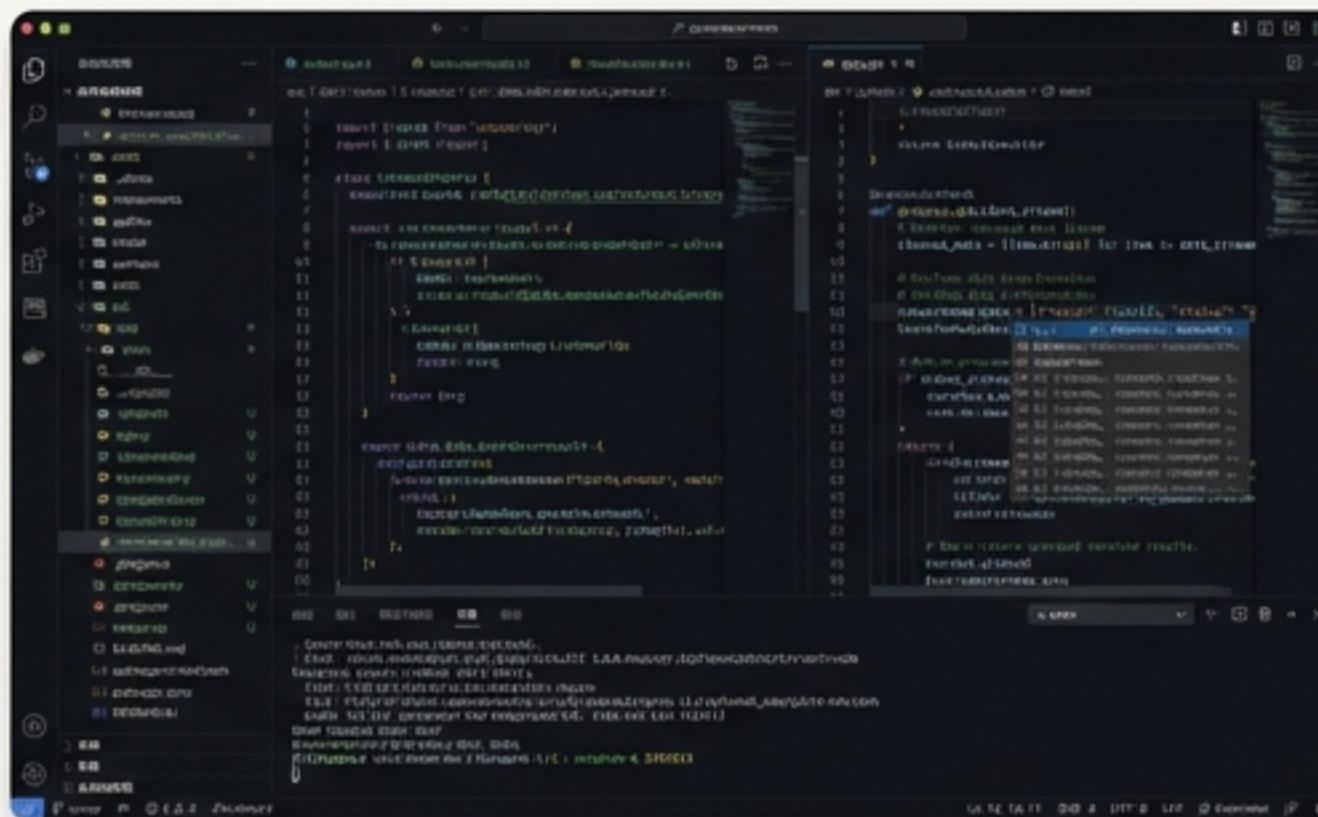


# 氛围、智能体与失效的工具

Armin Ronacher 对 2025 年软件工程的思考

# 2025：我停止亲手编程的那一年

“如果你在六个月前告诉我，我会更愿意担任一个**虚拟程序员实习生的工程主管**，而不是**自己敲击键盘**，我绝对不会相信。”



之前：Cursor

A screenshot of the Claude Code AI interface. On the left, there's a text input field containing the instruction: "用户：请为这个数据处理流程生成一个Python函数。". To the right, there's a response from the AI labeled "AI". The AI has generated the following Python code:

```
def process_data(data_stream):
    # Process incoming data stream
    cleaned_data = [item.strip() for item in data_stream if item]

    # Perform data transformation
    transformed_data = [{"value": float(d), "status": "ok"} for d in
        cleaned_data if d.replace('.', '', 1).isdigit()]

    # Return processed results
    return transformed_data
```

