VOL.02 · 2026

AI CODING · 2.0

丰饶之后

AI Coding 观察报告 2.0

2025H2 – 2026Q1

腾讯研究院

AI 透镜系列研究

2026 年 4 月

CONTENTS

目录

PROLOGUE

开篇：9 个月后回望

From seven non-consensus questions to six structural insights.

2025 年 7 月，腾讯研究院发布第一版《AI Coding 非共识报告》，提出 7 个行业非共识，判断 “AI Coding 是通用 Agent 的先验战场”，将 “从 2,500 万开发者走向数十亿构建者” 作为愿景。9 个月后，这 7 条非共识的验证情况如下。

---

01 产品形态：本地 vs 云端

🔄 三极并存：CLI / IDE / Cloud

一版没有简单站队，而是用 “本地×云端 / 交互辅助×自主执行” 四象限切分出 IDE/插件、CLI、Vibe Coding、异步 Coding Agent 四类，并把 CLI 单独称为 “进可攻退可守的通用潜力股”。9 个月后，这个判断兑现方式超预期：CLI 不只是通用，而是全面赢得开发者内循环（Claude Code 8 个月成为最受使用和喜爱的工具）；IDE 继续在专业场景坚守并 Agent 化（Cursor 3、Google Antigravity、VSCode Multi-Agent）；Vibe Coding 产品向设计等通用场景迁移；云端异步 Agent 则在“龙虾热”下将 IM 变为交互入口。四象限结构仍然成立，重心向 CLI 与异步侧迁移。

02 模型选择：自研 vs 第三方

🔄 模型选择：趋同与分化

一版的 “自研 + 第三方” 四象限仍是理解模型策略的基本框架，指出 “多模型策略 + 智能路由” 正在成为主流。9 个月后，原问题 “该选哪家模型” 已被更深层问题取代：六大商业模型 SWE-bench Verified 压缩到 1 个百分点区间内，开源 Qwen3-Coder 追至 80% 段位。但 Anthropic 2026.4 同时发布 Mythos Preview（93.9%，不公开）与 Opus 4.7（87.6%，公开）的双轨机制表明，前沿实验室的能力储备与已公开模型之间正在拉开新的差距。

03 用户价值：提效 vs 降效

✅ 已跨越争议期

一版在这条上最审慎：同时摆出吴恩达 “效率提升至少 10 倍” 和 METR 随机对照实验 “AI 让开发者慢了 19%”，让争议成为真正的非共识。9 个月后，METR 同批参与者在 2026.2 的后续实验中逆转为快 18%（CI -38% 到 +9%），30-50% 开发者拒绝 “无 AI” 条件。争议期已跨越，但一版埋下的测量论（“自我报告的时间节省与 PR 吞吐量指标之间存在脱节”）在 V2 谈 AI 生产力时仍然值得引用。

04 付费模式：固定 vs 按需

✅ 按需/信用制成为主流

一版已明确判断 “混合模式 38% 超过订阅/席位制 36% 成为最主流”，指出 “传统 SaaS 的固定订阅模式在 AI 高变动成本下出现结构性问题”。这条验证最彻底：所有主流产品（Cursor / Claude Code / Copilot / Devin / Replit Agent）都走向 Token / Credit / ACU（Agent Compute Unit）等抽象计费单元的按需或混合制。一个延伸判断：AI 的成本倒逼驾驭工程，每次 Agent 失败都是直接成本，这成为企业投资驾驭框架的直接商业理由。

05 企业态度：激进 vs 渐进

✅ 两极分化加剧

一版用 “从强制使用到进入绩效” 描述激进派路径，摆出 Dario Amodei “3-6 个月内 AI 写 90% 代码” 的最激进预测。9 个月后：微软、谷歌内部 AI 代码占比约 30%、Meta 未到 50%，Amodei 的 90% 没达成，但激进做法仍在扩散：Microsoft、Shopify 把 AI 使用计入绩效，Perplexity 的 “强制使用” 被更多公司采纳，Jellyfish 调研的 “仅 22.5% 有正式政策” 分化继续放大。一版的 “两极分化” 判断准确，加剧程度超预期。

06 组织影响：裁员 vs 扩张

🔄 同时发生，不同技能层

一版的关键数据（软件开发岗位仅为 2020.1 的 65%、初级岗位从 30% 降至 20%、高级岗位从 30% 升至 40%、“10 人做 100 人的事”、1,000 万 ARR 规则被改写 Cursor 20 人 / 1 亿 ARR）9 个月后每条都被进一步印证。但也出现了一版未充分展开的新维度：AI 不是简单替代 N 个人，而是在**拉高下限**（非开发者进入构建）的同时**提高上限**（高级工程师杠杆放大）。Staff+ 工程师 63.5% 是最重度 Agent 用户，最有经验的人受益最多。

07 市场格局：专业 vs 普惠

✅ “先验战场” 充分验证

这是一版判断力最强的一条：Karpathy Software 1.0→2.0→3.0（code→weights→prompts）、“代码 ≠ 编程，意图将成为编程的核心驱动力”、Replit CEO Amjad Masad 的 “往下走 / 留在中间 / 往上走” 三象限，每一个框架 9 个月后都被广泛引用并进一步深化。Vercel 注册用户翻番、Cursor 36 万个人开发者、GitHub 个人仓库年增 217%，专业开发者没有被取代但角色重塑，非开发者正在以 “构建者” 身份进入软件生产。

7 条非共识的验证汇聚到一个更深层问题：当这些争论尘埃落定之后，2026 年的 AI Coding 呈现出哪些真正的结构性图景？本版提炼为 **6 个洞察**，依次展开于下文六章。

本版六个洞察

Six structural insights for the AI coding landscape, 2026.

❶ 模型加速趋同，前沿差距不减

六大商业模型挤在 1 个百分点区间内；但 Opus 4.7 一次性 +6.8pp 跳升、Mythos Preview 更领先 6.1pp——“内部突破 + 阶段性降权公开”的双轨发布机制正在形成。→ 第二章

❷ Agent 原生成为工具演化的收敛方向

工具形态走向 Agent-First（Cursor 3 / Codex App），工具接口走向 Agent-native（CLI 赢内循环 / MCP 退外循环 / Skills 补齐非开发者层）。→ 第三章

❸ 代码生成规模化，验证成新瓶颈

“怎么实现”退出核心瓶颈。新瓶颈出现在规格定义（向前）和验证维护（向后）两端——Veracode 45% AI 代码含已知漏洞、GitClear 技术债务增 30-41%。→ 第四章 4.1

❹ 产品构建零门槛，品味、运营逐渐稀缺

YC W2025 25% 创业公司 95%+ 代码 AI 生成。但“原型墙”普遍存在——分发、运维、合规、品味成为新稀缺。→ 第四章 4.2

❺ SaaS 没有死去，它正在被重新分配

三场“Anthropic Day”定点打击中间层 SaaS（FactSet -10% / IBM -13.2% / Figma -6.89%），同时 Cursor $50B、Skills 生态两极壮大。→ 第五章

❻ 做什么和谁能做，开发者被双向重定义

做什么在变：开发者从编写者转向编排者；谁能做也在变：非开发者首次以“构建者”身份进入。就业在三层之间流动。→ 第六章

01

CHAPTER 01 · 质变时刻

质变时刻

From augmented completion to autonomous collaboration.

AI 编码工具跨越了第二道能力门槛——从代码生成到自主协作。五维证据汇聚，商业验证加速。

THRESHOLD 01 · 2024

THRESHOLD 02 · 2025–2026

IN THIS CHAPTER

1.1 两道能力门槛

AI 编码工具的发展存在两个可识别的能力门槛。2021 年 GitHub Copilot 首发、2022–2023 年完成商业化铺开，那是 “辅助式补全” 的时代：IDE 里逐行、逐块的智能提示；此后每一次门槛跃迁，都由一款 Anthropic 模型定义。

第一道（2024 年中，Claude 3.5 Sonnet）：AI 编码从 “补全工具” 跃升为 “可靠的代码生成助手”。一版报告记录过这一时刻——Replit CEO Amjad Masad 将 3.5 Sonnet 的发布称为 “变革性的突破”。门槛跨越的直接结果是产品形态的重构：AI 的能力边界从 Copilot 式的行内补全扩展到 Cursor 式的全栈写作，Cursor 在此后一年内从小众工具成长为 1 亿 ARR 的新范式。

第二道（2025 年末至 2026 上半年，Opus 4.5 / 4.6 / 4.7 一代）：从简单任务辅助跨越到稳定的自主式协作。AI 能理解整个代码库、自主完成多步骤任务。正如 Sonnet 3.5 标志了 “AI 辅助编程” 时代，Opus 4.5 标志了 “AI 协作工程” 时代——首次在 SWE-bench Verified 上突破 80% 门槛。这一感知转变在 Opus 4.5 发布后约一个月里被开发者社区逐步确认，甚至迎来了“Claude Christmas”，开发者社区趁着圣诞假期集体切换到 Claude Code，品味 Opus 4.5 带来的新体验。

能力门槛时间线

- 2024 年中
  第一道门槛：Claude 3.5 Sonnet
  Coding Agent 真正开始可用；能力边界从 Copilot 式补全扩展到 Cursor 式全栈写作
- 2025.2
  Claude Code 预览发布
- 2025.7
  METR 发布实验报告
  第一版《AI Coding 非共识报告》
- 2025.11
  第二道门槛：Opus 4.5 发布
  从简单任务辅助到稳定的自主式协作；“Claude Christmas” 现象
- 2026.2
  Opus 4.6 / Sonnet 4.6，1M context，METR 结论逆转
- 2026.4
  Mythos Preview + Opus 4.7（9 天连发，双轨机制形成）

1.2 五维证据

技术：SWE-bench Verified 上，**Opus 4.5（2025.11）首次突破 80% 门槛，达到 80.9%**，token 消耗较上一代下降约 65%，这是 “AI 协作工程” 时代的质变起点。后续 Opus 4.6、4.7 持续演进，并在 2026 年 4 月出现 Mythos Preview 这一内部能力线（详见第二章、第五章）。

