传媒行业深度报告

AI 编程：最有用、最愿付费、增长最快

2025 年 08 月 12 日

增持（维持）

关键词：#新产品、新技术、新客户

证券分析师 张良卫
执业证书：S0600516070001
021-60199793
zhanglw@dwzq.com.cn
证券分析师 周良玖
执业证书：S0600517110002
021-60199793
zhoulj@dwzq.com.cn
证券分析师 张文雨
执业证书：S0600525070007
zhangwy@dwzq.com.cn

行业走势

相关研究

《海外为鉴，研判 AI 应用产业趋势》
2025-07-27
《短剧出海，不止于“奈飞平替”》
2025-07-16

投资要点

- **AI 编程是当前人工智能领域最有用、用户最愿意付费且增长最快的应用方向之一。它并非简单的提效工具，而是重塑软件生产关系的新基建。软件正在吞噬世界，而 AI 正在吞噬软件，AI 编程正是这个吞噬过程的牙齿。** **1）最有用：** AI 编程直接作用于数字世界的核心生产活动，致力于解决“无限的软件需求”与“有限的开发者供给”这一根本矛盾。它通过赋能现有开发者并降低编程门槛，极大地放大了生产力。任何对软件开发效率的提升，都会通过项目的提前上线或关键漏洞的快速修复，产生被成倍放大的商业价值。**2）最愿付费：** AI 编程工具的 ROI 对企业和个人都清晰可见。企业能通过提升高薪工程师的效率在数日内收回成本，而开发者也愿意为提升个人核心竞争力付费。它不是 kill time，而是 save time。它为企业/个人带来的价值增加，直接决定了用户的付费意愿。此外，AI 编程任务的 Token 消耗量远超传统聊天应用，一个活跃开发者日均消耗可达数百万甚至千万级别，这直接驱动了底层模型厂商的 API 收入，形成了稳固的商业模式。**3）增长最快：** 底层模型的持续进步直接提升产品体验，而领先应用已开始利用宝贵的交互数据反哺模型优化，形成“模型-产品-用户-数据”的正向循环。同时，开发者社区的口碑效应结合产品驱动增长（PLG）模式，实现了高效的病毒式传播。**4）一级市场已经用真金白银投票：** Cursor 的 ARR 达 5 亿美元（2025/06），Claude Code ARR 达 4 亿美元（2025/07），Devin 的 ARR 0.7~0.8 亿美元（2025/07）。不仅 ARR，估值也在水涨船高。我们判断，AI 编程赛道正处爆发早期，值得长期关注。

- **基于此，我们看到了两重市场机遇：1）存量市场：** 针对全球近 3000 万专业开发者的“AI 化升级”，构成了一个付费意愿强、价值密度高的基础市场。其长期潜在市场规模（TAM）可达约 115 亿美元。**2）增量市场：** 更具想象力的部分在于通过“代码平权”赋能数以亿计的“泛开发者”（如产品经理、分析师等）。当 AI 将软件开发的成本和门槛降至极低时，大量过去因成本过高而被压抑的个性化软件需求将被释放，催生出一个远超存量市场的庞大衍生经济。我们测算，增量市场的潜在规模高达 150 亿美元（2030 年）。3）更深远的是，AI 编程能力是未来 AI Agent 的底层基础设施。AI Agent 执行任务（如调用 API、处理数据）的本质是一系列编程任务。因此，AI 编程的成熟将是解锁自主 AI 智能体的关键，其影响是指数级的。

- **AI 编程的发展路径可分为四个阶段：** 从早期的探索，到已成功商业化的 Copilot（辅助驾驶）模式，再到追求更高自主性的 Agent 模式，最终迈向能独立完成软件开发的 Autopilot（自主驾驶）模式。当前，Copilot 通过代码补全等功能显著提升了开发效率，而 Autopilot 是行业前沿的终极目标，但面临产出结果不稳定的挑战。技术上的核心瓶颈已从模型的长文本处理能力，转变为对大型复杂项目的“上下文管理”能力。这要求 AI 不仅能看到代码，更能理解整个代码库的架构、依赖关系和开发者的深层意图，这是实现更高阶智能的关键。未来的核心竞争壁垒在于私有的“过程数据”。这些数据反映了开发者与 AI 交互的真实工作流，是训练更智能模型的宝贵养料，其价值远超公开代码。能有效利用过程数据以优化模型、并融入开发者工作流形成使用习惯的产品，将构建起最强的竞争护城河。

- **竞争格局多元化，主要有四类参与者。1）VS Code Fork 系：** 以 Cursor 为代表，通过深度改造 VS Code 打造 AI 原生 IDE，以极致的集成体验快速获取用户。其挑战在于需要持续投入资源跟进 VS Code 的更新，并且面临固定订阅收入与按量计费的模型支出不匹配的商业困境。**2）小**

白平台：以 Replit 为代表，定位为从创意到部署的一站式平台。它们将 AI 代码生成作为获客的流量入口，真正的利润来源于后端的基础设施服务（如托管、部署）。这种模式通过提供全流程解决方案，建立了更深的护城河。**3）Agent 探索者与务实派**：以 Devin 为代表的探索者，致力于实现完全自主的“AI 软件工程师”，虽然愿景宏大，但当前在真实任务中的成功率有限，正从高预期向更务实的人机协作模式调整。而字节跳动的 Trae 则选择更务实的道路，聚焦解决当前开发者在工具间切换、上下文割裂的痛点，旨在通过优化当下体验和探索下一代工作流来赢得用户。**4）巨头与中国力量：**谷歌、Anthropic 等巨头正通过“模型即产品”的策略直接入场，对其产品（如 Gemini CLI、Claude Code）的持续迭代对应用层创业公司构成降维打击。与此同时，以 Qwen、Kimi 为代表的中国模型性能已追至世界前沿，并采取激进的开源策略，旨在成为 AI 时代的“Android”系统，构建全球技术生态。特别是 Kimi 凭借其行业领先的长文本处理能力，直击 AI 编程的核心瓶颈——上下文管理，展现出强大的竞争力。

- **投资建议：短期关注入口级应用，中期聚焦平台级工作台，长期布局行业生态。**短期内，关注能解决特定痛点、产品体验极致的“killing app”。这类公司凭借出色的产品力能快速获取用户，商业模式清晰，且是巨头青睐的收购对象。中期来看，随着市场整合，单纯的工具将面临增长瓶颈。投资机会在于那些能整合工具链、提供一站式解决方案的“超级工作台”。这类平台通过构建开发者难以迁移的工作流，能建立起强大的用户粘性和数据护城河，具备成长为行业巨头的潜力。长期展望，当通用 AI 编程能力商品化后，最高价值将体现在具体的行业应用中。此时的投资重点应转向：1）将 AI 编程与行业知识深度结合的垂直领域冠军；2）为开源生态提供企业级服务的供应商；3）产业链上游的“卖铲人”，即顶尖 AI 芯片公司和大模型厂商。
- **风险提示：**技术路径与实现风险，商业模式与盈利风险，市场竞争与生态风险。

内容目录

图表目录

1. AI 编程：最有用、最愿意付费、增长最快

编程是 AI 最有用、用户最愿意付费、增长最快的应用方向之一。这也是为什么，编程能率先突围，成为大厂与资本重兵布局的赛道。

软件正在吞噬世界，而 AI 正在吞噬软件。AI 编程，就是这个吞噬过程的牙齿。今天我们讨论的 AI 编程，可能只是未来 AGI 在软件工程领域的早期体现。

我们认为，AI 编程是当前人工智能领域中最具确定性的高增长赛道。它并非简单的“提效工具”，而是重塑软件生产关系的“新基建”，其核心投资逻辑在于：1）直击数字经济核心生产环节，具备极强杠杆效应；2) 商业模式已得到市场验证，企业与个人用户均展现出极高付费意愿；3) 依托开发者社区与产品驱动增长（PLG）模式，内生增长飞轮效应显著。

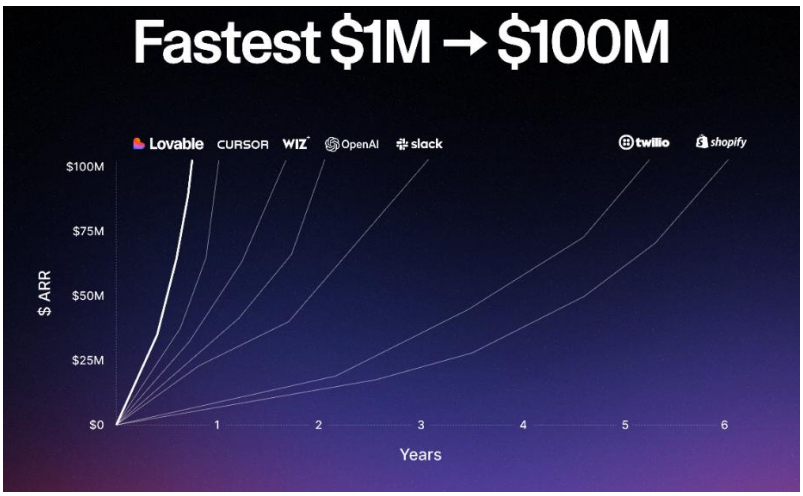
以 Cursor、GitHub Copilot 为代表的头部公司，其惊人的收入增长曲线初步验证了行业的爆发潜力。我们判断，AI 编程赛道正处于 S 型增长曲线的陡峭爬升阶段，未来市场空间广阔。