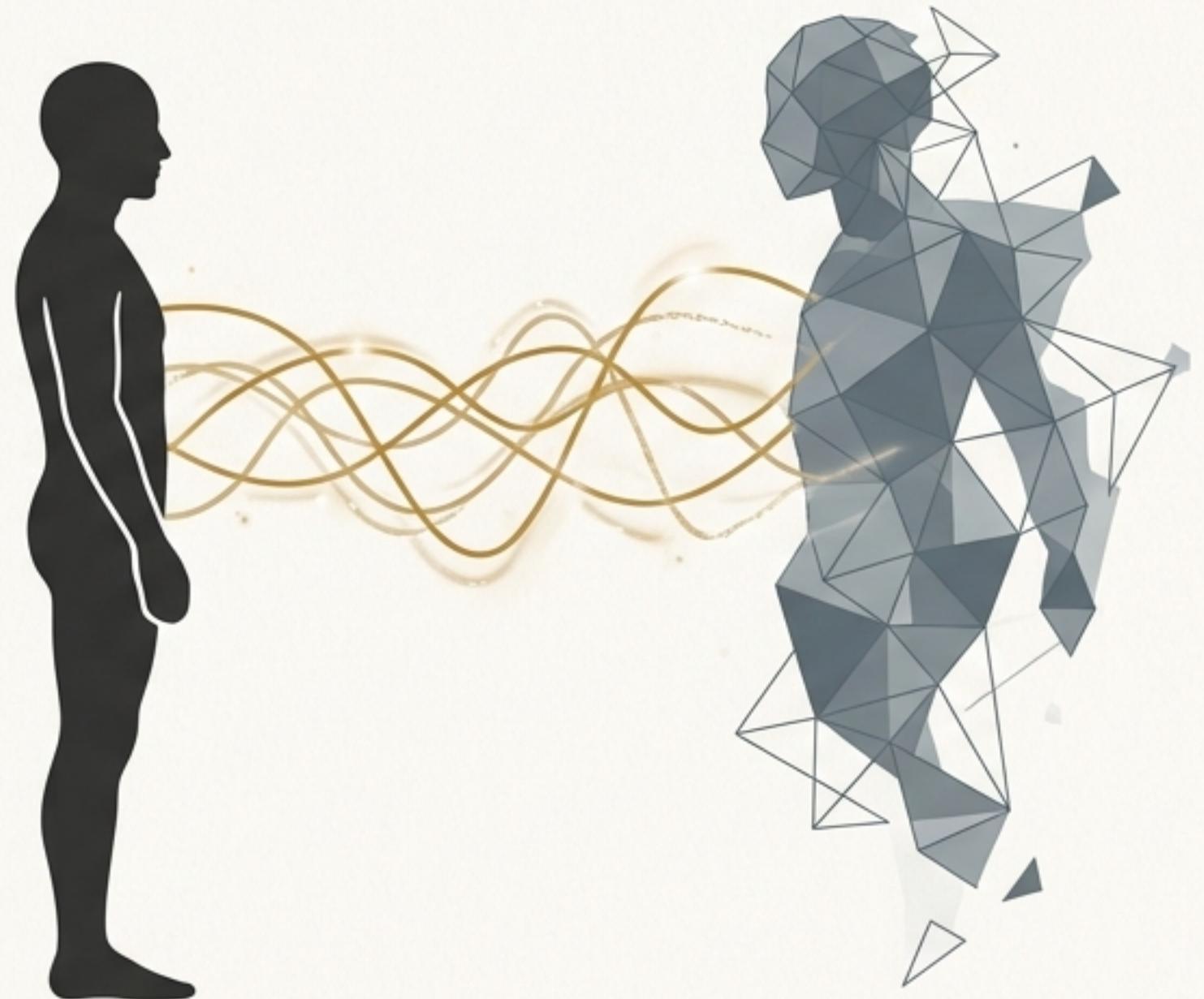
现在：Claude Code

# 我与机器：一种不安的伙伴关系

**拟人化的挑战：**与工具产生“拟社会关系”，令人感到奇怪和不安。

**“智能体”的局限：**能动性与责任应始终属于人类。

**情感的触发：**这些系统会触发我们的情感，需要小心应对。



# 一个由“氛围”驱动的工具箱

A

Amp

保时捷

代理编码工具中的苹果，  
精致、高端。

C

Claude Code

大众汽车

经济实惠的选择，  
高效、普及。

[π]