产品：Plan Mode、多 Agent 协作、1M context 窗口 GA、Claude Code Web / Mobile 接入全面铺开。

用户体验：“Claude Christmas” 现象：开发者社区在 2025 年圣诞前后集体切换到 Claude Code。

意见领袖：前特斯拉 AI 总监、OpenAI 创始团队成员 Andrej Karpathy 从 “模型就是垃圾” 转变为 “Opus 4.5 强了 10 倍”。Anthropic Claude Code 工程负责人 Boris Cherny 在 WIRED 专访中称 “编程基本上已经被解决了”，这一判断反映了工具开发者的乐观立场。

商业：Claude Code 收入从零增长到 10 亿美元（2025.12）再到 25 亿美元以上（2026.2）。Anthropic 估值增至 3,800 亿美元。a16z 2026.4 企业 AI 采纳报告显示 29% 的财富 500 强企业已是领先 AI 创业公司的正式付费客户（合同数据而非调查）；编码是 “领先近一个数量级的主导用例”，最佳工程师生产力提升 10-20 倍。

关键数字

80.9%
Opus 4.5 SWE-bench Verified
首次突破 80% 门槛

−65%
Opus 4.5 token 消耗
较上一代下降

29%
财富 500 强
已为头部 AI 创业公司正式付费

3,800亿
Anthropic 估值
（美元）

10–20×
最佳工程师
生产力提升（a16z）

25亿
Claude Code ARR
（美元 · 2026.2）

数据来源：Anthropic, Vellum AI, a16z Enterprise AI Adoption Report (2026.4), Yahoo Finance, Bloomberg

1.3 两个佐证

METR 研究的逆转。METR 于 2025 年初完成首次随机对照实验（使用 Claude 3.5/3.7 + Cursor Pro），2025 年 7 月发布结果——AI 让开发者慢了 19%。2026 年 2 月的后续更新逆转了结论：原始参与者组变为快 18%（置信区间 -38% 到 +9%，尚未达到统计显著水平），30-50% 开发者拒绝“无 AI” 条件。

METR 实验结论逆转

首次实验（2025 初）· 2025.7 发布

-19%

AI 让开发者变慢
Claude 3.5/3.7 + Cursor Pro

2026.2 · 后续更新

+18%

原始参与者变快
CI: -38% 到 +9%；30-50% 拒绝 “无 AI”

注：+18% 结果的 95% CI 为 -38% 到 +9%，尚未达到统计显著水平。数据来源：METR (2025.7, 2026.2)

Dogfooding（吃自家狗粮，指开发者自己使用自己开发的产品）。Claude Code 团队 95% 的代码由 Claude Code 编写；Claude Cowork 100% 由 Claude Code 编写，仅用 1.5 周。Anthropic 作为制造者的内部证言有利益相关性，Block 12,000 员工采用 AI 工作流、Cursor 67% 财富 500 强使用等第三方数据提供独立支撑。

95%

Claude Code 团队
代码由 AI 编写

100%

Cowork 代码
AI 编写，仅 1.5 周

12,000

Block 员工
已采用 AI 工作流

数据来源：WIRED (Boris Cherny 专访), Forbes (2026.1), The New Stack

02

CHAPTER 02 · 模型与驾驭工程

模型与驾驭工程

Convergence is commerce; divergence is the frontier.

商业模型在 coding 能力上趋同，前沿实验室的能力储备仍在加速分化。当模型趋同时，驾驭框架成为真正的竞争变量。

IN THIS CHAPTER

洞察 ❶

模型加速趋同，前沿差距不减。

2026.4 六大商业模型在 SWE-bench Verified 上压缩到 1 个百分点区间内——“选哪个模型”对大多数企业已不再是核心决策；但同月 Anthropic 在 9 天内先后发布 Mythos Preview（内部更强、不公开）与 Opus 4.7（一次性 +6.8pp 跳升），前沿实验室与已公开模型之间的能力储备差距在拉开。当模型趋同时，竞争力的决定因素转向驾驭框架与模型的协同。

2.1 商业模型趋同与前沿分化

截至 2026 年 4 月，SWE-bench Verified 排行榜清晰呈现两种趋势：

SWE-BENCH 排行榜 · 2026 年 4 月

模型	VERIFIED	PRO	说明
Claude Opus 4.5	80.9%	—	第二道门槛基准
Claude Opus 4.6	80.8%	53.4%	可用性进一步增强
Gemini 3.1 Pro	80.6%	54.2%	在Gemini 3 Pro基础上着重提升了编程
MiniMax M2.5	80.2%	—	价格民主化
Kimi K2.6	80.2%	58.6%	智能体群是亮点
GPT-5.4	~80.0%	57.7%	GPT+Codex分而又合

数据来源：Anthropic 官博、Vellum AI 第三方核对、SWE-bench 排行榜（2026.4）。Claude Opus 4.7（87.6%）与 Mythos Preview（93.9%）作为双轨发布机制的产物，不在此“趋同”区间，详见下页。

商业模型趋同。六家头部模型压缩在 1 个百分点区间。价格方面，MiniMax M2.5 单次对话成本约 $0.30/$1.20，比 Opus 4.5 的 $5/$25 便宜约 25 倍；开源阵营同步进入趋同核心区，Kimi K2.6 取得 Verified 80.2%、Pro 58.6%，与闭源旗舰并列。对不少企业用户而言，“哪个模型最聪明” 已不再是核心决策。

前沿能力分化。2026 年 4 月 16 日，Anthropic 发布 Opus 4.7，一次性将 SWE-bench Verified 推至 87.6%，较 Opus 4.6 的 80.8% 跳升 6.8pp；在更稳健的 SWE-bench Pro 上达 64.3%，领先 GPT-5.4 的 57.7% 达 6.6pp。这一跳升打破了持续 5 个月的商业趋同区间。

双轨发布机制浮现。更结构性的变化是：Opus 4.7 在 Mythos Preview（2026.4.7，SWE-bench Verified 93.9%，不对外发布）之后仅 9 天推出。Anthropic 在官方博文中明确：“Opus 4.7 是我们的第一个这类模型：它的网安能力不如 Mythos Preview 那么先进——实际上，在其训练期间我们试验了对这些能力进行差异化削弱的努力”——**“内部能力持续突破 + 阶段性差异化降权公开”**正在成为 Anthropic 的新发布节奏。

双轨发布机制 · MYTHOS 与 OPUS 4.7 的 9 天连发

2026.4.7 · 内部能力线（不公开）

Claude Mythos Preview

SWE-bench Verified **93.9%** · Pro **77.8%**；仅限 Project Glasswing 11 家防御性安全伙伴

↓ 9 天后，差异化降权 ↓

2026.4.16 · 外部公开线（GA）

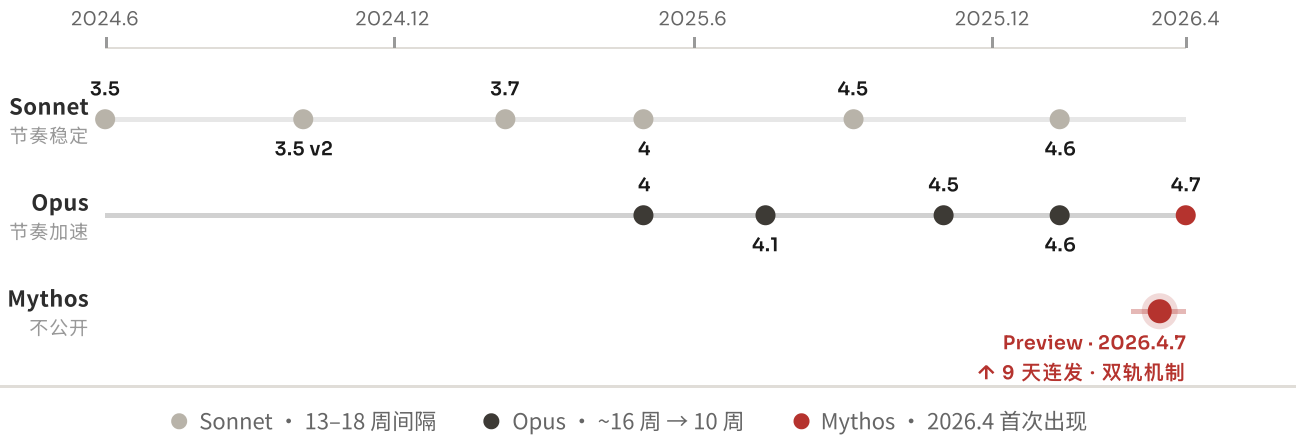
Claude Opus 4.7

SWE-bench Verified **87.6%** · Pro **64.3%**；Anthropic 官方定位 “first such model”

数据来源：Anthropic Opus 4.7 官方发布博文（2026.4.16）、Mythos Preview + Project Glasswing 公告（2026.4.7）

回望 22 个月的 Claude 发布史，如果把系列拆开看，三条曲线的含义并不一样：**Sonnet 系列**节奏稳定在 13–18 周，承担商业化铺量；**Opus 系列**节奏明显加速，从 ~16 周缩短到 10 周，承担旗舰能力推进；**Mythos** 2026 年 4 月首次出现，作为一条不公开的 “能力储备线”，与 Opus 4.7 仅隔 9 天形成双轨。我们看到的每一次公开模型升级，背后都有一个更强但不对外发布的版本作为源头。

CLAUDE 三系列发布节奏 · SONNET / OPUS / MYTHOS（2024.6 — 2026.4）



数据来源：Anthropic 官方发布记录（2024.6–2026.4）。Sonnet 系列某些早期版本发布日期为月份级保守口径。

2.2 驾驭工程（Harness Engineering）

驾驭工程指通过设计模型周围的系统（工作流、规格、约束、记忆、反馈循环）让 AI Agent 可靠工作的工程方向。当模型趋同时，这个新方向成为真正的竞争变量。

证据是直接的：SWE-bench Pro 上，scaffold（脚手架 / 驾驭框架）变化导致的分数波动是模型更换的 **22 倍**；同一模型 Opus 4.5 在不同 agent 系统上的得分差距可达 6-10 个百分点。