表1：主要 AI 编程产品的 ARR 情况

产品	ARR
Cursor	2025 年 6 月突破 5 亿美元
GitHub Copilot	2024 年 12 月约 4 亿美元
Claude Code	2025 年 6 月突破 4 亿美元
Lovable	2025 年 7 月突破 1 亿美元
Devin	2025 年 7 月约 0.7~0.8 亿美元
Windsurf	2025 年 4 月约 1 亿美元

数据来源：cursor，techcrunch，arr club，lovable，东吴证券研究所

图1：Lovable 和 Cursor 的 ARR 增长曲线十分陡峭



数据来源：Lovable 官网，东吴证券研究所

1.1. 为何最有用：直击数字世界的生产力核心

在所有 AI 应用场景中，编程之所以最有用，是因为它直接作用于数字世界最核心的生产活动，具备杠杆效应。

纯粹的生产力工具：不同于 AI 绘画、AI 音乐或通用聊天，AI 编程工具的价值主张极其明确：提升软件生产的效率和质量。它不是用来 kill time，而是用来 save time。

解决核心供需矛盾：现代社会对软件的需求是无穷无尽的，从企业数字化转型到个人生活便利，都需要软件来驱动。但专业的软件工程师供给却是有限且昂贵的。AI 编程工具，正是解决“无限软件需求”与“有限开发者供给”矛盾的关键。它通过赋能现有开发者、降低编程门槛，提升了软件的供给能力。

强杠杆效应：提升一个开发者的效率，其产生的价值远不止于节省了他个人的时间。一个功能的提前上线，可能意味着公司抓住了一个关键的市场窗口；一个 bug 的快速修复，可能挽回了巨大的商业损失。软件开发是典型的高杠杆活动，因此，任何能提升其效率的工具，其“有用性”都会被这个杠杆成倍放大。

图2：开发者对 AI Coding 的诉求

数据来源：founder park，东吴证券研究所

1.2. 为何最愿付费：清晰的 ROI 与 Token 消耗跃迁

付费意愿与价值感知直接挂钩。AI 编程工具之所以能让用户心甘情愿地付费，是因为它的 ROI 清晰、直接且可观。

第一，对企业而言，这是一笔稳赚不赔的投资。一个优秀的软件工程师，其年薪、福利等综合成本很高（例如数十万美元/年）。如果一款每月花费 20-50 美元的 AI 工具，能让他的工作效率提升 10%-20%（这在实践中是非常保守的估计），那么公司为此付出的成本在几天之内就能完全收回，剩余的都是纯利润。

第二，对开发者个人而言，这是提升核心竞争力的装备。对于追求卓越的开发者来说，AI 编程工具就像是士兵的先进武器、赛车手的顶级引擎。它能帮助他们更快地学习新技术、更高质量地完成工作、从繁琐的体力活中解放出来，专注于更具创造性的架构设计和业务逻辑。这种对个人核心竞争力的直接提升，使得开发者有购买意愿。

第三，它切中了行业最深的痛点。编程过程中充满了大量重复、繁琐且易错的工作，例如：编写样板代码、配置环境、调试、写测试用例等。这些是开发者长期以来的痛点。AI 编程工具精准地解决了这些问题，这直接转化为了强烈的付费意愿。

第四，需求模式的跃迁：从以聊天为代表的对话式 AI，到以编程为代表的工作式 AI，这不仅仅是应用场景的切换，更是对底层算力需求模式的一次数量级跃升。一个普通用户使用聊天 App，其交互是低频、轻量的，单次交互的 Token 消耗通常在几百到几千的级别。而一个开发者使用 AI 编程工具，其工作模式完全不同：交互是高频且持续的；上下文是极其庞大的，AI 需要加载整个项目的代码库、依赖、配置等，初始上下文加载轻易就能达到数十万甚至上百万 Token；任务是连续的，从规划到编码再到测试，整个工作流下来，Token 消耗量达到百万级别是家常便饭。

一个看似简单的任务，其 Token 消耗拆解可能如下：

上下文读取：AI 一次性读取项目核心代码（假设 50 万 Tokens）。

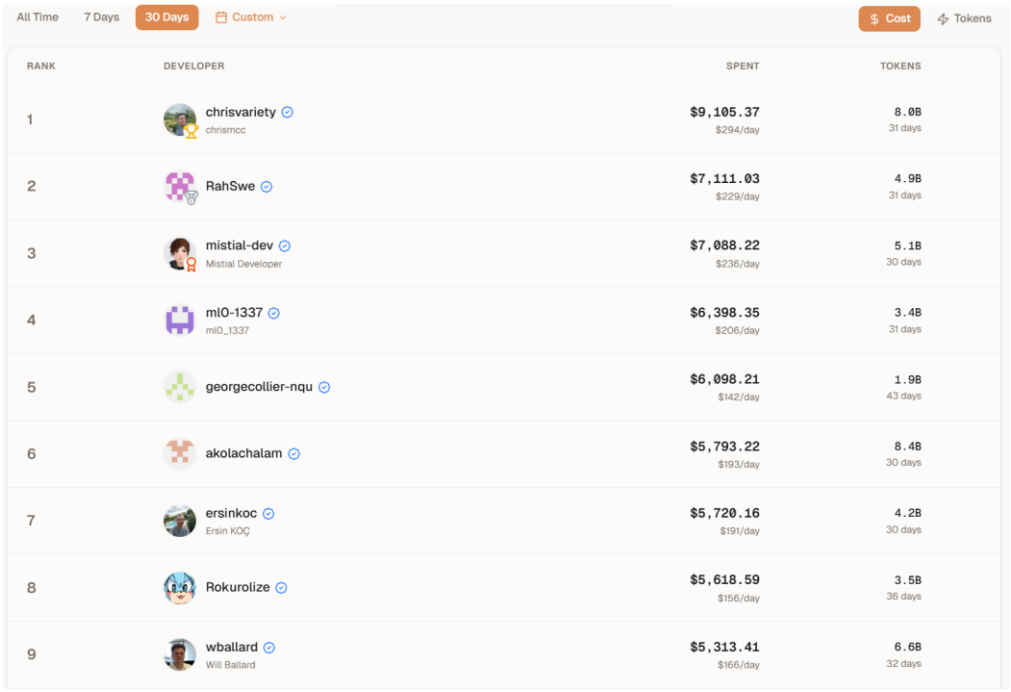
规划与交互：多轮内部规划和工具调用（假设产生 10 万 Tokens）。

代码生成与修改：初版代码生成及后续多轮修改（假设总计消耗 45 万 Tokens）。

总消耗：完成此任务的总 Token 消耗可能达到 **105 万 Tokens**。

一个活跃的开发者日均 Token 消耗量可轻松达到数百万甚至千万级别，是传统聊天机器人用户的数百倍甚至上千倍。假设未来 3-5 年 AI 编程在约 5400 万活跃用户（专业+泛开发者）中达到 30%的渗透率，人均日消耗 200 万 Tokens，每日总消耗将达到 108 万亿 Tokens。以目前每百万 Tokens 1-3 美元的 API 定价计算，模型方可以获得的收入规模在 394 亿美元~1182 亿美元。

图3：Claude Code 上的“大 R 用户”一天能花掉几百美元



数据来源：Claude Code Leaderboard，东吴证券研究所

注：统计区间为 2025/7/8-8/8 的一个月

来自头部模型公司的真实业绩，印证了这一判断。 根据 Moonpig AI 负责人 Peter Gostev 在 2025 年的分享，Anthropic 的 ARR 在 7 个月内从 10 亿美元飙升至 50 亿美元，其增长主要由 API 收入驱动。尤为关键的是，其 API 收入的核心来源正是 AI 编程，两大编程工具客户 Cursor 和 GitHub Copilot 就为其贡献了高达 14 亿美元的收入。其自有产品 Claude Code 的 ARR 也高达 4 亿美元。这证明了，AI 编程应用已成为驱动基础模型层商业价值的核心引擎之一，其对算力和 Tokens 的消耗拉动效应是显著的。