Pi

黑客之选

开源的选择，  
开放、可定制。

共同点：开发者为自己构建的工具，只是在权衡取舍上有所不同。

# 当“感觉不错”成为最高标准

**核心矛盾：**我们更多地谈论“氛围”(vibes)，这挑战了半个世纪的软件工程经验。

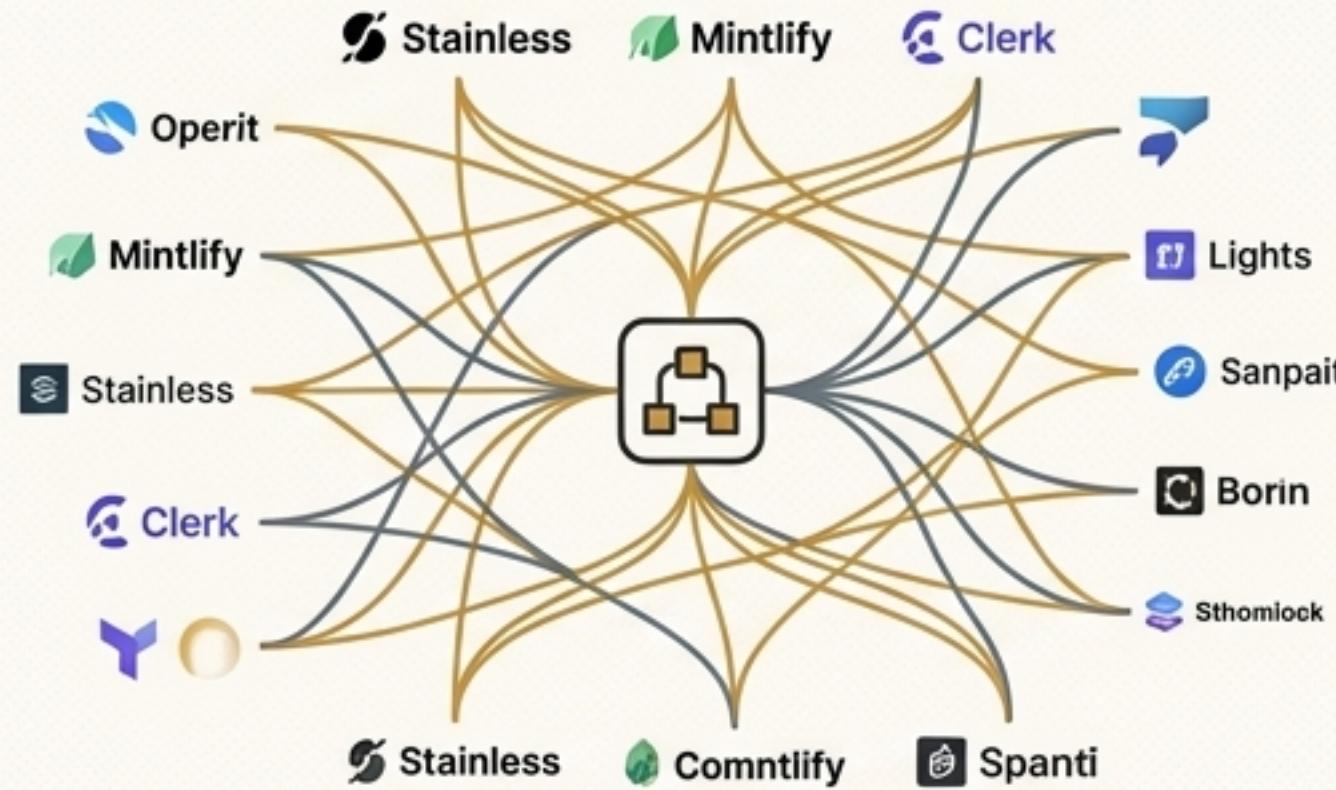
**缺乏数据：**观点多基于个人感受，而非客观指标。

**利益影响：**需警惕财务利益或有意“信号传递”对观点的影响。



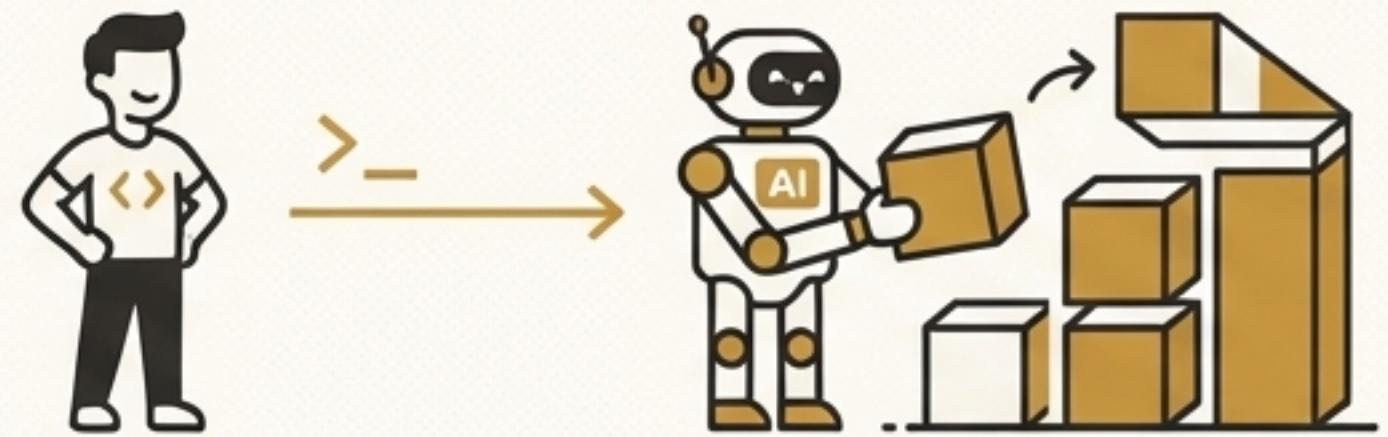