术语演进：2025.11 → 2026.4

- **2025.11**
  HumanLayer 创始人 **Dex Horthy** 在 AI Engineer Code Summit 首次系统阐述 “harness engineering” 术语
- **2026.2.5**
  HashiCorp 联合创始人 **Mitchell Hashimoto** 博文《My AI Adoption Journey》将概念推入主流开发者视野
- **2026.2.11**
  OpenAI 发布同名博文《Harness engineering: leveraging Codex in an agent-first world》引爆行业讨论
- **2026.4.2**
  ThoughtWorks **Martin Fowler / Birgitta Böckeler** 发布正典化长文《Harness engineering for coding agent users》

技能演进路径

OpenAI 的案例值得展开。团队曾每周五花 20% 时间清理 Agent 生成的低质量代码（“AI slop”），但发现这种人工清理不可扩展。转折点是将核心架构原则编码进仓库——用不变量约束 Agent 行为、用 “文档园丁” 智能体自动维护上下文、用生成 / 判别 / 修正三 Agent 循环替代单 Agent 工作。这一经验的本质是：**与其在下游修复 Agent 的错误，不如在上游约束 Agent 的行为空间**。

Anthropic 基于 Claude Agent SDK 的规划者→生成者→评估者架构同样强调 “分离做工作的 Agent 和评判工作的 Agent”。Stripe、Shopify、Airbnb 等企业已在实践各自的驾驭框架。

Multi-Agent 编排：以小见大

Claude Code 的演进路径体现了多 Agent 协作的设计逻辑。2025.7 发布的 Sub-agents 解决的是一个具体问题：单个 Agent 上下文过长时会失去连贯性，用子 Agent 隔离上下文可以保持聚焦。2026.2 发布的 Agent Teams（实验性）则解决了更高一层的问题：多个 Agent 之间如何协调——Team Lead 分配任务，Teammates 通过 Mailbox 直接 P2P 通信，共享任务列表支持依赖管理。从 “一个大脑指挥手脚” 到 “一个团队各司其职”，这一跃迁的实质是将人类软件工程中的分工协作模式迁移到 Agent 层面。

2025.7 · SUB-AGENTS

一个大脑指挥手脚

单 Agent 上下文过长时失去连贯性，子 Agent 隔离上下文保持聚焦

↓

2026.2 · AGENT TEAMS

一个团队各司其职

Team Lead 分配任务，Teammates 通过 Mailbox 点对点通信，共享任务列表支持依赖管理

Kimi 从 K2.5（2026.1）到 K2.6（2026.4.19）将这一方向推到极致：**最多 300 个 sub-agents 并行、12 小时连续执行、4,000+ tool calls**，采用 PARL 训练，以 open-weight 方式进入趋同核心区。Codex Subagents（explorer / worker / default 三角色）、VSCode Multi-Agent、Cursor Agent Tabs 全面跟进。**并行多 Agent 已成为基线预期**。

Claude Code 源码泄漏的启发

2026 年 3 月 31 日，Claude Code v2.1.88 在 npm 发布时意外暴露了 512,000 行 TypeScript 源码（数小时内被 fork 超 41,500 次，Anthropic 已撤回泄漏版本）。这次事件意外提供了关于驾驭框架设计的最完整公开案例：40+ 离散能力的 Tool System（类 Unix 系统调用层）、46,000 行的 Query Engine、**KAIROS 守护进程**（将 Agent 从请求-响应模式转变为持久后台进程，目前在主流开源框架中尚未出现类似设计）、**autoDream 记忆整合系统**（空闲期间跨会话合并观察）。VentureBeat 评价这为竞品提供了 “字面意义上的蓝图”。同时，这一事件也表明 AI 工具自身正成为供应链安全的新维度（详见第五章）。

512,000 行

TypeScript 源码
意外暴露

41,500+

数小时内
被 fork 次数

2.3 模型与驾驭框架的协同进化

驾驭工程不是一次性的框架搭建，而是一个随模型进步持续演变的设计过程。Anthropic 的长时运行 agent harness 实践提供了最清晰的案例。

每个 harness 组件都编码了一个关于模型局限的假设。在 Sonnet 4.5 时代，Anthropic 的驾驭框架需要将任务分解为 sprint、每个 sprint 后重置上下文窗口，因为模型在上下文变长时会失去连贯性，甚至出现 “上下文焦虑”（context anxiety），提前匆忙结束工作。这些组件是必要的，但也引入了编排复杂性和额外延迟。Opus 4.5 发布后，情况发生了变化：模型可以连续工作 2 小时以上不需要 sprint 分解，上下文焦虑基本消失。团队因此移除了 sprint 构造，改为单次连续会话配合自动 compaction。

协同进化：模型进步如何重组驾驭框架

Opus 4.7 把这种协同推向新深度。合作伙伴在 Anthropic 官方博文中反馈：“它甚至会在开始系统代码前先做证明（does proofs on systems code before starting work），这是过去的 Claude 模型从未表现过的新行为”——这意味着**模型开始自己承担一部分验证工作**。原本属于 harness evaluator 组件的 “验证” 功能，正在被模型能力吸收。但驾驭框架的 evaluator 并不会消失——它只是向 “未被模型能力覆盖的验证边缘” 移动。

这种动态关系有一个重要推论，也是 Anthropic 博客的核心结论：模型每一代进步都重新划定哪些复杂性应留在驾驭框架、哪些已被模型能力吸收。这使得驾驭工程成为一个持续的系统设计学科——它不会被更好的模型淘汰，反而随模型进化持续创造新的设计空间。

> *有趣的驾驭组合空间不会随模型进步而缩小，而是移动。*
>
> — Anthropic,《Harness design for long-running application development》, 2026.3.24

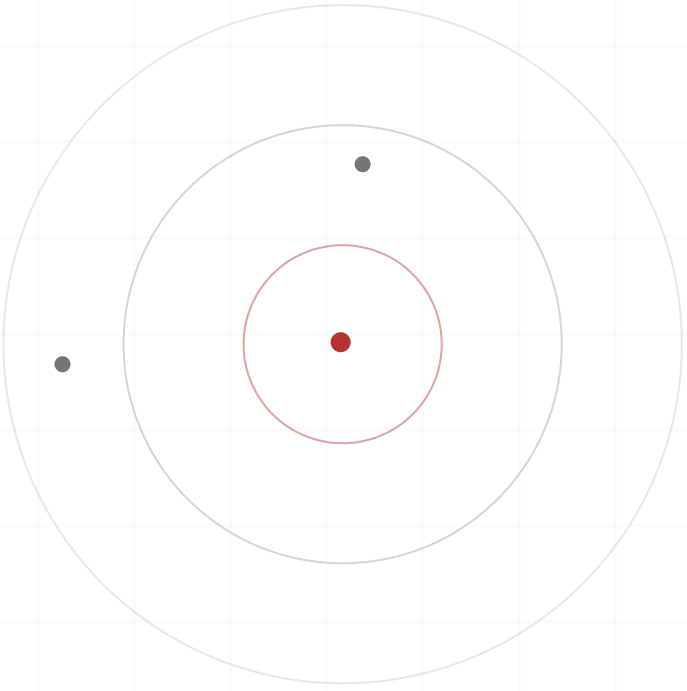
03

CHAPTER 03 · 工具生态的重塑

工具生态的重塑

Agent-native is the converging direction of tool evolution.

形态走向 Agent-First，接口走向 Agent-native。Skills 补齐非开发者层级，三层分工，同一个收敛方向。



IN THIS CHAPTER

洞察 ❷

Agent 原生正成为工具演化的收敛方向。

形态层面，Cursor 3、Codex App、Google Antigravity 等 Agent-First 应用把 IDE 从 “代码编辑器+AI 插件” 升级为 “Agent 编排平台+代码视图”，工具的主要使用者从人转向 Agent 组；接口层面，CLI 赢得 Agent 内循环（代码是 Agent 的原生语言），MCP 退守企业外循环，Skills 以 SOP 封装成为非开发者的首选接口。形态、接口两个层面都在向 “Agent 原生” 收敛——**给 Agent 最好的工作环境，是 Agent 编排平台；给 Agent 最好的能力接口，是代码与 SOP**。

3.1 Agent-First 转型：工具形态的重新定义

Pragmatic Engineer 2026.1-2 对 906 名开发者的调查显示：Claude Code 发布仅 8 个月即成为最受使用和最受喜爱的工具（46% 喜爱度，Cursor 19%，Copilot 9%）。小公司 75% 选 Claude Code，大企业 56% 选 Copilot。

三极格局：ARR 与用户规模

产品	ARR	估值 / 用户
Claude Code	>$25 亿	Anthropic 整体 $3,800 亿
Cursor	$20 亿	$293 亿 → 谈 $500 亿
GitHub Copilot	~$20 亿	N/A（Microsoft 旗下）
Codex App	—	400 万周活（2026.4.8）

数据来源：Yahoo Finance, Bloomberg, TechCrunch, OpenAI (Sam Altman 2026.4.8 推文)

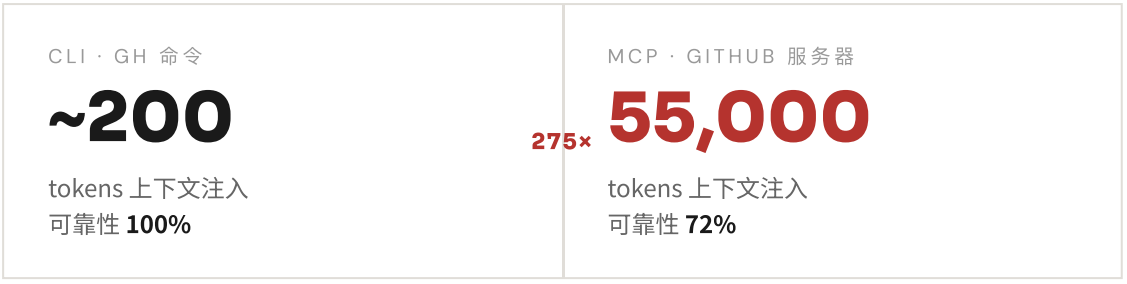
但 ARR 排名只是表面格局。真正的结构性变化是 **IDE 的定位从 “代码编辑器 + AI 插件” 升级为 “Agent 编排平台 + 代码视图”**。Cursor 3.0（2026.4.2）推出 Agents Window 支持跨仓库并行运行多个 Agent，自称 “AI 软件开发的第三纪元”。Google Antigravity（2025.11）从第一天就围绕 Agent 编排设计。VSCode 全面跟进 Agent Mode 和多 Agent 编排。OpenAI Codex App 2026.2 以 macOS 独立桌面应用形式发布，定位为 “agentic software development 的指挥中心”，可同时管理多个 AI 编码 agent，上线两个半月周活跃用户达 400 万。