1.3. 为何增长最快：“模型-产品-用户-数据”的飞轮效应

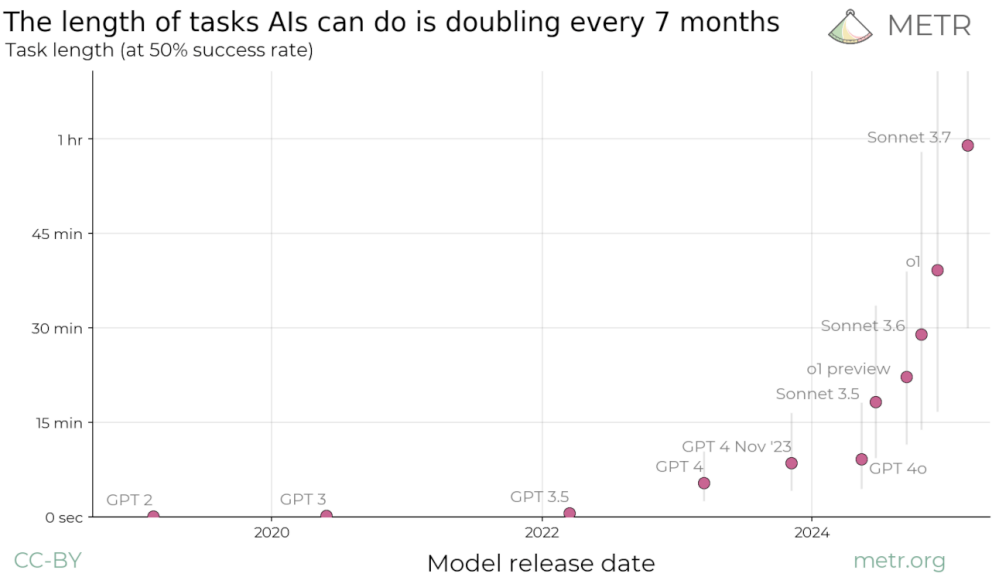
AI 编程产品的能力迭代很快，形成了“模型-产品-用户-数据”的飞轮效应。

图4：AI 编程产品的迭代飞轮：模型-产品-用户-数据

数据来源：东吴证券研究所

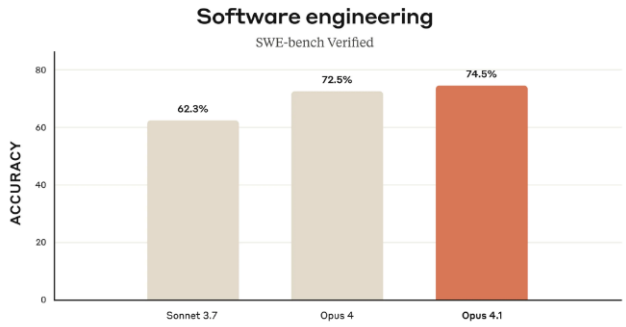
第一，AI 编程是基础模型能力最先成熟的应用场域。模型在编程、推理和长上下文处理能力上的进步，会直接让 AI 编程工具变得更加好用。在 2023 年初，当大模型的能力尚处于发展早期，普遍存在笨拙和不够普适的问题。对于非技术领域的用户而言，这种不完善可能会导致使用体验差，难以被广泛接受。然而，编程领域的从业者，即程序员，本身就具备深厚的技术背景和专业知识。他们能够理解 AI 工具的局限性，并利用自身的技能去弥补 AI 的不足，例如通过更精准的 Prompt 引导 AI，或对 AI 生成的代码进行调试和优化。这种“人机协作”的模式，使得 AI 工具的缺陷在编程领域变得更可接受，甚至能通过技术共振产生协同效应。

图5：模型可以处理越来越复杂和费时的任务



数据来源：replit，东吴证券研究所

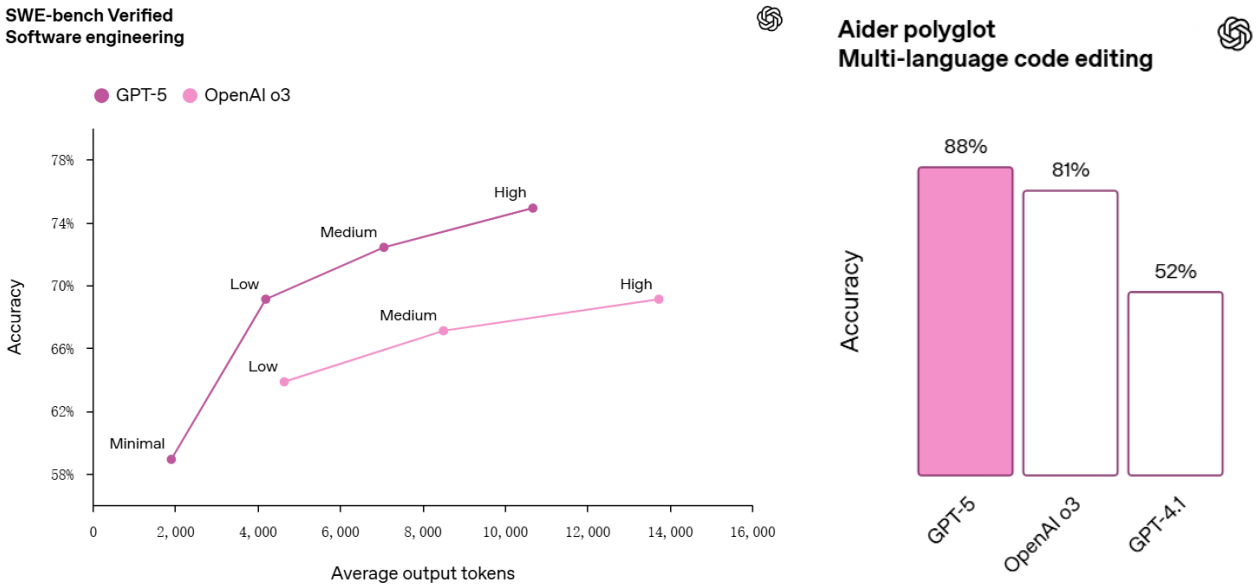
图6：Claude 模型家族的编程能力持续提升



	Claude Opus 4.1	Claude Opus 4	Claude Sonnet 4	OpenAI o3	Gemini 2.5 Pro
Agentic coding SWE-bench Verified[1]	74.5%	72.5%	72.7%	69.1%	67.2%
Agentic terminal coding Terminal-Bench[2]	43.3%	39.2%	35.5%	30.2%	25.3%
Graduate-level reasoning GPQA Diamond	80.9%	79.6%	75.4%	83.3%	86.4%
Agentic tool use TAU-bench	Retail 82.4% Airline 56.0%	Retail 81.4% Airline 59.6%	Retail 80.5% Airline 60.0%	Retail 70.4% Airline 52.0%	– –
Multilingual Q&A MMMLU[3]	89.5%	88.8%	86.5%	88.8%	–
Visual reasoning MMMU (validation)	77.1%	76.5%	74.4%	82.9%	82%
High school math competition AIME 2025[4]	78.0%	75.5%	70.5%	88.9%	88%

数据来源：机器之心，东吴证券研究所

图7：GPT-5 的编程能力超越 o3 和 GPT-4.1



数据来源：OpenAI，东吴证券研究所

第二，数据飞轮效应。因为程序员用户能够容忍 AI 的早期不足并积极使用，这为产品积累了大量的用户行为数据和代码生成数据。这些数据反过来又可以用于持续优化 AI 模型和产品功能，形成一个正向的“数据飞轮”，推动产品和技术的快速迭代与进步。这种早期用户群体的特性，使得 AI 编程成为基础模型能力落地并快速验证的理想场景。

第三，代码是高质量的训练语料，编程是易于验证的任务类型。1）与自然语言相比，代码是一种高度结构化、逻辑严谨且规则明确的语言。每一行代码都有其特定的语法和语义，错误会直接导致程序无法运行或产生预期之外的结果。这种强约束性使得代

码成为训练大模型推理能力和逻辑能力极其优质的语料。模型可以通过学习大量的代码，更好地理解程序逻辑、数据结构以及算法，从而提升自身的编程能力和问题解决能力。**2）**代码能否运行、测试用例能否通过，都能提供非常明确的奖励信号，这使得强化学习等 AI 训练方法在编程领域非常适用。

第四，AI 编程的用户群体是天然的传播 KOL： AI 编程工具普遍采用“先试用，后付费”的产品驱动增长（PLG）模式。开发者可以轻松上手，亲身体验到效率提升的快感。一旦形成了使用习惯和依赖，付费转化就水到渠成。这种“自下而上”的增长模式，无需昂贵的市场推广。此外，开发者是一个高度聚集、乐于分享、崇尚技术的群体。他们在 GitHub、Stack Overflow、Twitter(X)、Reddit 等社区高度活跃。一款真正好用的工具，会通过口碑效应，在这些社区里实现“病毒式”传播。一个“Wow Moment”（比如 AI 一秒生成一个复杂算法），就足以引发一场小范围的传播风暴。

总结：AI 编程已形成“更好的模型-更好的体验-更多的用户-更多的数据”的迭代飞轮。

2. 市场空间：从存量替代到增量创造，vibe coding 实现代码平权

我们认为，AI 编程的市场机遇应从两个维度考量：其一，是对现有专业开发者市场的**存量替代与升级**，这是赛道价值的基础；其二，是通过“代码平权”赋能数亿知识工作者所开启的**增量创造**，这是赛道潜力的想象空间。一级市场对头部公司给出的高估值，已明确反映了对这一巨大变革的预期。

2.1. 存量市场：专业开发者的 AI 升级（~115 亿美元）

全球专业软件开发者是 AI 编程工具最直接、付费意愿最强的群体，我们测算其稳态市场规模超百亿美元，具备高度确定性。

核心用户基数： 全球软件开发者数量约 2870 万人，构成了一个庞大且高价值的用户池。目前，市场先驱 GitHub Copilot 的渗透率仅约 5%（130 万付费用户），表明市场尚处蓝海，天花板远未触及。