# 外包一切 vs. 自主构建

## 趋势一：万物皆可外包



提高用户体验的门槛，但也增加了依赖性。

## 趋势二：万物皆可自建



AI 赋能个人，轻松构建过去需要外包的复杂系统。

# 根本性危机：我们沿用半世纪的工具正在失效

失效的基石 I: GitHub 的 Pull Request 模型

失效的基石 II: Git 版本控制系统



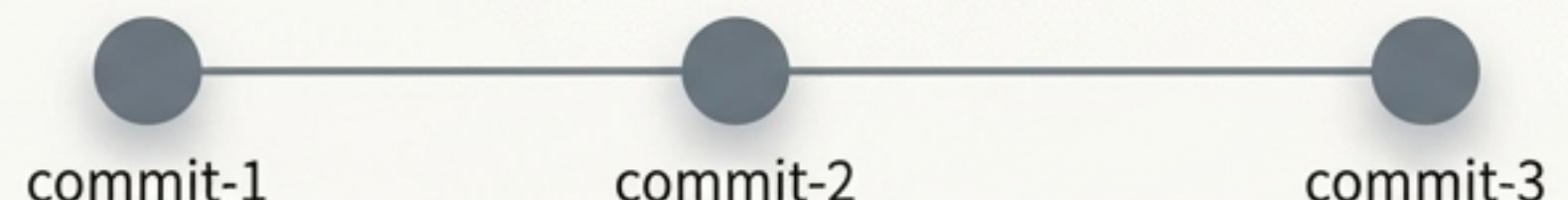
