 工具循环
上下文限制和工具异常处理
OpenAI 兼容适配、评论位置校验

系统能力

工程价值

按变更准备一致的代码上下文
让分析按需调用工具并收敛
把结果交付为 PR 可用的反馈

Git author 核验：cc1a2c5 · 02ae655 · 6f2352c · 84dd437 · 7c071c7

项目定位：面向代码审查的专用 agent harness

任务与上下文、工具执行、循环限制、结构化结果和工作流集成。

> 效果评估可看误报率、采纳率、历史缺陷检出和审查成本；当前没有统计就不报提升比例。

审查反馈如何进入开发协作

发现契约风险

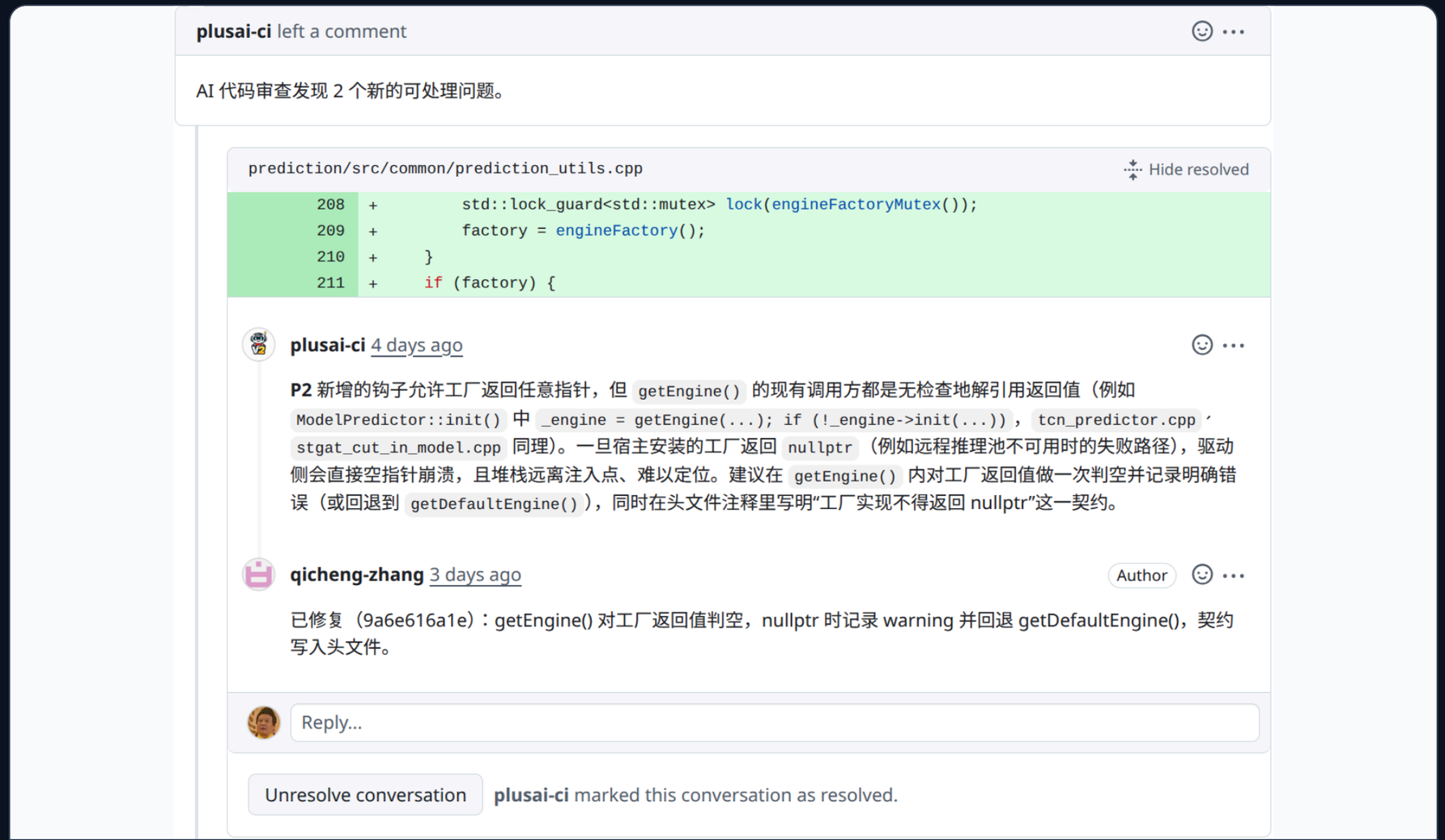
getEngine() 的返回值与调用方假设存在关联。

作者回应修改

截图显示判空与回退处理的回复，以及讨论被关闭。

来源仍需区分

评论带 geniecr:finding；对应扩展实现尚未定位。



用户提供的真实截图，裁剪保留原文。GenieCR 跟踪实现待核对。

已实现的运行支撑，以及后续演进方向

调度

任务状态

当前：进程内队列与在途去重
演进：持久任务、重启恢复与多实例协调

执行

工具执行边界

当前：命令 / 路径检查、超时和截断
演进：独立只读工作区与隔离执行环境

评测

质量与成本

当前：结构与行号校验、运行指标
演进：标注样本、误报 / 采纳率、成本基线

反馈闭环

跨提交问题生命周期

当前：按指定 head 做一次审查
待核对：GenieCR finding 跟踪与 resolve 实现

> 实际路由使用自实现 Anthropic / OpenAI 工具循环。
> 保留的 Claude Agent SDK 代码尚未接入当前主路径。