

第一部分

VibeCoding 的现状和瓶颈

你一定干过这件事

打开 IDE → 新聊天 → **Ctrl+A** → **Ctrl+C** → 切 AI → **Ctrl+V** → 「请提供更多上下文」→ 心态崩了 🌟

某工位贴着「禁止关机」大字报——不是因为怕丢数据，是 IDE 关了 AI 就断了 😂

全行业的日常：人肉 Copy-Paste 工程师 🦴

VibeCoding：听起来很美，用起来很累

✨ 宣传语

用自然语言指挥 AI 写代码
不用写代码了

😓 现实

你得守在屏幕前
next next next

📋 喂上下文

项目结构、技术栈、编码规范.....每次新对话重来一遍

👁️ 看输出

AI 改错地方了？你得帮它改

🔗 做 Git 操作

建分支、提 PR、解冲突，还是你的活

🔍 看 CI 日志

挂了？你来看日志定位修复

根本瓶颈——AI 没有记忆

AI 每次对话都是空白的——关掉窗口，一切归零

你**必须**一直守着它：不知道项目结构、历史决策、代码风格

全行业方案：超长上下文（账单爆炸）、RAG（检索到废弃代码）、记忆框架（换个项目重来）……本质都是给失忆症贴创可贴 🩹

瓶颈不在 AI 不够聪明，在于——

AI 没有住进代码仓库

你的项目上下文全在 Git 里，但它看不见。

第二部分

AI Native Git

从同步到异步，从工具到放大器

让 AI 住进仓库

Git 仓库是 AI 最完美的外部记忆——让 AI 原生嵌入仓库的生命周期

 **Issue**

= 需求记忆

做什么、为什么做

 **PR**

= 决策记忆

怎么做、Review 了什么

 **CI**

= 质量记忆

跑没跑通、哪里挂了

 **代码库**

= 完整记忆

每次 commit 写入，每次 diff 检索

「蓦然回首，记忆就在身后。」

核心差异——同步 vs 异步

维度	VibeCoding	AI Native Git
你要做什么	守在 IDE 前	说一句话，去干别的
AI 上下文	你手动喂	Git 自动提供
人机关系	人机绑定	人机解耦
Git / CI	你来	NPC 自动
你的角色	监督+动手	决策+验收 ✓
AI 是什么	写代码工具	决策力杠杆 ✓
并行能力	1 件	N 件 ✓

场景对比——修 Bug

VibeCoding (同步)

1. 开 IDE
2. 贴 Bug 描述 + 相关代码
3. AI 写修复
4. 手动建分支 → 提交 → 推送
5. 建 PR
6. CI 挂了 → 看日志修 → 再推

30-60min

AI Native Git (异步)

1. Issue 里 @NPC
2. NPC 自动读库、定位、修复
3. 建分支、提 PR、CI 通过
4. 你回来 Review
5. Merge 

< 2min



↓ 95%+
人工时间



零成本
上下文



全自动
全流程



提需求+验收
人的价值

场景对比——新功能开发

VibeCoding

1. 喂文档+规范+代码
2. 一段段让 AI 写
3. 自己拼装
4. 手动提 PR
5. 手动改 Review

1-3h

AI Native Git

1. @NPC + 设计文档
2. NPC 自动读库实现提 PR
3. Review
4. NPC 迭代

< 5min



↓ 97%+
人工时间



自动理解
架构



零摩擦
迭代



一人当一队
并行开发



同时 @ 多个 NPC —— 「一个人当一支团队」

AI 作为放大器

异步释放的不是时间，是决策力。



做关键决策

方向、架构、取舍
这些 AI 替不了



攻克 Key Point

人判断方向
AI 快速实现迭代



并行指挥

前端、后端、测试
同时推进

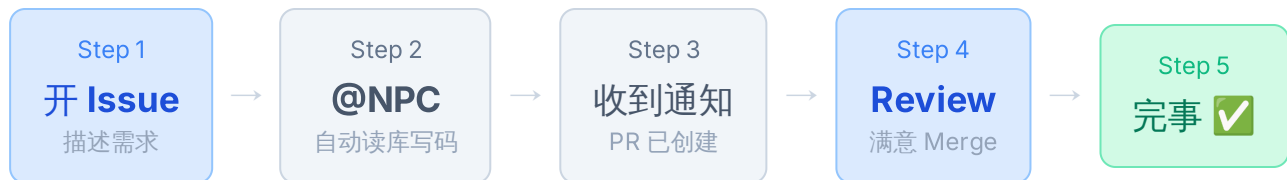
AI 不是替你干活的工具，
是 放大你决策力的杠杆。

你决策的质量，决定了团队产出的天花板。

第三部分

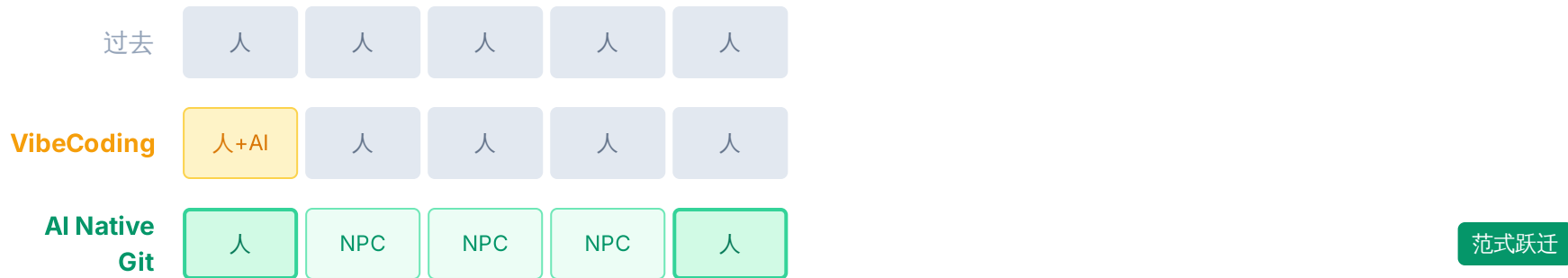
新研发流程

新流程长什么样？



指标	VibeCoding	AI Native Git	改进
Bug 修复人工时间	30-60min	< 2min	↓ 95%+
功能开发人工时间	1-3h	< 5min	↓ 97%+
并行任务数	1	N	N 倍
精力分配	80%执行+20%决策	10%验收+90%决策	质变

团队协作的跃迁



流程本身变了

不再需要全员盯着屏幕等 AI 输出——每个人只在自己负责的环节介入

VibeCoding : 你守着 AI 干活。
AI Native Git : 你验收 AI 的活。

AI 不是替你干活的工具，
是放大你决策力的杠杆。

AI Native Git , 启动。 

Q & A



来吧，尽管问