# Git 已无法承载“对话式”的编码历史

**局限 1：缺乏提示历史（Prompt History）**

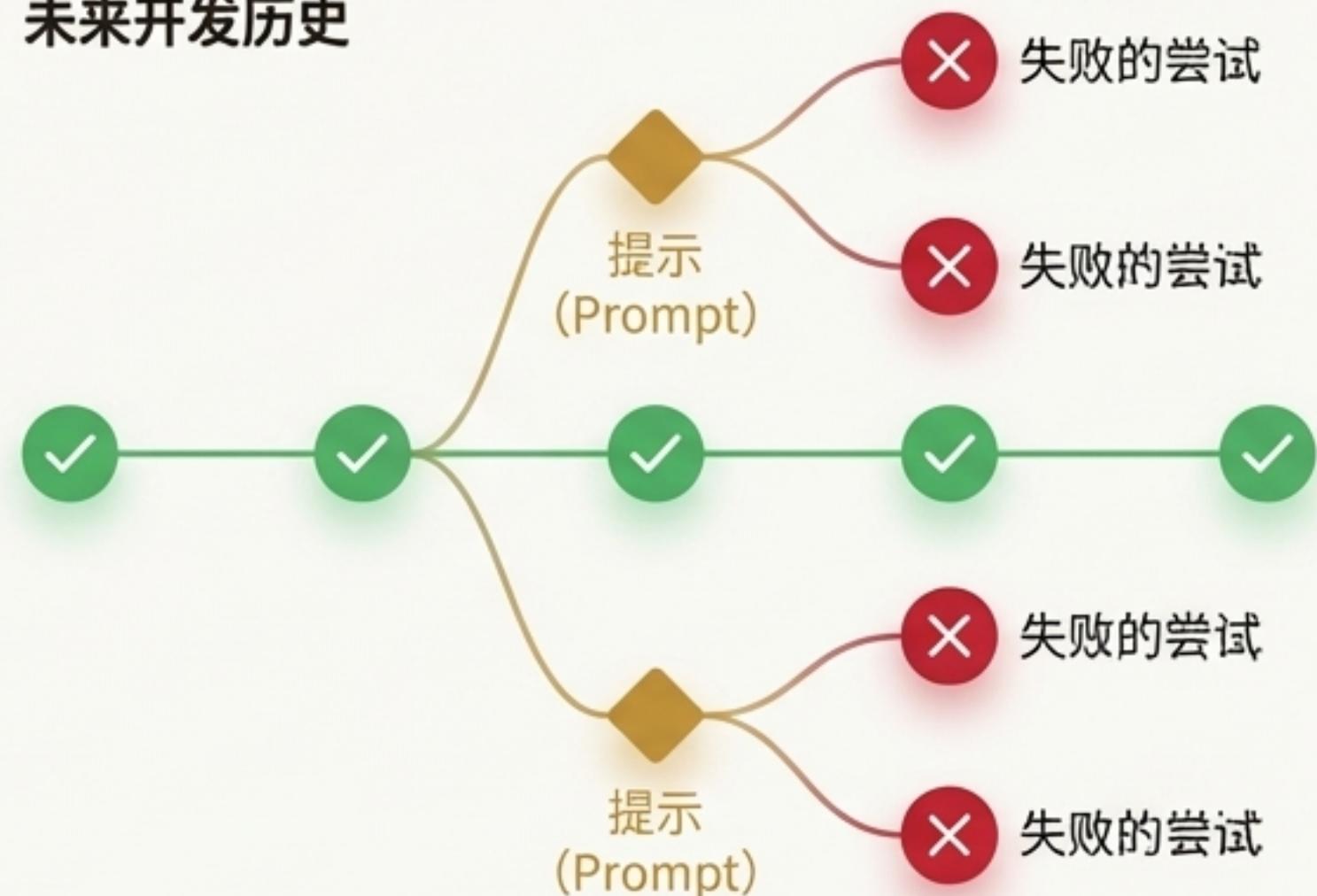
**局限 2：丢失失败的价值（Value in Failures）**

**局限 3：人类审查困难（Human Review）**

传统 Git 历史



未来开发历史



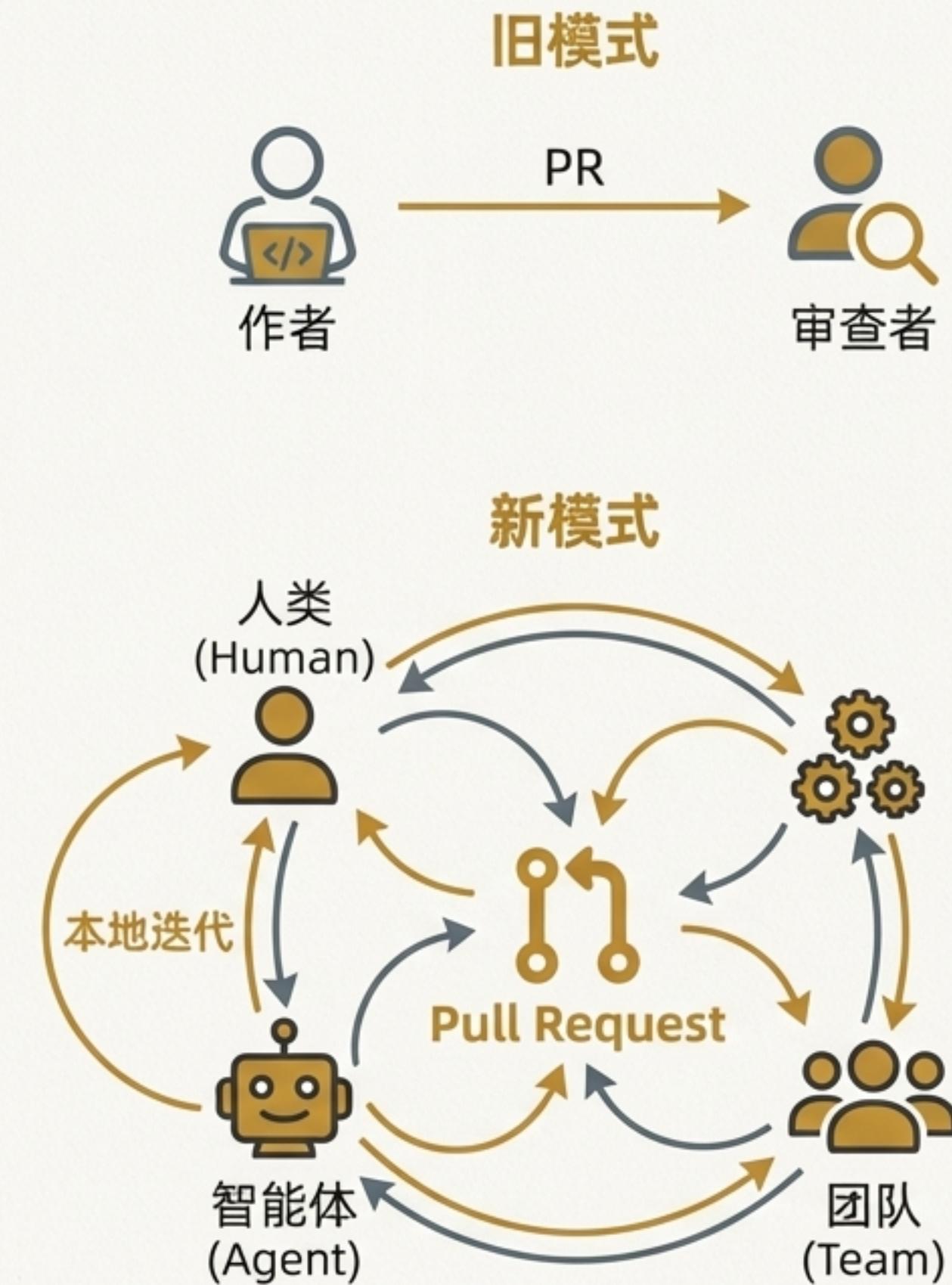