年均用户价值（ARPU）： 综合主流工具定价（月费 20-30 美元）及企业版更高的付费能力，我们设定综合 ARPU 为 400 美元作为中性假设。

市场规模测算（TAM/SAM）：

总潜在市场（TAM）： 理论上，仅专业开发者市场即可支撑起一个巨大的基本盘。

2870 万开发者×400 美金/年 ≈ 115 亿美金

可服务市场（SAM）： 我们判断，AI 编程工具有望在 3-5 年内，成为开发者不可或

缺的基础设施。以 40%的中性渗透率估算，中期可触达的市场规模已相当可观。

115 亿美元 TAM×40%≈46 亿美元

这是一个稳固的、高价值的存量市场。

2.2. 增量市场：“代码平权”赋能泛开发者（150 亿美元）

我们认为，AI 编程最大的想象力在于其打破软件创造的专业壁垒，实现“代码平权”，将软件开发能力从“专业生产者”扩散至“泛开发者”群体。

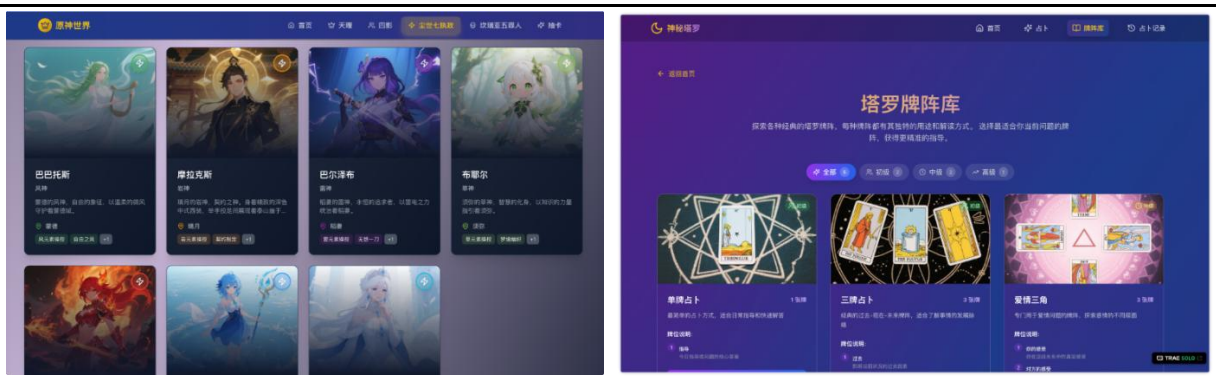
降低成本：过去开发一个软件项目，可能需要多名程序员数周乃至数月的工作。这是一个高门槛、高成本的活动。而 AI 编程的出现，使得软件的生产成本可以被指数级地压缩。这意味着，过去只有大型企业或特定项目才能负担的软件开发成本，现在普通个人或小型团队也能轻松承担。

增加需求：在网约车出现之前，出租车市场供给有限，打车贵且不方便，很多潜在的出行需求被压抑。而滴滴通过聚合大量私家车，极大地增加了出行服务的供给，显著降低了出行成本，使得那些原本不会打车、或打车成本过高的人群，也能享受到便捷的出行服务。这并没有取代传统的出租车，而是创造了一个全新的、远超传统规模的巨大市场。

同样地，在软件领域，长期以来存在着海量的个性化、小众化、或临时性的软件需求，由于传统开发成本过高而无法被满足。例如，一个小型咖啡店可能需要一个定制化的会员管理系统，一个个人创作者需要一个独特的作品展示网站，或者某个社区需要一个临时的活动报名工具。这些需求，在过去因为高昂的人力成本而被放弃。

当 AI 编程将软件开发的门槛和成本降到极低时，这些过去被压抑的需求将被彻底释放。每个人都可以成为软件的“创造者”和“拥有者”，无论是简单的工具、定制化的应用，还是用于特定目的的自动化脚本。这将催生出一个由海量个性化软件构成的巨大市场，其规模可能远超当前由标准化产品主导的软件市场。

图8：用户使用 Trae Solo 制作的网页



数据来源：founder park，东吴证券研究所

“泛开发者”群体的量化：我们将有软件创造需求、但缺乏专业技能的知识工作者（如产品经理、分析师、科学家、创业者等）定义为“泛开发者”。保守估计，其潜在规模至少是专业开发者的 5-10 倍，即 1.5 亿至 3 亿人（2030 年）。

增量空间测算：我们假设泛开发者的 ARPU 为 100-200 美元，低于专业开发者，主要考虑到该群体的付费场景可能更轻量化，付费意愿和能力相对较低，且市场需要通过更具性价比的定价来完成初期用户教育和习惯培养。但其庞大的基数将开启一个远超存量市场的增量空间。据我们测算，该增量市场潜在规模高达 **150 亿美元（2030 年）**。更重要的是，这仅是工具层收入，由数亿新晋创造者催生的应用与服务生态，其衍生经济价值将呈几何级数增长。

表2：2025E-2030E AI 编程市场规模测算（TAM/SAM）

	2025E	2027E	2030E
专业开发者	**核心逻辑：存量替代，渗透率与 ARPU 双升**		
用户基数（百万）	30	35	40
渗透率	15%	35%	50%
ARPU（美元）	400	450	500
市场规模（亿美元）	**18**	**55**	**100**
泛开发者	**核心逻辑：增量创造，用户基数与付费习惯养成**		
用户基数（百万）	150	200	300
渗透率	5%	15%	25%
ARPU（美元）	100	150	200
市场规模（亿美元）	**8**	**45**	**150**
合计市场规模（亿美元）	**26**	**100**	**250**

数据来源：东吴证券研究所测算

2.3. 蝴蝶效应：编程能力是 AI Agent 的底层基础设施

AI 编程是更底层的基础设施，在 AI 编程能力没有充分成熟之前，其他领域的 Agent 很难真正实现突破性发展。

设想一个为我们分析财报、预订旅行或管理社交媒体的 AI 智能体。如果它无法与数字世界进行交互，那么再强大的分析和规划能力也都是空谈。而这种交互的本质，就是一系列**编程任务**：

- 调用一个 API 来获取实时股价。
- 写一段 Python 脚本来清洗和处理数据。

- 与数据库进行交互以查询用户信息。
- 甚至动态生成一个简单的界面来展示结果。

没有一个强大、可靠的 AI 编程引擎，这些智能体能够思考，却无法行动。**AI 编程能力，就是赋予这些智能体在数字世界中行动的“双手”**。

AI 编程能力就像未来 AI 生态的“操作系统”的一部分。**大语言模型**提供了思考和推理的“**CPU**”。而 **AI 编程**则提供了**操作系统**——构建应用、服务和系统的底层技术支撑。

因此，AI 编程能力的成熟，其影响是指数级的。一个代码生成工具可靠性的提升，不仅仅是让程序员工作更快，它真正解锁的是成千上万种此前因技术上不可行而无法被创造出来的、能真正自主执行任务的 AI 智能体。这正是其“蝴蝶效应”的精髓所在。

就像 iOS 规范了手机编程的方向，许多创新应运而生。当每个人都能用编码模型开发自己的应用时，会出现大量定制化应用，世界将变得非常不同。

3. 技术演进与核心壁垒

3.1. 发展路径：Copilot→Pilot→Agent→Autopilot

AI 编程的发展路径可以划分为四个阶段。

图9：AI 编程的四个阶段

数据来源：founder park，东吴证券研究所

第一阶段：早期探索（“结对编程”模式）。早期的 AI 编程尝试可以追溯到类似 Tabnine 这样的辅助工具。这类工具试图通过机器学习方法辅助开发者，但受限于当时模型的能力，效果并不理想，未能形成主流，更多是局部的代码补全和修复。

第二阶段：Copilot 时代（“辅助驾驶”模式）。AI 编程领域的真正转折点发生在 2021 年，微软与 OpenAI 联合推出了 GitHub Copilot。**Copilot 模式就像程序员的搜狗输入法，**核心功能是代码补全——根据上下文，预测并推荐程序员接下来可能要写的代码片段，这极大地加速了编码过程，减少了重复性劳动和查找时间。除了代码补全，这一模式还

包括智能纠错、代码重构建议、单元测试生成等功能，所有这些都围绕着提升编码效率这一目标。它让程序员能够更专注于解决核心逻辑问题，而非繁琐的语法和细节。得益于 GPT-3 等大语言模型的进步，Copilot 模式成功找到了清晰的 PMF，获得了广泛的用户基础和付费意愿，成为目前 AI 应用中少数已实现规模化商业变现的赛道。

第三阶段：Agent 模式。在 copilot 模式下，开发者依然是主导者，AI 则扮演副驾驶的角色，提供实时的代码建议、补全和代码片段。而在 Agent 模式下，AI 追求更高的自主性。开发者真正需要什么？不是更多的代码补全，而是从繁琐、低效的沟通中解放出来。开发者花在让模型理解自己意图上的时间，可能比写代码还多。Cursor 想解决的就是这个根本问题。

第四阶段：Autopilot 模式。Autopilot 模式代表着 AI 编程的更高阶段，其目标是实现更高程度的自动化，将 AI 的角色从“辅助”提升到“自主”甚至“主导”。它不再仅仅面向专业程序员，而是希望将编程能力普惠到更广泛的用户群体，最终实现“代码平权”。Autopilot 模式旨在让非专业人士也能通过自然语言描述需求，由 AI 自主地生成、调试、测试乃至部署完整的软件应用。这意味着 AI 将从帮助人类编写代码，转向独立“创造”软件。