Agent-First 的本质是：工具的主要使用者从人转向 Agent 组。敲键盘的人不再直接产出代码，而是编排、审查多个 Agent 的并行产出。从补全（2022-24）到 Agent（2024-25）到 Agent-First（2026-），开发工具形态正在被重新定义。

3.2 CLI vs MCP：Agent-native 的接口语言

2026 年 3 月，一场关于 Agent 如何获取外部能力的辩论引发关注。OpenClaw 开发者称 “MCP 是个错误”，Perplexity CTO 宣布弃用 MCP。

表面上看，争议焦点是效率：GitHub MCP 服务器注入 **55,000 tokens** 上下文，`gh` CLI 仅需约 **200 tokens**，差距 275 倍。ScaleKit 实测 CLI 可靠性 100% vs MCP 72%。但效率差距只是表象，背后有一个更深层的原因。



数据来源：ScaleKit benchmark, Reddit r/ClaudeAI (2026.3)

CLI 对 Agent 来说是原生语言。AI Coding Agent 的核心能力就是写代码和执行命令。当 Agent 调用 `gh pr list --json` 时，它在做的事和写一行 Python 代码没有本质区别：都是生成一段文本指令并执行。LLM 的训练数据中包含海量真实世界的 shell 命令和脚本，CLI 调用是模型最熟悉的交互模式。正如 Cloudflare 在 Code Mode 技术博客中指出的：“LLM 写代码来调用 API 比直接用工具调用更好——因为训练数据中有海量真实代码，但只有少量人造的 tool call 示例。” MCP 的工具调用协议则需要注入协议描述、解析工具列表、格式化调用、解析返回，每一步都是额外的抽象层。这不是 MCP 设计差，而是对 Agent 来说，**CLI 是母语，MCP 是需要翻译的外语**。

行业正在收敛到分层路由：CLI 赢在本地快速迭代（内循环），MCP 赢在企业跨系统协调（外循环）。MCP SDK 月下载仍达 9,700 万，在集中化认证、结构化输出、审计日志等企业场景中不可替代。

中国市场的集中 CLI 化（2026.3 一周内）

应用	时间	覆盖范围
网易云音乐	2026.3.23	首个开放核心音乐服务的 CLI
钉钉	2026.3.27	AI 表格、考勤、日历、审批等
飞书	2026.3.28	2,500+ API，11 个领域
企业微信	2026.3 底	长连接 Bot 模式

中国独特之处在于**平台驱动**（超级应用主动 CLI 化），海外更多是**开发者驱动**。这条路径可能跳过 IDE 阶段直接在 IM 中使用 Agent。

3.3 Skills：Agent 生态的 SOP 层

CLI 和 MCP 都是面向 Agent 执行层与集成层的接口，它们解决了 “Agent 如何做事” 的问题。但还有一个更高的抽象层在 2026 年清晰浮现：**SOP（标准作业程序，Standard Operating Procedure）本身的封装与复用**。

Skills 是什么。Anthropic 2025.10.16 发布的 Agent Skills，把一个文件夹（`SKILL.md` + 可选脚本与资源）作为 Agent 的 “入职手册”。启动时 Claude 只加载每个 Skill 约 100 tokens 的元数据；触发后再展开正文、按需执行脚本。这套 “渐进式披露”（Progressive Disclosure）让一个 Agent 可以挂上无界数量的专业能力而不炸上下文。2025.12.18，Anthropic 把规范开放为 **agentskills.io** 标准，使其从 Claude 特性升级为跨厂商 SOP 封装协议。

三层架构：SKILLS / MCP / CLI 不是竞争而是封装

层级	承担	用户面向
Skills（SOP 层）	何时做、怎么做、失败如何恢复	领域专家 / 非开发者
MCP（集成层）	连外部系统：API、数据库、SaaS	平台 / 集成工程师
CLI（执行层）	基础设施操作、工具母语	Agent 与开发者共用

三份独立技术分析（Anthropic 官方、IntuitionLabs、Milvus / Zilliz）给出一致的分层模型

三者**不是竞争而是封装**：Skills 可以直接调用 CLI（Anthropic 官方 `pdf` Skill 就是 markdown + Python 脚本的形态），也可以与 MCP 配合（MCP 拿数据、Skill 解读数据，即 “MCP fetches, Skills interprets”）。

Skills 让非开发者第一次成为 Agent 作者。Skills 的最小形态仅需一个带 YAML 头的 markdown 文件，一个没有编程背景的用户可以在 10 分钟内写出第一个 Skill，不用写一行代码。对比之下：写 MCP 服务器需要理解 JSON-RPC 协议和服务器开发；写 CLI 工具需要 Shell 或 Python 能力。Anthropic 的 Skilljar 官方课程、Snyk 面向 solopreneur（独立创业者）的 “Top 8 Skills” 专题、Axton Liu 这样的内容创作者为自己的写作/视频/商业分析构建 80+ Skills——这些都指向一个新的生态定位：**Skills 是 Agent 生态里第一个让非开发者直接成为作者的层级**。

生态规模一个季度翻倍。至 2026.4，SKILL.md 已有 **30+ 工具**原生或兼容支持，明列包括 Claude Code、OpenAI Codex CLI、GitHub Copilot、Gemini CLI、JetBrains Junie、Cursor、Windsurf、Amp、Goose、Roo Code、Trae、OpenCode、Letta 等。GitHub Copilot 甚至一次识别 `.github/skills/`、`.claude/skills/`、`.agents/skills/` 三个目录，坐实了跨厂商意图。

SKILLS 生态规模（2025.11 → 2026.4）

30+ 工具原生或兼容 支持 SKILL.md 规范	**2,600+** Curated Skills 目录 （一个季度翻倍）	**90万+** SkillsMP 聚合索引 （含爬取/重复）
10万 Everything Claude Code 单仓库 GitHub star	**42K** Superpowers（J. Vincent） GitHub star，已入官方市场	**10分钟** 非开发者写出 第一个 Skill 的时间

数据来源：noqta.tn SKILL.md 30+ 工具综述 (2026.4)、SkillsMP、James Bachini《Claude Code Skills Gold Rush》、Ronnie Parsons

> *Skills 是 Agent 生态里第一个让非开发者直接成为作者的层级。*
>
> — 本报告研究判断（基于 IntuitionLabs 2026.2 / Milvus 2026.4 / noqta.tn 2026.4 三份独立技术分析）

Skills 与 SaaS 的关系耐人寻味。SaaS 把 SOP 封装成带收费 UI 的应用；Skills 把 SOP 解封装成任何 Agent 可调用的开放 markdown 标准。中间层单功能 SaaS 的 SOP 部分正在被 Skills 吸收，这也是第五章 “SaaS 重新分配” 的一个底层机制。

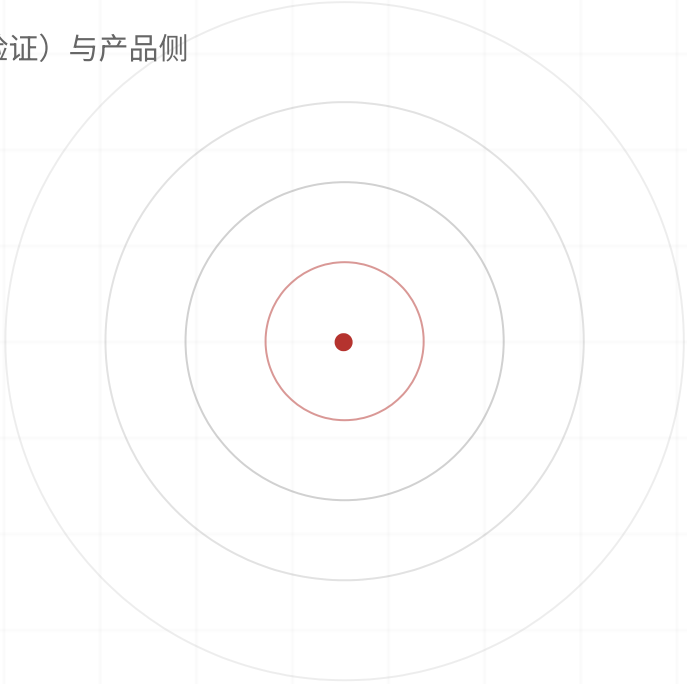
04

CHAPTER 04 · 当构建不再稀缺

当构建不再稀缺

Abundance in building. Scarcity in shipping.