# 代码审查的边界正在模糊

**僵化的角色：**审查工具不允许作者“审查”自己的代码。

**不可见的迭代：**大量人机审查发生在本地，对团队不可见。

**核心愿景：**代码审查需要成为版本控制系统（VCS）的一部分。

Source Han Sans CN Regular



# 高质量工程与“代码倾倒”的冲突

**问题：**未经审查的机器生成贡献是对他人的“一种侮辱”。

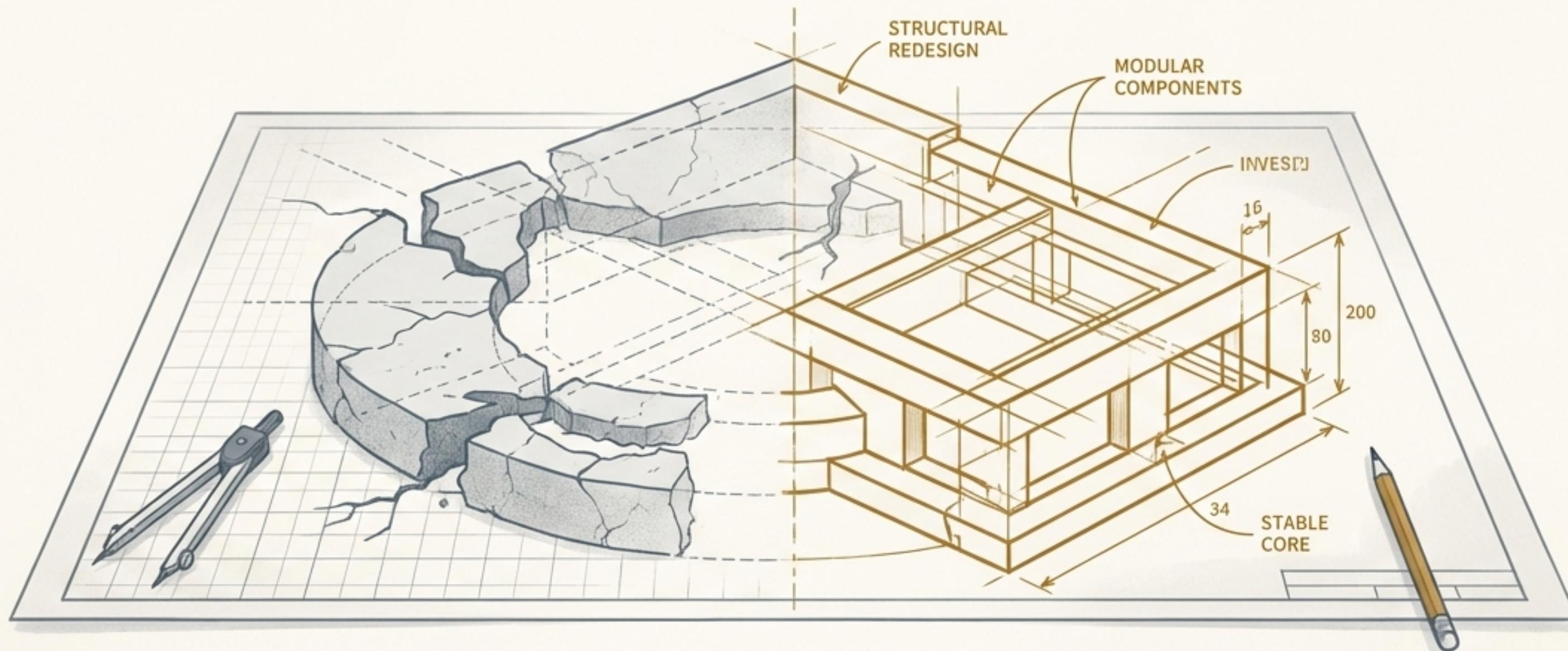
**感受：**阅读这样的 PR 令人“怒火中烧”(rage-inducing)。