Devin 是这一模式下的代表产品，其目标是实现一个全能的“AI 软件工程师”，能够独立理解需求、规划任务、编写代码、调试测试并最终交付完整的软件项目，而人类更多扮演监督和验收的角色。这是当前研发的前沿阵地，也是所有顶尖玩家努力的方向。

Autopilot 模式的核心挑战在于如何实现产出结果的**高质量和可控性**。氛围编程（Vibe Coding），即 AI 生成的代码或应用，其质量和可用性往往像“开盲盒”一样，充满了不确定性。有时能生成令人惊喜的结果，有时则完全无法使用，甚至出现难以调试的错误。当前 AI 模型在理解复杂需求、处理边缘情况、保证代码鲁棒性等方面的不足。例如，目前 Devin 的任务失败率仍然较高。

未来需要通过更成熟的产品和技术方案来解决这一挑战。这意味着需要更强大的模型、更精细的 Agent 架构、更完善的工具链以及更有效的人机协作机制，以确保 AI 能够稳定、可靠地生成高质量的软件。只有解决了氛围编程的不可控性，Autopilot 模式才能真正落地并被大众接受。

图10：从 L1 到 L5：AI 编程的五个层级

Level	High-level Approaches	Example Popular Products
L1	Code-level Completion	GitHub Copilot, Tabby
L2	Task-level Code Generation Ticket to Code IDE with Chat	ChatGPT, Claude aider, cline, 16x Prompt Cursor, Continue, PearAI, Windsurf
L3	Project-level Generation Ticket to PR Prompt to UI	Claude Code, Codegen, Sweep Pythagora, Plandex v0
L4	PRD to Production AI Software Engineer	bolt.new, Trickle, Lovable Devin, Genie, Engine, devlo, Gru
L5	AI Development Teams	AutoDev, MetaGPT, MGX

数据来源：16x prompt，东吴证券研究所

3.2. 核心挑战：从“长文本”到“上下文管理”

当前最大的挑战已不再是模型能够处理多少字数的**上下文长度**，而是如何有效地进行**上下文管理**，尤其是在面对数万乃至数十万行的真实代码项目时。

在早期的 LLM 时代，模型的输入窗口（即一次性能够处理的文本长度）非常有限，通常只有几千到一两万 Tokens。

对于编程而言，这意味着模型在处理代码时，一次只能“看”到非常有限的代码片段。当程序员需要 AI 协助处理一个跨越多个文件、涉及复杂逻辑的功能时，模型往往无法获得完整的语境。例如，如果一个函数定义在一个文件，它所依赖的类定义在另一个文件，而使用它的地方又在第三个文件，早期模型很难同时理解这三者之间的关系，因为它“记不住”或“看不到”所有的相关代码。

这导致 AI 编程工具只能在非常局部的范围内提供辅助（如单行补全、简单函数生成），难以完成涉及多文件、多模块的复杂任务，大大限制了其在真实项目中的实用性。

近年来，随着 LLM 技术的飞速发展，模型的上下文窗口已得到显著扩展，从早期的几千 Token，发展到如今的数十万甚至上百万 Token。这意味着模型理论上可以一次性“读入”整个代码库，或者至少是项目中的大部分核心文件。

上下文长度的增加，确实缓解了 AI 在处理跨文件引用、理解大型函数逻辑等方面的部分挑战。现在，我们可以将更多的相关代码输入给模型，让它获得更全面的信息。

尽管上下文长度不再是主要障碍，但新的、更复杂的挑战随之浮现——“上下文管理”。这不仅仅是简单地将所有代码一股脑地塞给 AI，而是指：

1）全局认知与架构理解：在数万甚至数十万行的真实项目中，AI 需要形成对整个

代码库的**全局认知**。这包括理解项目的整体架构、不同模块之间的依赖关系、数据流向、设计模式、以及团队约定和隐式规则。人类程序员可以通过多年的经验和对项目的参与来建立这种全局认知，但 AI 需要通过更智能的方式来构建和维护。

2）信息筛选与优先级：在一个庞大的代码库中，并非所有代码都与当前任务相关。AI 需要具备智能的**信息筛选能力**，能够识别出与用户意图、当前修改目标最相关的代码片段、文档、配置信息，并进行优先级排序。如果将所有信息都喂给 AI，不仅效率低下，还可能因“噪音”过多导致模型困惑。

3）动态适应与记忆：代码库是动态变化的，新的代码被添加，旧的代码被修改或删除。AI 需要能够实时地**动态更新其对上下文的理解**。同时，它还需要具备一种“长期记忆”的能力，记住过去的操作、用户的偏好、以及历史修改的原因，避免重复犯错或提出不符合项目风格的建议。

4）多文件/多模块协调：当修改一个地方的代码时，往往会牵一发而动全身。AI 不仅需要在一个文件中进行修改，更要理解这种修改可能对其他文件或模块产生的影响，并能跨文件、跨模块地进行一致性地调整和修复。这需要高度的**协调能力**。

5）理解人类意图的“深层上下文”：除了代码本身，人类开发者在提出需求时，往往带有“为什么要做这个改变”、“希望达到什么效果”等深层意图。AI 需要通过对自然语言的深入理解，将这些模糊的意图转化为具体的编程操作，这需要它超越代码本身，理解人类的思考模式和项目目标。

为什么“上下文管理”更难？代码并非线性文本，而是高度网状的结构。文件之间通过引用、继承、接口等方式相互关联，形成复杂的依赖图。理解这些非线性关系，远比简单地读取一段文本要困难。许多项目上下文并非显式地写在代码或文档中，而是存在于团队的共识、历史决策、甚至不成文的规范中。AI 很难直接获取这些隐性知识。每个项目都是独特的，有其特定的技术栈、风格和历史包袱。AI 需要具备足够的通用性来处理各种项目，但同时也要有足够的能力去学习和适应每个项目的特异性。

结论：“上下文管理”是当前 AI 编程迈向更高阶自主化的核心挑战。解决这一瓶颈，将是未来 AI 编程工具能否真正实现“AI Agent 从创意到端到端解决问题”的关键。

3.3. 竞争壁垒：过程数据是终极护城河

对于开发者这类高度依赖工具和工作流的群体而言，习惯本身就是一种强大的护城河。当某个 AI 编程工具（如 Cursor）通过卓越的体验，成功在开发者群体中建立了强大的心智占有，成为其默认或首选工具时，就如同在用户心中建立了一个首选品牌。开发者会围绕它建立起自己的工作流程、快捷键记忆和交互习惯。即使有新的、可能在某些方面更好的工具出现，用户也需要花费时间、精力和心理成本去学习适应，打破原有的舒适区。这种高昂的转换成本，有效地阻挡了新进入者或竞争者的侵蚀。

公开的互联网数据（如 GitHub 开源代码）在训练大模型方面已接近饱和，其边际效益正在递减。未来 AI 竞争的胜负手，在于谁能更有效地获取、管理和利用私有的、高价值的“过程数据”。

一个优秀的 AI 编程工具，其核心竞争力不在于 UI 或集成多少模型，而在于它能否成为一个高效的“上下文操作系统”。即在用户与云端大模型之间，建立一个强大的**上下文预处理和编排层**。

上下文数据，指的是在用户与 AI 交互过程中产生的，反映其真实意图、偏好和工作流的动态数据。具体到编程领域，它包括：开发者如何提问、如何修改 AI 生成的代码、最终采纳了什么、在哪个环节卡住了、对哪些建议给出了负反馈等等。这些数据是训练出更懂编程、更懂开发者意图的 AI 的关键养料，其价值远超公开的静态代码。一个积累了大量优质过程数据的产品，其 AI 模型会越来越“懂”用户，从而提供更好的服务，这反过来又会增加了用户粘性，形成正向数据飞轮。

例如，2025 年，OpenAI 计划以约 30 亿美元收购 Windsurf，并非看重其产品或用户基数本身，而是其背后积累的海量、宝贵的“过程数据”，这些数据对于训练下一代强大的编程 Agent 至关重要。然而，这笔交易最终意外地失败了。原因在于，微软作为 OpenAI 的主要合作伙伴，其与 OpenAI 的协议授予了其对 OpenAI 技术的广泛访问权，这其中也可能包括被收购公司的知识产权。Windsurf 方面对将核心技术和数据暴露给微软心存疑虑，这一分歧最终导致交易破裂。这也表明过程数据是足以影响科技巨头间合纵连横的重要资产。

4. 竞争格局：VS Code Fork 系、小白平台、独立工具、巨头 MaaP

AI 产品按照产品形态可以分为四类。

图11：四类 AI 编程产品

VS Code 分支（Forks）

- **Cursor:** 早期AI编程IDE，从代码补全发展到Agent和MCP支持，功能强大但复杂性增高。
- **Windsurf:** 类似Cursor，用户体验更佳，支持MCP服务器和编辑器内应用预览。
- **Trae (字节跳动):** 免费额度与用户体验良好，但Agent功能与上下文管理对大型代码库支持有限。