代码生成退出瓶颈，新的瓶颈同时在工程侧（验证）与产品侧（分发运营）两端出现。



IN THIS CHAPTER

第二章展示了模型能力趋同和驾驭工程的成熟。第三章呈现了工具生态的 Agent-First / Agent-native 重塑。这两个因素叠加产生了一个深层后果：**代码生成正在退出瓶颈位置**，但新的瓶颈同时在工程侧和产品侧两端出现。本章讨论这两组新瓶颈。

洞察 ❸

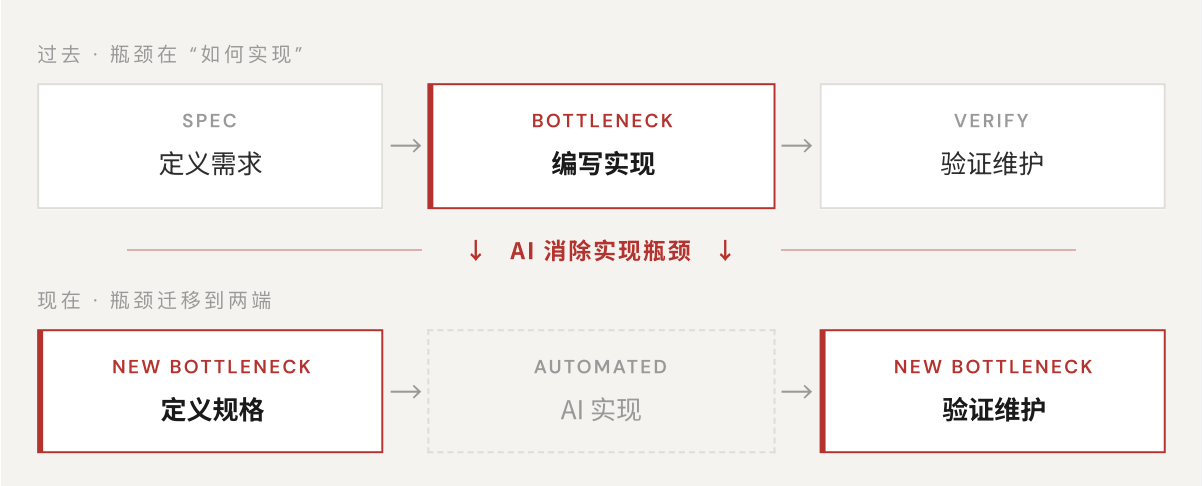
代码生成规模化，验证成新瓶颈。

SWE-bench 87.6%、Opus 4.7 甚至开始在写代码前自证，“怎么实现” 正在退出核心瓶颈。新瓶颈出现在向前的 “规格定义” 和向后的 “验证维护”：Veracode 45% AI 代码任务含已知漏洞，GitClear 2.11 亿行代码分析技术债务增 30-41%。下一波价值创造不在更好的代码生成，而在更好的规格、验证、维护基础设施。

4.1 瓶颈迁移：代码生成规模化，验证成新瓶颈

当 SWE-bench 87.6% 意味着大多数标准编码任务可以被自动完成时，“如何实现” 不再是软件工程的核心难题。Lean 定理证明器创始人、Amazon Web Services 应用科学首席研究员 Leo de Moura 早在趋同来临前就预判了这一转变：“验证、测试和规格一直是瓶颈，不是实现。”

瓶颈迁移：从 “如何实现” 到两端



量化数据正在验证他的判断：

向后看：验证与维护成本上升。Veracode 发现 45% 的 AI 代码任务引入已知安全漏洞。GitClear 分析 2.11 亿行代码后发现，虽然代码产量大幅增长，但代码重复量增加 4 倍、重构活动下降 60%、技术债务估计增加 30-41%。Apiiro 发现 AI 代码每月引入超过 10,000 个新安全发现，6 个月内增长 10 倍。AI 让生成代码变得极为廉价，但验证代码是否正确、安全、可维护的成本并没有同步下降。

45%
AI 代码任务
引入已知安全漏洞（Veracode）

30–41%
技术债务
估计增加（GitClear 2.11 亿行）

10×
AI 代码新安全发现
6 个月内增长（Apiiro）

向前看：规格定义成为新核心工件。KTH 皇家理工学院的实验将瓶颈迁移的逻辑推演到了极致：AI Agent 可从 **926 字英文规格完整自举自身**。当规格（specification）变成稳定的核心工件、而代码变成从规格生成的可再生副产品时，工程的重心自然从 “写代码” 转向 “写规格”。开发者社区已经出现了这种转变的端倪，有人描述新工作流为 “用英语编程，提交的是 Go 语言”。

AI 也开始在验证端发力。Claude Code Review 将 PR 获实质性评论率从 16% 提升至 54%。Opus 4.7 的 “自我验证” 行为（合作伙伴观察到它 “在开始系统代码前先做证明”）把一部分验证工作内化到模型本身。但整体上，验证基础设施的进步远远落后于生成能力的飞跃。a16z 2026.4 数据显示，49% 的组织还没能用 AI 构建可用的企业软件：瓶颈并非在生成侧，而在验证和整合侧。

范式转变：Karpathy 在 2025 YC 演讲中把这一转变概括为 **Software 1.0（code）→ 2.0（weights）→ 3.0（prompts / specs）**。一版报告 9 个月前引用了这一框架，在 Opus 4.7 和 KTH 实验的今天，它正在从理论落地为工程实践。

Andrej Karpathy 在 YC 演讲中提出的范式演进图

洞察 ❹

产品构建零门槛，品味、运营逐渐稀缺。

瓶颈迁移的直接后果是构建软件的门槛大幅降低：YC W2025 批次 25% 创业公司 95%+ 代码由 AI 生成，Solo founder 比例从 23.7% 升至 36.3%。但 “原型墙” 普遍存在——AI 消除了 “从零到原型” 的门槛，但分发、运维、合规、留存这些让产品真正运营起来的能力，反而更加凸显。Addy Osmani 提出的 “70% 问题” 精确描述了这一困境：AI 代码看似 70% 正确，但完成剩余 30% 的代价往往超过从头手写。

4.2 构建者群体扩大与 “原型墙”

瓶颈迁移的直接后果是：构建软件的门槛大幅降低。当 “写代码” 不再是硬性门槛时，更多人能够参与构建。

这一趋势已有量级层面的证据。YC W2025 批次 25% 创业公司的代码库 95% 以上由 AI 生成。Solo founder 比例从 23.7% 升至 36.3%。Epic Games 超过 50% 的 Claude Code 使用来自非开发者角色，Block 的非工程师员工自己构建 MCP 服务器。Creator Buddy 以零技术背景实现 $300k ARR；Base44 单人 6 个月被 Wix 以 $80M 收购，这些不是 “玩具” 现象，而是有商业验证的构建者群体扩大。Vibe Coding（凭直觉提示 AI 生成代码的编程方式）市场 2026 年估计规模 47 亿美元，预计 2030 年达 250 亿美元。

构建者扩大 · 关键数据

25%
YC W2025 批次创业公司
95%+ 代码由 AI 生成

36.3%
Solo founder 比例
（从 23.7% 上升）

50%+
Epic Games Claude Code
使用来自非开发者

$80M
Base44 单人 6 个月
被 Wix 收购

$47亿
Vibe Coding 市场
2026 年规模估计

$300k
Creator Buddy
零技术背景实现 ARR

数据来源：TechCrunch (YC W2025, Base44)、One Founder Capital、WIRED、Taskade Vibe Coding 市场报告

“原型墙”：从 “能做” 到 “能用” 的鸿沟

在 vibe coding 社区反复出现一个模式：第一周兴奋（AI 快速生成 MVP）→ 第三周担忧（安全、扩展性、边缘情况浮现）→ 第二月放弃（维护成本超预期）。构建原型 $20/月，维护可能升至 $200/月。Forbes 2026.3 对 Lovable 等 vibe coding 平台留存危机的报道将这一现象命名为 “**Prototype Wall**”（原型墙）。Google Chrome 团队工程师、开发者效能布道师 Addy Osmani 提出的 **“70% 问题”** 精确描述了这一困境：**AI 代码看似 70% 正确，但完成剩余 30% 的代价往往超过从头手写**。GitClear 的 2.11 亿行代码分析也指向同一结论：技术债务增加 30-41%。

原型墙 · 三阶段模式

第 1 周 · 兴奋

AI 快速生成 MVP

成本约 $20/月；看似完成度 70%

↓

第 3 周 · 担忧

安全、扩展性、边缘情况浮现

“70% 问题” 暴露：剩余 30% 的代价可能超过从头手写

↓

第 2 月 · 放弃

维护成本超预期

成本升至 $200/月；技术债务累积 30-41%；项目放弃

数据来源：Forbes《Beyond the Hype: Why 'Vibe Coding' Leaders Are Facing a Retention Crisis》(2026.3)、Addy Osmani《70% Problem》、GitClear 2.11 亿行代码分析

新稀缺在哪

“原型墙” 的本质是瓶颈迁移的产品化表现：AI 消除了 “从零到原型” 的门槛，但 “从原型到可运营产品” 所需要的能力仍然稀缺：**分发**（找到并留住真正付费的用户）、**运维**（7×24 保证可用性、处理边缘场景）、**合规**（安全、隐私、审计、数据治理）、**品味**（选什么做、为谁做、做到什么程度）。

一版报告引用 Vercel CEO Guillermo Rauch 的话：“当机器可以非常廉价地构建东西时，你不再关心 '如何' 完成，而是关心最终结果的 '感觉' 如何。” Replit CEO Amjad Masad 把品味（taste）放在 AI 时代的核心竞争力位置——当生成代码变成商品，**选什么做、为谁做、做到什么程度**成为人保留下来的价值。

这也是第三章 3.3 节 Skills 浮现的深层意义：当 Skills 让非开发者能用 markdown 写出 Agent 能力，他们面临的同样是 “原型墙”。Snyk ToxicSkills 研究扫描 3,984 个公开 Skills，**36% 含安全缺陷、13.4% 含严重缺陷**，这个生态正在重走 10 年前 npm 和 5 年前 VS Code Extension 的老路，OWASP 已推出 Agentic Skills Top 10。

4.3 赛道消融：从 AI Coding 到通用 Agent

当构建门槛向数十亿潜在构建者扩大、Agent 成为工具的主要使用者时，“AI Coding” 作为独立品类的边界自然开始消融。第一版报告判断 “AI Coding 是通用 Agent 的先验战场”，正在被验证。

编程能力本质上就是通用 Agent 的能力栈。读写文件 + 执行命令 + 迭代修复：Claude Code → Agent SDK → Cowork（被 Anthropic 定义为 “面向不写代码的人的 Claude Code”），OpenClaw 将编程 Agent 架构迁移到通用个人助手（160K GitHub 星）。Epic Games 超过 50% 的 Claude Code 使用来自非开发者角色，这条 “Coding → General Agent” 的路径已清晰。

AGENT 产品的边界消融

Agent 入口也在多样化。Claude Code Channels（2026.3）将 Telegram / Discord / iMessage 变成 Agent 遥控器。IDE、终端、手机 IM、Web、桌面应用，多入口汇聚趋势清晰。如果当前趋势延续，“AI Coding” 作为独立品类可能在 2027 年逐步融入更广义的 “AI Agent”。

> *编程能力处于所有其他应用的上游，因为它是任何软件的核心构建块。AI 对编码的加速将加速所有其他领域。*
>
> — a16z 企业 AI 采纳报告，2026.4

05

CHAPTER 05 · 格局与安全

格局与安全

SaaS reallocated. Attack and defense, both accelerated.