**希望：**需要建立新的“社会契约”和良好行为规范。



# 重塑软件开发的基石

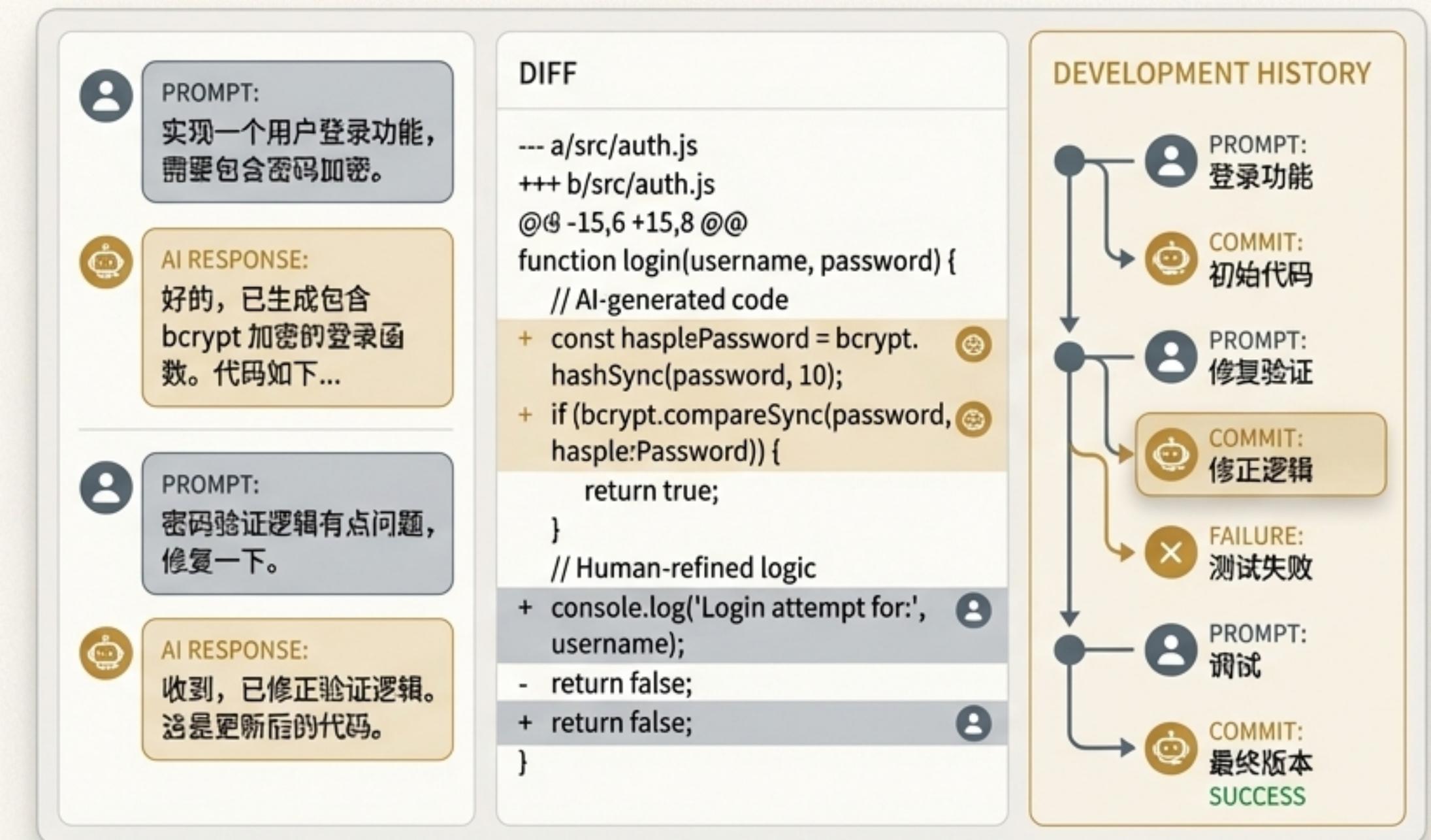
这不是预测，而是关于我们应在何处投入精力的“愿望”。



版本控制 · 代码审查 · 可观测性 · 工程文化

# 愿景一：可追溯、可审查的开发流程

- ✓ 集成提示历史
- ✓ 可见的迭代路径（包括失败）
- ✓ 区分人机贡献
- ✓ 智能历史压缩与检索

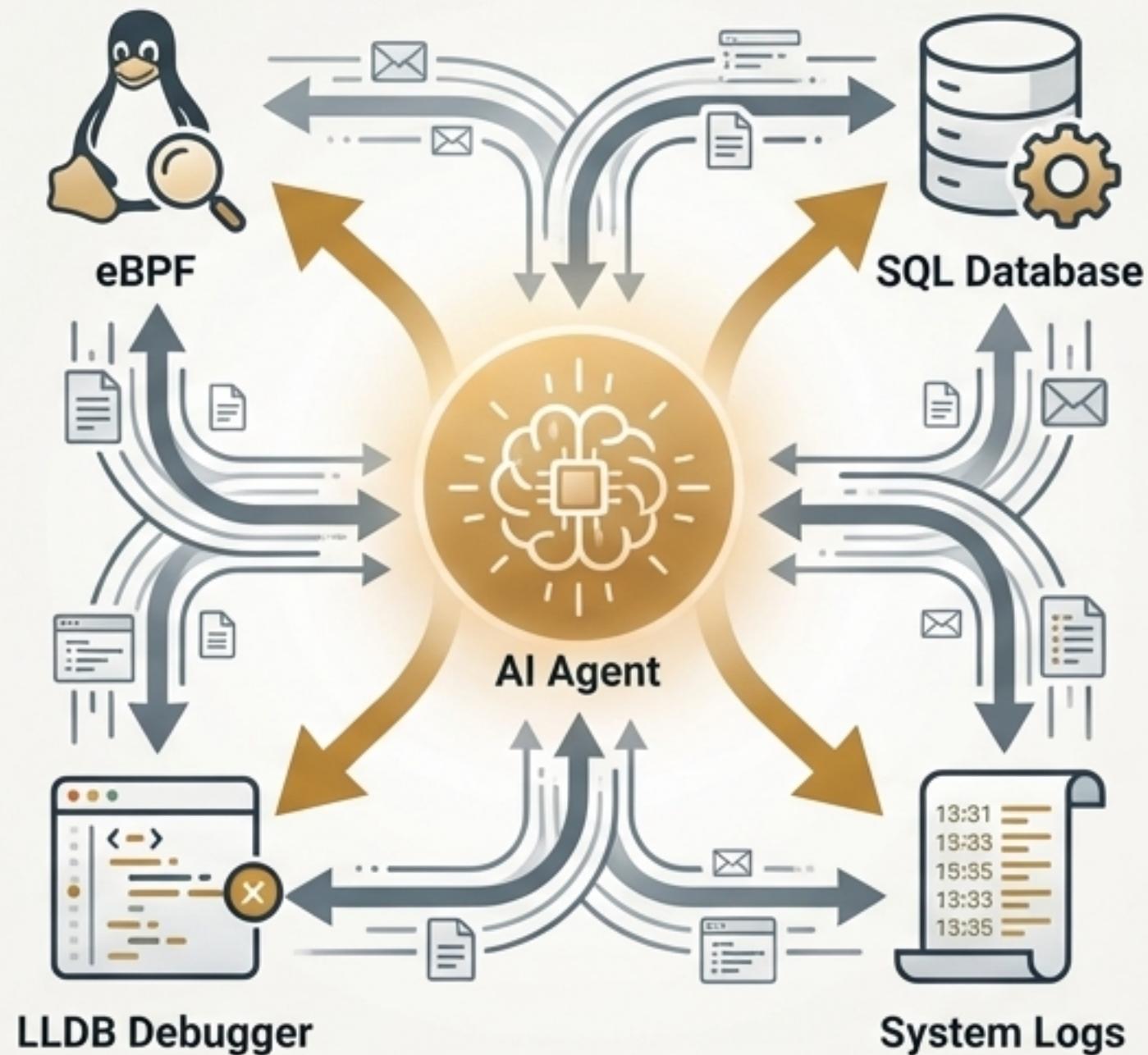