VS Code 扩展

- **Amp:** 团队协作的自动编码Agent，按Token付费，专注于高质量输出，支持CLI自动化。
- **Augment:** 索引并分析代码库，提供问答和代码补全，免费版会训练用户代码。
- **Continue:** 兼具聊天和Agent模式，支持MCP集成，需手动指定上下文文件。
- **Cline:** Agent功能侧重任务自动化与“代码预测”，但Token消耗高昂。
- **Sourcegraph (Cody):** 企业级开发者工具，以“跨代码库感知”能力著称，支持搜索、重构和安全修复。

全栈应用

- **Tempo Labs:** 通过AI提示词可视化构建全栈应用，集成支付与数据库，生成PRD和用户流程图。
- **Bolt.new / Bolt.diy:** 从Figma导入设计，在浏览器内运行VS Code IDE，与Supabase集成。
- **Lovable.dev:** 对非编码者友好，能精准修改UI，集成Supabase和GitHub。
- **Replit & Base44:** 综合开发环境，支持AI辅助编程。

独立工具

- **Devin (Cognition Labs):** 自主AI软件工程师，能规划、实现、调试、测试代码，通过Slack交互。
- **Aider:** 面向高级用户的终端结对编程工具，对话驱动开发。
- **Claude Code (Anthropic):** 终端工具，能“读取和理解”代码库并保留会话记忆，Agent功能侧重任务自动化。
- **Fynix:** 代码演进追踪工具。
- **Pythagora:** 适合Node.js新应用，但UI和现有代码库支持有待提升。

数据来源：medium，东吴证券研究所

4.1. VS Code Fork 系：Cursor 和它的模仿者们

4.1.1. Cursor：选择难而正确的道路，重构 IDE，定义下一代开发体验

Cursor 实现了病毒式增长：其 ARR 从 2024 年 8 月的约 1900 万美元，增长至 2025 年 4 月的约 3 亿美元，并在 2025 年 6 月突破了 5 亿美元。其估值也随之飙升，在 2025 年 6 月的 C 轮融资中达到了 **99 亿美元**。

Cursor 的成功得益于两方面：

一方面，Cursor 快速集成市面上最强大的模型。另一方面，也是更重要的，Cursor 没有在**插件**这个红海里竞争，而是选择深度魔改（fork）VS Code 的开源代码，重构了一个 AI 原生的 IDE——一条“难而正确”的道路。

为什么要做独立的 IDE？

VS Code 提供了强大的插件生态系统，但其插件 API 在 UI 自定义、对编辑器核心行为的深层控制以及与编辑器其他部分的交互深度方面存在固有的限制。插件通常在一种相对沙盒化的环境中运行，其能力受限于 VS Code 团队预先定义和暴露的接口。

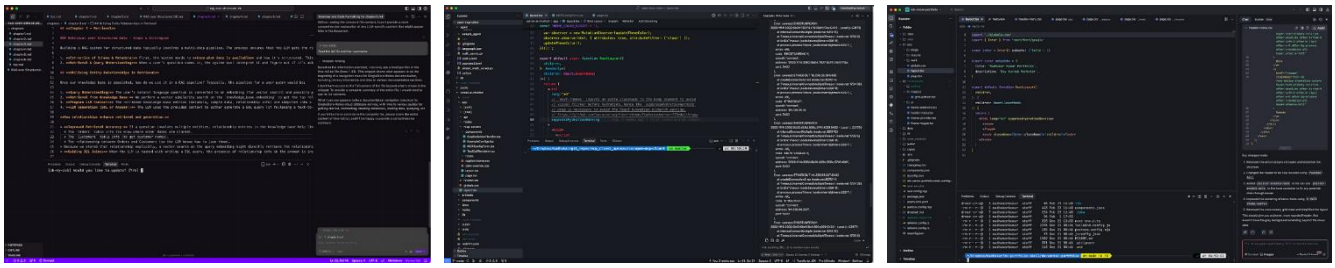
对于 Cursor 团队而言，他们的目标不仅仅是开发一个简单的代码补全或建议工具，而是要打造一个 AI native IDE——一个 AI 从根本上融入并重塑开发体验的集成开发环境。这种深度的集成要求对编辑器的 UI、交互逻辑、甚至是核心工作流进行大幅度的定制和创新。

例如，要实现像 Cursor 那样流畅且直观的多行代码 diff 展示（AI 修改建议以类似 Git diff 的形式直接在编辑器中呈现，用户可以方便地逐行接受或拒绝），或者像 Composer 那样能够理解用户意图并跨多个文件进行复杂、协调一致的代码生成与修改，标准的插件模型可能无法提供足够的灵活性和控制力。这些功能往往需要对编辑器的渲染机制、事件处理、文件系统交互以及核心编辑逻辑进行更底层的修改。

独立 IDE 的模式，使得 Cursor 成功地将自己从一个工具，变成了一个平台。每个开发者在 Cursor 上完成的每一次交互、修复的每一个 Bug、连接的每一个代码库，都在为这个平台添砖加瓦，让它变得更智能、更不可或缺。这种基于**上下文**的网络效应，是其最深的护城河。它不仅在蚕食存量市场，更在通过降低门槛创造增量市场。

Cursor 的成功是深度集成的胜利，但其挑战也源于此。Fork VS Code 为其带来了极致的体验，但也意味着它选择了一条**负重前行**的道路，需要持续投入巨大资源以跟进 VS Code 的主干更新。这更像是在**重建承重墙**，而非简单的**装修**。其长期壁垒，取决于它能否将当前的体验优势，转化为不可替代的**数据飞轮**。

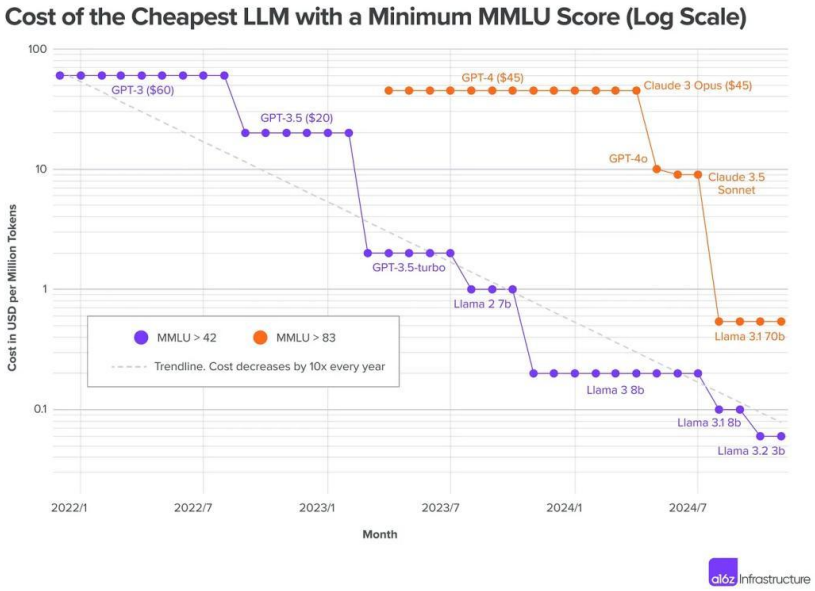
图12：VS Code Fork 类产品（左 Cursor/中 Windsurf/右 Trae）



数据来源：medium，东吴证券研究所

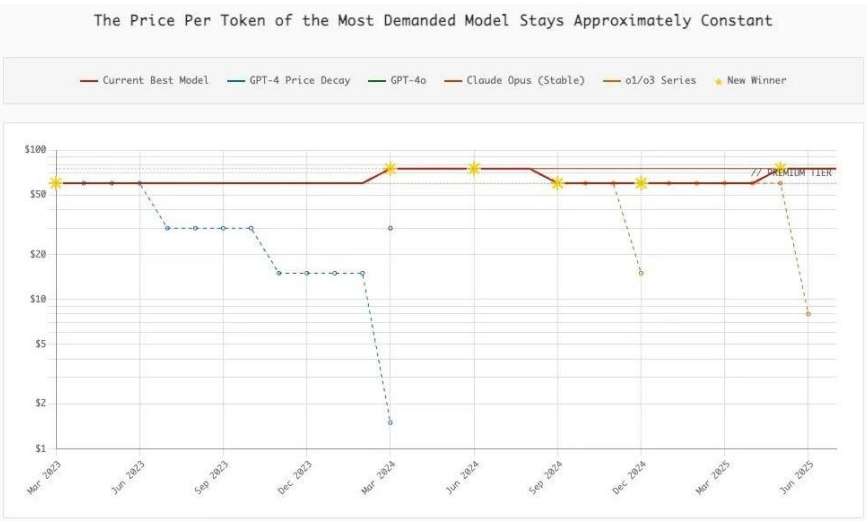
Cursor 的困境：固定的收入 vs 按量计费的支出。一方面，AI 行业普遍存在的“模型训练成本会下降，因此订阅服务未来能盈利”的信念是一个陷阱。现实是，用户只需求最先进的（SOTA）模型，而 SOTA 模型的推理价格居高不下。随着 AI 能力增强（如深度研究、Agent 化），单次任务消耗的 Token 数量正以指数级增长。这导致 AI 服务的单位经济模型（UE）非但没有改善，反而在持续恶化。