SaaS 被重新分配而非消灭；供应链攻击面演化出三个新形态；
AI 同时降低了攻击和防御的门槛。

IN THIS CHAPTER

洞察 ❺

SaaS 没有死去，它正在被重新分配。

过去三个月 Anthropic 至少制造了三场 “Anthropic Day”：2.5 Cowork + Opus 4.6 → FactSet 单日 -10%；2.23 COBOL 博客 → IBM -13.2%（25 年最大）；4.17 Claude Design → Figma -6.89%。受害者有清晰模式——都是 “把 API 包成带收费 UI” 的单功能中间层 SaaS。与此同时，Cursor 估值从 $29.3B 跃至 $50B，2026.4.21 SpaceX/xAI 更以 $60B 收购选项加持，并用 Colossus 百万 H100 等效算力支持 Cursor 训练 Composer 2.5；Anthropic 拿下 Q1 企业 AI 支出 37%，Skills 目录 2,600+ curated——AI 原生基础设施与自建生态同步爆发。被淘汰的不是 SaaS，而是其中 “复杂度封装层” 的那一部分。

5.1 SaaS 的重新分配

三场 “Anthropic Day”。过去三个月里，市场已经形成 “Anthropic 发布 → 某 SaaS 下跌” 的条件反射：

“ANTHROPIC DAY” · 三场定点股价事件

日期	ANTHROPIC 动作	受冲击公司	股价影响
2026.2.5	Cowork 行业插件 + Opus 4.6	FactSet / S&P / Moody's	**FactSet -10%**
2026.2.23	COBOL 现代化博客	IBM	**-13.2%** 25 年最大单日跌幅
2026.4.17	Claude Design 发布	Figma / Adobe	**Figma -6.89%** IPO 以来累计 -80%

数据来源：CNBC、Fortune、Yahoo Finance、Barron's、Bloomberg、Business Insider（2026.2–4）

CNBC 和 Fortune 在 2026.2 同时使用了 “**SaaSpocalypse**” 一词。2026 YTD 软件股大盘（IGV）-22%，Microsoft -21%，Salesforce -26%，Workday -36%，Asana -51%，Figma 自 IPO 累计 -80%——不是全面下跌，而是**结构化下跌**。

内部压力与反应

Business Insider 2026.4 长篇调查呈现了三巨头的真实状态。一位 Microsoft 销售回忆客户 CTO 原话：“我自己就能搭。我为什么还需要你？” Workday CEO Aneel Bhusri 反驳：“AI 正在杀死 SaaS 的叙事被夸大了——Anthropic 和 OpenAI 自己都在用 Workday。再多的 vibe coding 也无法复制我们处理的工资、社保号、跨国合规的复杂度。” 但 Workday 股价 2026 YTD 已 -36%。Asana CEO Dan Rogers 承认：“协调问题不会消失，它会指数级地扩大。” Gartner 2026.2 定性：Cowork 是 “任务级知识工作的潜在颠覆者”，而非 “核心业务 SaaS 的替代者”。

a16z 的反方论点

a16z 在 2026.2.6 博文《Death of Software. Nah.》中明确反对末日论：软件会比以往任何时候都多，但**被打破的是 SaaS 的 lock-in，不是 SaaS 本身**。播客《Is SaaS Dead in a World of AI?》进一步指出：“SaaSpocalypse 是神话；vibe code 一切也是谎言；但 AI agent 正在打破传统 SaaS 的 lock-in。”

真正的图景：价值在重新分配

细看受害名单：设计工具（Figma）、金融数据（FactSet/S&P/Moody's）、遗留系统（IBM）、单一工作流协同（Asana）——**全是 “把 API 包成带收费 UI” 的单功能中间层 SaaS**。同一时期反而加速壮大的：

同时壮大的两极

复杂度承担层（平台）

Cursor $29.3B → $50B → SpaceX $60B 收购选项 · Anthropic 拿下企业 AI 支出 37%

平台承担合规、可靠性、数据治理、嵌入式 Agent 平台等复杂度

极简自建层（SKILLS 生态）

Skills 规范 30+ 工具支持 · Curated 目录一个季度翻倍

非开发者第一次能直接把工作流程自建成可复用的 SOP，绕过中间层

IDC 预测：纯座位（per-seat）计费在 2028 年前作废，行业正在从 “per seat” 向 “per outcome / consumption” 迁移。SaaS 中真正被淘汰的不是 SaaS 本身，而是中间 10 年 “把 API 包成一个带 seat 收费的 UI” 的那一层——**复杂度封装层**。而 SaaS 中仍然稀缺的是**复杂度承担层**。

从“为工具付费”到“为产出付费”。更深层的重塑在计价单位本身：Salesforce Agentforce 按 conversation 计费（约 $2/次），Intercom Fin 按“已解决工单”计费，Zendesk 推出 Resolution Pricing，Anthropic 与部分企业客户试点按“节省工时”折算。当 AI 能交付任务结果而非仅提供工具时，SaaS 的计价单位正从“按座位”向“按产出”迁移——这可能是比定点 Anthropic Day 更深层的重塑力量。

5.2 供应链安全：三种新的攻击面

AI Coding 系统性放大了供应链风险。基础事实：19.7% 的 LLM 推荐包名不存在（幻觉包名攻击 / slopsquatting），43% 的幻觉包名在多次查询中一致重复——**可预测的攻击目标**。MCP 服务器成为新攻击面（30+ CVE）。Snyk 扫描 3,984 个公开 Skills，36.82% 存在安全缺陷。95% 组织使用 AI 工具开发，仅 24% 做全面安全评估。Apiiro 发现 AI 代码每月引入超过 10,000 个新安全发现，6 个月内增长 10 倍。

本报告撰写期间，**三起里程碑级事件在一个月内接连发生**，每一起都各自揭示了一种此前被忽视的新攻击面：

三类新攻击面 · 一个月内三发

事件	攻击入口	攻击目标	特征
LiteLLM 2026.3.24	传统包（PyPI）	AI 工具	AI 工具成为新目标
Axios 2026.3.31	传统包（npm）	传统 JS 生态	旧范式 + AI 加速
Vercel / Context.ai 2026.4.19	**AI 工具的 OAuth**	授权该工具的 所有组织	**AI 工具成为身份劫持跳板**

① LiteLLM 供应链攻击（2026.3.24）——攻击者向流行的 LLM 代理库 LiteLLM 的 PyPI 包注入恶意依赖，劫持运行该代理的节点，将开发者本地调用各家模型时的 API key 与敏感上下文回传至外部服务器。首次证明：**AI 工具本身正在成为攻击目标**——被污染的不是业务代码，而是连接模型与应用的 AI 中间层。

② Axios 独家爆料（2026.3.31）——Axios 独家披露的 npm 供应链事件中，攻击者借 GitHub Actions pipeline 污染构建产物，导致 Claude Code 等多个下游工具的源码级敏感内容泄漏。这是传统 npm 供应链范式的延续，但 AI 显著加速了攻击者的情报收集与漏洞识别——**旧范式 + AI 加速**。

③ Vercel / Context.ai 身份劫持（2026.4.19）——第三方 AI 平台 Context.ai 的 Google Workspace OAuth 应用被入侵，攻击者通过该 OAuth 信任链接管一位 Vercel 员工的工作账户，进而横向渗透枚举读取部分未标记为 “Sensitive” 的生产环境变量。Vercel CEO Guillermo Rauch 在公告中明确指出攻击者 “被 AI 显著加速（significantly accelerated by AI）”。事件扩散到整个 Web3 生态——由于 Vercel 是 Next.js 的主要维护者并承载大量 dApp 前端，Solana DEX Orca 等项目在 24 小时内紧急轮换所有部署凭证。BreachForums 上随后出现卖家叫卖 Vercel 内部数据库与源码，价格 200 万美元。

三件事的共同模式：AI 工具链正成为新的攻击面。三起事件虽入口各异，但共同指向一个事实——企业授予 AI 工具的权限和信任边界，远超过传统对 “一个 npm 包” 的审计强度。AI 工具为了 “读邮件、看文档、管会议” 而普遍拿到宽范围 OAuth；AI 工具为了 “跑模型、调 API” 而普遍持有高权限 token——这些都在悄然打开供应链的前门。

5.3 网络安全新局面：攻防门槛的对称下降

AI 对网络安全的影响远不止供应链风险。2026 年 4 月正在形成一个 AI 网安拐点：**AI 同时大幅降低了攻击和防御的门槛，传统安全格局面临根本性重塑**。

一手证据：Carlini 用 Claude 发现 Linux 内核 23 年漏洞

Anthropic Research Scientist、ML 安全领域代表性研究者（Google Scholar 67,200+ 引用）Nicholas Carlini 在 [un]prompted 2026 AI 安全会议上演示了使用 Claude Code（Opus 4.6）发现 Linux 内核漏洞的过程。方法极其简单：将 Claude Code 指向内核源码，逐个文件扫描。结果是发现了多个远程可利用的堆缓冲区溢出，其中一个藏在 NFS 驱动中长达 23 年（2003 年引入），允许攻击者通过网络读取敏感内核内存。至少 5 个漏洞已被确认修复，还有数百个潜在崩溃尚未来得及验证。

> *我这辈子从没找到过这类漏洞。这非常非常非常难做到。用了这些语言模型，我有一堆。*
>
> — Nicholas Carlini, Anthropic Research Scientist, [un]prompted 2026（2026.3.25）

关键的代际差异：Opus 4.1（8 个月前）和 Sonnet 4.5（6 个月前）只能找到 Opus 4.6 发现的一小部分漏洞。安全能力随模型通用能力呈指数级提升。