## 愿景二：赋予智能体洞察系统的超能力

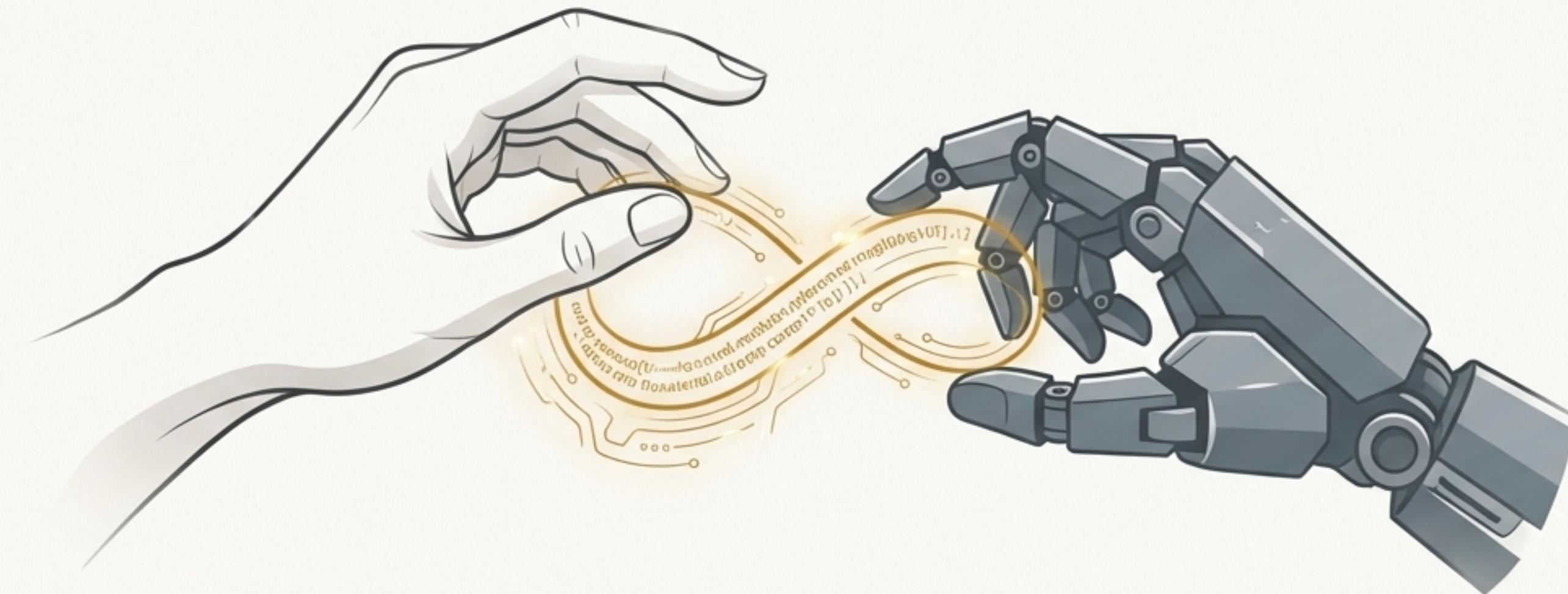
**解锁复杂查询：**LLM 可编写 eBPF、SQL 等复杂程序。

**实现动态配置：**由 LLM 完成过去因复杂而难以实现的用户不友好的“脏活累活”。

**核心逻辑：**给机器的反馈回路越好，结果就越好。



# 新时代的工匠精神



“这不仅仅是把未经审查的代码扔出来，  
然后让另一个人去收拾烂摊子。”

# 讨论与思考

我们如何为这个新时代做好准备？



内容灵感源自 Armin Ronacher 的文章《A Year of Vibes》