图13：LLM 成本每年下降 10 倍



数据来源：A16Z，机器之心，东吴证券研究所

图14：SOTA 大模型的价格维持刚性



Model	Time Period	Output Price ($/M tokens)
GPT-4 (winner)	Mar 2023	$60.00
GPT-4	July 2023	$30.00 [50% drop]
GPT-4	November 2023	$15.00
GPT-4	March 2024	$1.50
Claude 3 Opus (becomes winner)	March 2024	$75.00
GPT-4o	March 2024	$30.00
Claude 3.5 Opus (becomes winner)	June 2024	$75.00
o1 (becomes winner)	September 2024	$60.00
o1	Dec 2024	$15.00
o3 (becomes winner)	December 2024	$60.00
Claude 4 Opus (becomes winner)	May 2025	$75.00
o3	June 2025	$8.00

数据来源：机器之心，东吴证券研究所

4.1.1. Windsurf：被谷歌“人才收购”，证明了过程数据的重要性

2025 年 5 月，多家媒体报道 OpenAI 已与 Windsurf（前身为 Codeium）达成协议，计划以约 30 亿美元的价格进行收购。然而，这笔交易在最后阶段意外破裂。根据后续报道，交易失败的核心障碍在于微软。作为 OpenAI 的主要投资者和合作伙伴，微软与 OpenAI 的协议授予其对 OpenAI 技术的广泛访问权。Windsurf 方面担心这会使其核心知识产权和宝贵的“过程数据”暴露给微软，而 OpenAI 无法解决这一僵局，导致交易最终告吹。

在 OpenAI 收购失败后，谷歌在 2025 年 7 月 11 日迅速介入。但谷歌并未收购 Windsurf 公司本身，而是以一笔价值约 **24 亿美元**的交易，采取了“人才收购”的模式，将 Windsurf 的 CEO Varun Mohan、联合创始人 Douglas Chen 及部分核心研发团队招致麾下，并获得了其部分技术的非排他性许可。这表明**人才战争白热化，**顶尖的 AI 编程产品和工程团队已成为科技巨头争夺的战略性资产，其价值甚至可以与公司本身相提并论；也反映了 OpenAI 与微软之间复杂而紧张的合作关系。

2025 年 7 月 14 日，Cognition（Devin 的母公司）宣布，将收购 Agentic IDE 公司

Windsurf。收购包括 Windsurf 剩余的知识产权、产品、商标、品牌、业务，以及其剩余的团队。其计划是，短期内，Windsurf 团队将继续按原样运营，Cognition 将在未来几个月内大力投入，将 Windsurf 的能力和独特 IP 整合到 Cognition 的产品中。截至 2025 年 7 月，Windsurf ARR 达 8200 万美元，拥有 350 多个企业客户和数十万日活跃用户。

4.2. 小白平台：Replit 和它的追随者们

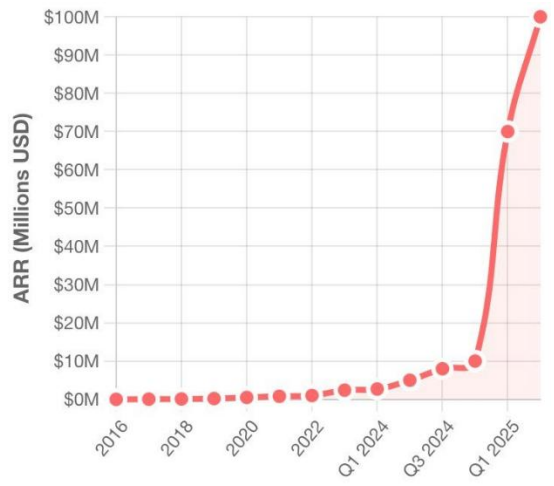
4.2.1. Replit：生成即上线，构建即运行

Replit 在 2024 到 2025 年间实现了爆炸性增长，其 ARR 在短短九个月内从不足 1000 万美元飙升至超过 1 亿美元。

定位于“全栈平台”而非仅为工具，为用户提供从 0 到 1 的体验。Replit 不仅仅提供 AI 代码生成功能，而是将 AI Agent 定位为流量入口，真正的核心是后端的基础设施，涵盖了托管、数据库、部署、监控等应用生命周期的所有环节。目标是成为一个“小白版的 AWS”，帮助小白用户从一个想法开始，完整地构建出一个产品（例如网站或软件）。Replit 的目标是让任何有创意的人都能通过软件解决问题，成为“通用问题解决器”。Replit CEO Amjad 认为，未来人们仍会用自然语言与 AI 交互，但底层的展现方式将不再是原始代码，而是某种更高级的、基于代码的抽象界面或视图。

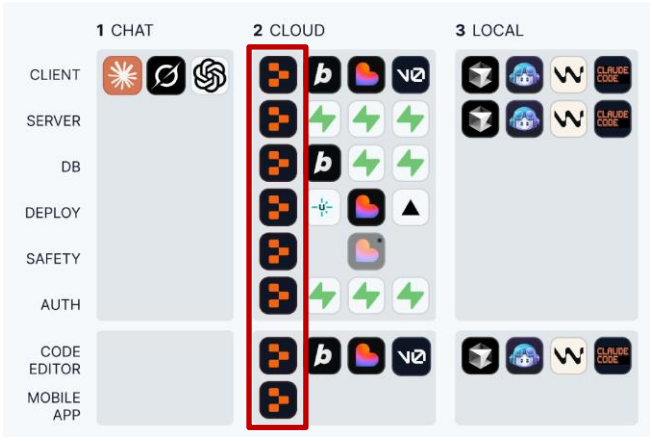
生成即上线，构建即运行：与许多竞品不同，Replit 强调从创意到上线的无缝体验。用户通过 Agent 生成的应用可以立即部署和运行，这得益于基础设施。CEO Amjad 强调，公司大量的工程投入都在底层基础设施上。与 Devin 那种长时间独立工作的“异步”模式不同，Replit Agent 更注重与用户的**合作与互动**。它会向用户展示规划，寻求确认，并提供及时的反馈，让用户感觉是在与一个助手并肩工作，而不是简单地把任务外包出去。发布后，根据用户反馈，Replit 迅速迭代，增加了更多透明度，并推出了一个类似 Cursor 的、更轻量级的 Assistant 功能，用于快速、小范围的代码修改。

图15：Replit ARR 快速增长



数据来源：InfoQ，东吴证券研究所

图16：Vibe Coding Landscape



数据来源：InfoQ，东吴证券研究所

商业模式：每个 token 都在亏钱，每层都赚钱。Replit 可以在代码生成环节亏钱获客，但在后续的应用托管和使用中实现变现。

1）**每个 token 都在亏钱：**对 Replit 而言，酷炫的 AI 代码生成功能是吸引用户注册和使用的“钩子”。它是一个非常强大的流量入口。为了让这个入口的门槛尽可能低，吸引海量用户，Replit 可能不会在 AI 功能上设定高昂的利润率，甚至可能是在补贴用户。也就是说，用户为生成代码支付的费用，可能还覆盖不了 Replit 付给 OpenAI 或 Anthropic 的 API 成本。从单纯的功能角度看，只卖 AI 推理服务（卖 token）的利润空间很薄，且竞争激烈。AI 代码生成本身不是 Replit 的主要盈利点，而是一个战略性的“获客成本”。

2）**每一层都在赚钱：**Replit 在 AI 功能之外提供的全栈平台服务，是 Replit 真正的“护城河”和利润来源。“层”指的是平台基础设施：当用户用 AI 生成了一个应用后，这个应用需要运行起来。这就需要一系列的后端服务。一旦用户在 Replit 上构建的应用开始有了真实用户，或者变得越来越重要，他们就很自然地会付费使用 Replit 提供的这些基础设施服务，而不会轻易迁移到别的平台。这部分业务的商业模式成熟（类似 AWS、Vercel 等云服务商），利润率也远比简单地转售 AI token 要高和稳定。

我们认为 Replit 找到了一个清晰的产品定位——**赋能普通人进行软件创造**，并通过优秀的产品设计和快速迭代，成功地将这一理念变为了现实。Replit 不是在卖一个“功能”，而是在卖一个从创意到上线再到运维的一整套解决方案，这使得它的商业模式更稳固，护城河也更深。

4.2.2. Lovable：存在被 Figma Make 替代的风险

Replit 推出之后，出现了类似的小白编程软件，例如 base44、lovable、bolt、v0。

如果说 Replit 解决了产品“如何实现和运行”的问题，那么 Lovable 的目标则是解决产品“看起来怎么样”的问题。Lovable 的核心功能是快速创建看起来非常逼真的产品界面模型，用于在产品开发的早期阶段进行展示、获取反馈和验证设计想法，而无需编写实际代码。

Lovable 可能会被 Figma Make 替代：

Lovable 的核心价值（制作原型）本身就是 Figma 的核心功能之一。Figma 作为一个行业领先的、功能强大的设计和协作平台，已经拥有非常成熟和广泛使用的原型制作能力。

Figma 是一个拥有庞大用户基础和完整生态系统的平台。对于设计师和产品团队来说，他们已经深度使用 Figma 进行界面设计、组件库管理等工作。在这个平台上直接完成原型制作，比切换到另一个单一功能的工具（如 Lovable）要高效得多。

Lovable 的护城河很浅。它的功能相对单一，很容易被像 Figma 这样的“巨无霸”平台通过增加或优化一个功能模块而完全覆盖和替代。一旦 Figma 决定在这方面投入更多资源，用户很可能会选择留在 Figma 的生态系统内，从而导致 Lovable 失去市场。