Mythos 把这一趋势推到极端

Mythos Preview 发现了涵盖所有主流操作系统和浏览器的数千个零日漏洞——许多存在 10-27 年，经过大量人工审计和数百万次自动化测试都未被发现。标志性案例：OpenBSD 27 年历史的 TCP SACK 漏洞（发现成本不到 **50 美元**）；FFmpeg 16 年历史的解码器缺陷（被 fuzzer 测试 **500 万次**仍未发现）。**Anthropic 的核心判断是：这些安全能力不是专门训练的结果，而是通用能力（代码理解 + 推理 + 自主性）提升的涌现性副产品**。

23年	**<$50**	**500万**	**77%**
NFS 漏洞 存在时长（Carlini）	TCP SACK 漏洞 发现成本（Mythos）	FFmpeg fuzzer 测试 仍未发现	AI agent 网安竞赛 漏洞识别率（前 5%）

双轨降权 + 身份验证准入：行业新范式

2026 年 4 月，前沿实验室的发布策略出现了清晰的结构性转向。

ANTHROPIC · 2026.4.7 → 2026.4.16

Mythos Preview（不公开）→ 9 天 → Opus 4.7 GA（降权公开）

官方承认 “训练期间差异化削弱 cyber 能力”；首推 Cyber Verification Program；Project Glasswing 11 家防御联盟；Anthropic 定位为 “first such model”

OPENAI · 2026.2 → 2026.4.9

Trusted Access for Cyber → 准备限制发布 cyber 专用模型

基于身份验证确保增强网安能力流向正确使用者；承诺 $10M API credits 加速防御；Axios 独家报道后续限制发布模型

两家都采用相同策略：**限制访问 + 防御优先**。“双轨降权发布 + 身份验证准入” 正从 Anthropic 特例变成行业默认。

安全模型的前提假设需要重写

2026 IBM X-Force Threat Index 显示面向应用的公开攻击增长 44%。国际 AI 安全报告（2026.2）指出 AI agent 在一场主要网安竞赛中识别了 77% 的漏洞，排名前 5%。NYT 2026.4.6 专题《AI Is on Its Way to Upending Cybersecurity》标志着这一话题从专业圈进入公众视野。

传统安全依赖 “**摩擦**”（漏洞难以发现、利用需要高技能）而非 “硬障碍” 来保护系统。当 AI 将漏洞发现成本降至 20-50 美元、将利用门槛从 “顶尖研究员” 降至 “任何人 + 一个模型” 时，**整个安全模型的前提假设需要更新**：

旧假设

假设漏洞难以发现

传统安全架构依赖发现门槛高

↓ AI 改变假设

新假设

假设漏洞会被 AI 快速发现

OpenAI 和 Anthropic 选择限制发布本身就是对这一新现实的承认

5.2 节中描述的防线建设（AIBOM、PR 级安全扫描、幻觉包检测、OAuth Surface Audit 等）将从 “最佳实践” 加速成为合规要求。但前沿实验室的主动降权仍无法解决 5.2 节 Vercel 事件揭示的那一类风险——**分布式 AI 工具生态本身仍在放大供应链入口**。前沿模型在降权，但小型 AI 工具的 OAuth 权限面在扩大——这两个趋势并行发生，共同刻画了 2026 年的安全新格局。

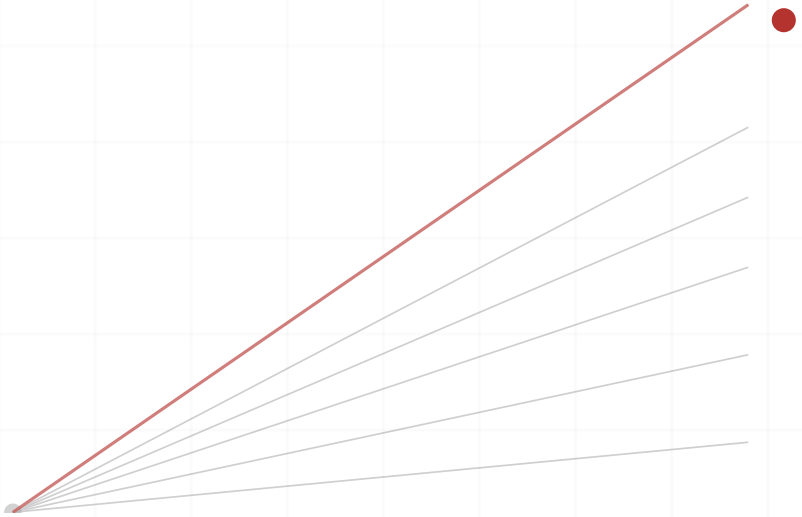
06

CHAPTER 06 · 面向未来

面向未来

Developer redefined, on both ends.

做什么在变——编写者转向编排者；谁能做也在变——非开发者真正入场。就业在三层之间流动。



IN THIS CHAPTER

洞察 ❻

做什么和谁能做，开发者被双向重定义。

“做什么” 在变：开发者从 “编写者” 转为 “编排者”，Staff+ 工程师 63.5% 是最重度 Agent 用户，判断力与系统理解力成核心技能；就业结构三层同步流动。“谁能做” 也在变：非开发者正在以 “构建者” 身份进入：Epic Games 超 50% Claude Code 使用来自非开发者、Block 非工程师员工自己构建 MCP 服务器、Skills 让一个人用 markdown 在 10 分钟内就能写出第一个 Agent 能力。“一人公司” 不再是边缘现象。

6.1 开发者角色转型

角色变化有具体含义。在旧范式中，开发者的主要工作是理解需求、编写实现、调试修复。在新范式中，编写实现被 Agent 大幅承担，开发者的时间分配正在向上游迁移：定义规格、设计约束、评估 Agent 产出、管理多 Agent 协作。Anthropic CEO Dario Amodei 在达沃斯 2026 的表述是：“工程师编辑 Claude 的产出，而非自己写代码。”

往下走 / 留在中间 / 往上走

Replit CEO Amjad Masad 在第一版报告中提出的三象限框架，9 个月后正被更多证据支持：

三个方向的开发者角色

往下走 · 安全

系统编程、嵌入式、安全关键

越接近底层，越难被替代

留在中间 · 危险

全栈开发、CRUD 应用、标准框架

GitHub 最常见 = AI 最擅长

往上走 · 机会

产品设计、用户体验、商业策略

越接近用户，越需要创造

这三个方向的岗位变化在数据上已经出现分化：Indeed 数据显示软件开发岗位需求仅为 2020.1 水平的 65%，初级岗位占比 30%→20%，高级岗位（7 年+经验）占开放职位近 40%。而 LinkedIn 数据显示驾驭工程相关岗位发布量增长 250%，**新岗位主要在上下两端，中层在压缩**。

教育体系已经开始响应

Stanford 开设 “The Modern Software Developer” 课程，鼓励学生 “如果能在整个课程中不写一行代码，那更好”。MIT 开设 “No Code and Agentic AI” 专业课。Code.org 新 CEO 表示计算机科学教育将 “看起来非常不同，理解技术如何工作比写代码本身更重要”。CS 招生方面，62% 美国大学报告 CS 招生下降，但学生涌向 AI 专业，**学生在用脚投票**。

6.2 非开发类构建者真正入场

与开发者角色转型同时发生的，是 “开发者” 定义本身的扩大。

非开发者以 “构建者” 身份成为软件生产的正式参与者。Epic Games 超过 50% 的 Claude Code 使用来自非开发者。Block 的非工程师员工自己构建 MCP 服务器。Bolt.new 60-70% 的用户不是传统开发者，典型用户包括设计师、学生、健身教练、销售人员、教育企业家、老师。这一群体在过去是 “不会写代码所以用 SaaS”，现在是 “用 AI 让他们能写代码 / 规范能力”。

Skills 把这一扩大推到了能力制造端。第三章 3.3 节介绍的 Skills 标准让一个非开发者能在 10 分钟内用 markdown 写出第一个 Skill，不用写一行代码。Anthropic 的 Skilljar 官方课程、Snyk 面向 solopreneur 的 Top 8 Skills 专题、Axton Liu 一个人为自己的内容 / 视频 / 商业分析工作构建 80+ Skills，这些都指向一个事实：**非开发者不再只是 AI 工具的使用者，而真正成为 Agent 能力的作者**。

“一人公司”从边缘现象变为主动选择。独立开发者与 solo founder 正迅速从少数选项变为显性趋势：YC 批次中 solo founder 的比例从一年前的 23.7% 升至 W2025 的 36.3%。标志性案例：Ma Or Gomberg 一人用 6 个月把 Base44 从零做到 $80M 被 Wix 收购（2025.6）；Pieter Levels（Levelsio）以 10+ 个产品组合维持 “indie empire” 的商业模式被广泛模仿。“一人公司”因此不再只是“还没找到联合创始人”的过渡态，而是一种**主动选择的组织形态**——AI 让一个人能承担过去 5-10 人团队的产出，产品、设计、运维、客服、营销都交给 Agent 组协同完成。

但非开发者面临的 “原型墙” 比专业开发者更陡。缺乏工程纪律的 “一人公司” 维护成本可能 4 倍增长，技术债务的积累对小团队更为致命。Skills 生态的 36% 安全缺陷率（Snyk ToxicSkills）也提醒：当能力制造权下放时，治理和审计能力同步稀缺。下一步的机会在 “**非开发者的驾驭工程**”，为 non-technical 构建者提供与专业开发者同等强度的验证、监控、维护工具。

6.3 就业结构重组：从分层到流动

真实图景比 “AI 取代程序员” 复杂。2026 年初 20.4% 科技裁员归因 AI。HBR 对 1,006 名管理者的调查显示企业因 AI 的 “潜力” 而非 “实绩” 裁员。BBC 中文报道了三个中国人从 “排队装龙虾” 到 “排队卸载” 仅 40 天的经历——就业冲击是真实的。

变化的本质是结构性的。三层同步流动：

就业结构三层流动

层级	岗位特征	变化
高层	架构、产品定义、判断力	↑ 价值上升（30%→40%）
中层	管理 Agent 的技术项目经理	★ 新增岗位（LinkedIn +250%）
底层	初级编码任务	↓ 被压缩（30%→20%）

Staff+ 工程师 63.5% 是最重度 Agent 用户；数据来源：Pragmatic Engineer (906 人) / Indeed / LinkedIn

Staff+ 工程师（指 Staff Engineer 及更高级别，通常对应互联网公司的资深 / 技术专家 / 架构师层级）是最重度 Agent 用户（63.5%）——**最有经验的人受益最多**，因为他们的判断力和系统理解力正是驾驭工程所需的核心技能。

“ARR 规则被改写” 是变化的另一面。Cursor 0→1 亿 ARR 用了 21 个月 20 人；Bolt.new 0→2,000 万 ARR 用了 2 个月 15 人；Lovable 0→1,000 万 ARR 用了 2 个月 15 人。Cursor 在 20 人 1 亿 ARR 时没有传统销售、市场、HR 部门——这不是人才浪费，而是劳动力的流动性全面提升后，对 “组织” 概念本身的重新定义。**“10 人做 100 人的事” 从预言变成了常态**。

6.4 展望

回顾本报告的六个洞察，几条趋势方向已经清晰：

Agent-First 将成为默认开发环境。Cursor 3.0、Google Antigravity、VSCode Agent Mode、Codex App 全面铺开。12 个月内，“没有 Agent 协作” 的开发环境将像 “没有版本控制” 一样不可想象。

Coding Agent 与 General Agent 的边界将持续消融。Claude Code → Cowork → Agent SDK → Channels 的路径已清晰。当 Agent 能力栈（代码 + 文件 I/O + 命令执行）天然等于通用 Agent 能力栈时，品类融合是逻辑必然。

前沿能力的释放方式将持续重塑安全格局。我们将看到更多 “Mythos → Opus 4.7” 式的降权路径，更多 Project Glasswing 式的准入联盟，但现实中 AI 工具本身的身份劫持风险（Vercel 事件）会继续放大，直到 AIBOM / OAuth Audit 等新基础设施普及。

“原型墙” 将驱动下一轮工具创新。当前 AI Coding 的主要价值在 “从零到原型”。下一个价值增量在 “从原型到产品”：即验证、分发、运维、合规工具。掌握驾驭工程的 “一人 + Agent 团队” 将存活，只依赖 vibe coding 的项目将面临维护危机。

“开发者” 的定义将继续扩大。从 2,500 万专业开发者到数亿会说英语的构建者，这一扩展的步伐不会放缓。但随之而来的是对治理、审计、可靠性这些新基础设施的刚性需求，这决定了 “一人公司” 是否真的能跑起来。

> *丰饶之后，稀缺并未消失——它迁移了。*
> *代码不再稀缺，稀缺的是判断力、验证能力、品味，以及把这一切持续运营下去的工程纪律。*
> *这是下一段竞争的起点。*

附录

附录 A：关键时间线

2024.6
Claude 3.5 Sonnet 发布（第一道能力门槛）

2025.2
Claude Code 发布

2025.7
第一版《AI Coding 非共识报告》发布；METR RCT 首次实验

2025.10
Anthropic 发布 Agent Skills 机制

2025.11
Opus 4.5 发布（第二道能力门槛）；Google Antigravity；Dex Horthy 首次系统阐述 “harness engineering”

2025.11–12
Cursor $293 亿估值；Claude Code $10 亿收入

2026.1
Kimi K2.5 + Agent Swarm

2026.2
Opus 4.6 / Sonnet 4.6 发布；Codex App 发布（macOS）；OpenAI Harness Engineering 博文

2026.2
METR 结论逆转；Anthropic Series G $300 亿；OpenAI Trusted Access for Cyber

2026.3
Cursor 谈 $500 亿；Claude Code Channels；Skills 目录 2,600+ curated；a16z《Death of Software. Nah.》

2026.3.23-30
网易云音乐 / 钉钉 / 飞书 / 企业微信集体开源 CLI

2026.3.24
LiteLLM 供应链攻击

2026.3.25
Nicholas Carlini “Black-hat LLMs” 演讲

2026.3.31
Axios 供应链攻击；Claude Code 源码泄漏

2026.4.2
Cursor 3.0 发布

2026.4.7
Claude Mythos Preview + Project Glasswing 联盟

2026.4.16
Claude Opus 4.7 GA（Mythos 降权公开，首次打破商业趋同）

2026.4.17
Claude Design 发布，Figma 单日 -6.89%

2026.4.19
Vercel / Context.ai 供应链事件

2026.4.21
Codex App周活突破400万；Cursor x SpaceXAI 算力合作

…

附录 B：术语表

术语	定义
驾驭工程 / Harness Engineering	设计让 AI Agent 可靠工作的系统：工作流、规格、约束、记忆、反馈循环
MCP / CLI	Model Context Protocol：Anthropic 主导的 AI-工具通信协议；CLI：命令行接口，在 AI 语境下指 Claude Code / Codex CLI 等终端 Agent 工具
Skills / Agent Skills	Agent 可复用的能力封装：markdown SOP + 可选脚本，通过渐进式披露按需加载
SWE-bench Verified / Pro	基于真实 GitHub issue 的代码修复基准；Verified 为人工筛选版，Pro 为更难版本
Agent / Subagent / Agent Teams	自主执行任务的 AI 实体 / 受主 Agent 调用的子代理 / 多 Agent 协作模式
METR / RCT	METR：独立 AI 能力评估组织；RCT：随机对照实验
slopsquatting	幻觉包名攻击：攻击者抢注 LLM 幻觉推荐的不存在包名
SCA / SBOM / AIBOM	软件成分分析 / 软件物料清单 / AI 物料清单，分别追溯依赖、软件、AI 系统来源
Dogfooding	“吃自家狗粮”：开发者使用自己开发的产品来验证可用性
原型墙 / Prototype Wall	AI 生成的 MVP 在走向可运营产品时遇到的持续性鸿沟
Vibe Coding	凭直觉提示 AI 生成代码的编程方式
ARR	Annual Recurring Revenue，年度经常性收入，SaaS 行业核心营收指标

附录 C：参考文献与信源说明

本报告采用文中自然标注方式引用信源，以下为主要信源列表，按类别整理。

实验室与技术报告

Anthropic — Opus 4.5/4.6/4.7 发布博文、Mythos Preview / Project Glasswing、Agent Skills、Agent Teams、Harness design for long-running apps
OpenAI — Harness Engineering 博文、Codex App 发布、Trusted Access for Cyber
METR — RCT 首次实验（2025 初完成，2025.7 发布）、后续更新（2026.2）
SWE-bench — Verified / Pro 排行榜
Vellum AI — Opus 4.7 第三方基准核对
KTH 皇家理工学院 — AI Agent 自举实验 (arXiv 2603.17399)
腾讯研究院 —《AI Coding 非共识报告》(2025.7.24)

行业调研与市场数据

a16z — 企业 AI 采纳报告 (2026.4.8)、Death of Software. Nah. (2026.2.6)
Pragmatic Engineer — 906 人开发者工具调查 (2026.1-2)
Retool — AI Build vs Buy 报告 (2026.2)
GitClear — 2.11 亿行代码质量分析；Veracode — AI 代码安全报告
HBR — 管理者 AI 裁员调查 (1,006 人, 2026.1)
IBM — X-Force Threat Index 2026；国际 AI 安全报告 (2026.2)
IDC — per-seat 计费 2028 前作废预测

媒体报道

WIRED — Boris Cherny 专访；Forbes — 回归自建、Vibe Coding 留存危机
NYT — “AI Is on Its Way to Upending Cybersecurity” (2026.4.6)
Axios — OpenAI 网安独家 (2026.4.9)、Anthropic Opus 4.7
Bloomberg — Cursor $50 亿估值；Business Insider — Inside Big Software's fight for its life (2026.4.7)
CNBC — IBM AI casualty、SaaSpocalypse (2026.2)
Yahoo Finance — Claude Design / Figma (2026.4.17)
Fortune — Trillion-dollar selloff (2026.2.6)
VentureBeat — Cowork、Claude Code 源码泄漏；TechCrunch — YC W2025、Base44 收购
BBC 中文 — AI 就业影响报道 (2026.3)；The New Stack — Block 12,000 员工采用；The Verge — Opus 4.7 cybersecurity

安全研究

Nicholas Carlini — “Black-hat LLMs”, [un]prompted 2026 (2026.3.25)
Datadog — LiteLLM 供应链攻击分析 (2026.3.24)
CrowdStrike — Agentic Tool Chain 攻击模式 (2026.1)
Snyk — ToxicSkills 扫描报告 (2026.2)；Spracklen et al. — 幻觉包名攻击研究 (USENIX Security 2025)
Vercel — April 2026 Security Incident Bulletin (2026.4.20)；CoinDesk — Web3 集体轮换密钥 (2026.4.20)

Skills 生态

Anthropic Engineering — Equipping agents for the real world with Agent Skills (2025.10.16)
IntuitionLabs — Claude Skills vs. MCP: A Technical Comparison (2026.2.9)
Milvus / Zilliz — MCP vs CLI vs Agent Skills Compared (2026.4.1)
noqta.tn — SKILL.md Open Standard 30+ Tools (2026.4.11)

开发者与业界声音

HumanLayer CEO Dex Horthy — “harness engineering” 首次系统阐述 (2025.11)
HashiCorp 联合创始人 Mitchell Hashimoto — My AI Adoption Journey (2026.2.5)
ThoughtWorks Martin Fowler / Birgitta Böckeler — Harness Engineering 正典化 (2026.4.2)
Andrej Karpathy — Opus 4.5 评价、Software 1.0/2.0/3.0 YC 演讲
Anthropic CEO Dario Amodei — 达沃斯 2026 发言
Google Chrome Addy Osmani — “70% 问题”；Lean 创始人 Leo de Moura — 验证、测试和规格是瓶颈
Replit CEO Amjad Masad — 往下走/留在中间/往上走 (V1 引用)
Vercel CEO Guillermo Rauch — Prototype Wall + 攻击 AI 加速声明；Cloudflare — Code Mode 技术博客

丰饶之后，稀缺并未消失，
它迁移了。

After the abundance, what remains scarce is judgment, verification, taste, and the discipline to keep it running.

研究团队

曹士圯 · 余一 · 袁晓辉

& CodeBuddy

腾讯研究院

AI 透镜系列研究

2026 年 4 月