简而言之，Figma 可以替代 Lovable，是因为 Lovable 所提供的核心价值，Figma 不仅也能提供，而且能在一个更全面、更强大的平台上一站式地提供，这使得 Lovable 作为一个独立工具的竞争力显得非常脆弱。

表3：Figma Make 和 Lovable 核心功能对比

功能	Figma Make	Lovable
Figma 集成	可无缝集成到 Figma 中	无原生集成；通过 GitHub 导出
开发速度	用于原型制作速度快，用于编辑速度慢	更快的构建时间和更流畅的修订
代码输出	仅前端（HTML/CSS/JS）	全栈，包括后端
布局稳定性	初始布局稳固，但编辑时可能会出现问题	在更改过程中保持布局完整性
字体处理	出色的本地字体支持	标准网页字体实现
部署	选项有限	用于简化部署的 GitHub 集成
学习曲线	易于学习（可在 Figma 内使用）	中等难度；需要掌握 Git 知识并能给出清晰的指令

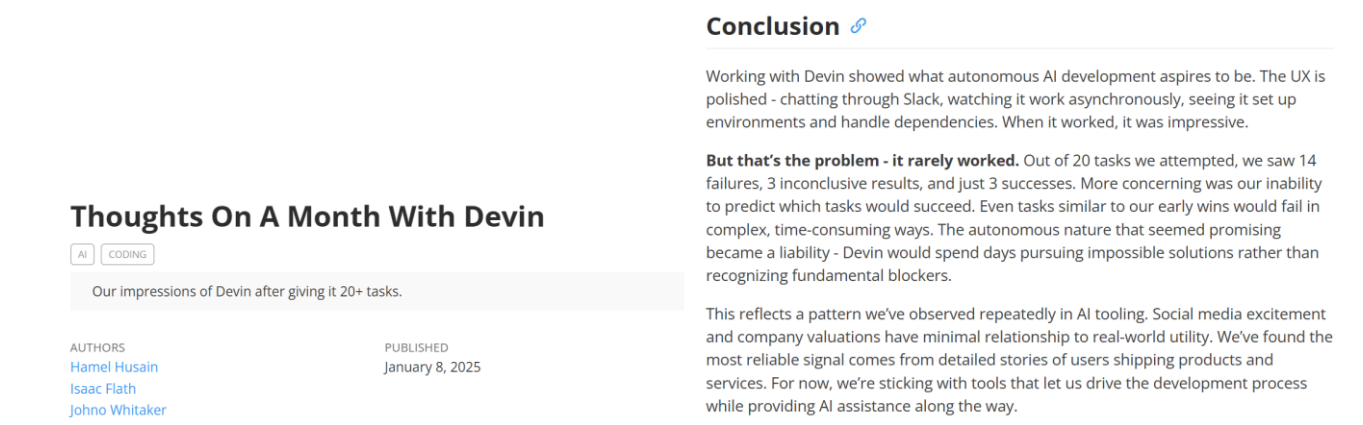
数据来源：Bonanza，东吴证券研究所

4.3. Agent 探索者：Devin 的美好明天与现实挑战

Devin 的愿景是成为 AI 软件工程师——一个宏大但实现起来困难重重的目标。在 Devin 发布初期，凭借“全球首位 AI 软件工程师”的定位和创始团队的“天才光环”（10 枚国际信息学奥赛金牌），并迅速获得了 20 亿美元的高估值。然而，这种预期很快就遭遇了现实的挑战。

根据 Answer.AI 在 2025 年 1 月的评测，Devin 在 20 个真实世界的任务中仅成功完成了 3 个（成功率 15%），并且在执行任务时常常陷入技术死胡同，或产生过于复杂、无法使用的解决方案。用户评测普遍认为，Devin 更像一个需要“手把手指导的初级开发者”，其工作流与开发者的实际习惯脱节，且每月 500 美元的高昂定价和无试用期策略引发了质疑。

图17：2025 年 1 月 AnswerAI 对 Devin 进行了 20 项任务测试（仅 3 项任务成功）



Thoughts On A Month With Devin

AI CODING

Our impressions of Devin after giving it 20+ tasks.

AUTHORS
Hamel Husain
Isaac Flath
Johno Whitaker

PUBLISHED
January 8, 2025

Conclusion

Working with Devin showed what autonomous AI development aspires to be. The UX is polished - chatting through Slack, watching it work asynchronously, seeing it set up environments and handle dependencies. When it worked, it was impressive.

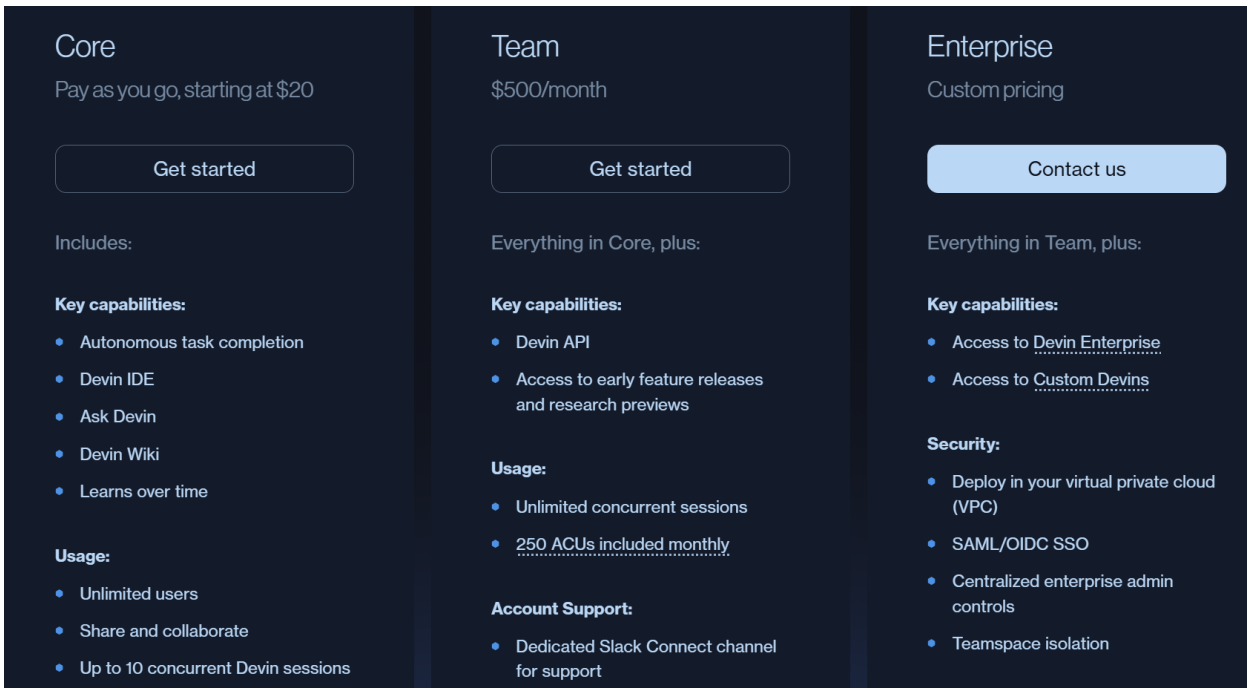
But that's the problem - it rarely worked. Out of 20 tasks we attempted, we saw 14 failures, 3 inconclusive results, and just 3 successes. More concerning was our inability to predict which tasks would succeed. Even tasks similar to our early wins would fail in complex, time-consuming ways. The autonomous nature that seemed promising became a liability - Devin would spend days pursuing impossible solutions rather than recognizing fundamental blockers.

This reflects a pattern we've observed repeatedly in AI tooling. Social media excitement and company valuations have minimal relationship to real-world utility. We've found the most reliable signal comes from detailed stories of users shipping products and services. For now, we're sticking with tools that let us drive the development process while providing AI assistance along the way.

数据来源：AnswerAI，东吴证券研究所

面对质疑，Cognition Labs 迅速做出调整。2025 年 4 月，公司发布了 **Devin 2.0**，新版本中，用户可以与 Devin 更紧密地协作、审查和编辑其工作。同时，增加了交互式规划、代码库搜索（Devin Search）和 Devin Wiki 等新功能。最关键的是，推出了每月 20 美元起的新计划，从一个遥不可及的高价企业工具，转变为一个更易于被广大开发者接受的产品。这表明 Cognition 正在努力弥合最初的预期差距，走向一条更务实的产品路线，试图在 toB 大单和 toC 订阅之间寻找平衡。

图18：Devin 新推出了 20 美元/月的付费方案



Core
Pay as you go, starting at $20
Get started
Includes:
Key capabilities:
- Autonomous task completion
- Devin IDE
- Ask Devin
- Devin Wiki
- Learns over time

Usage:
- Unlimited users
- Share and collaborate
- Up to 10 concurrent Devin sessions

Team
$500/month
Get started
Everything in Core, plus:
Key capabilities:
- Devin API
- Access to early feature releases and research previews

Usage:
- Unlimited concurrent sessions
- 250 ACUs included monthly

Account Support:
- Dedicated Slack Connect channel for support

Enterprise
Custom pricing
Contact us
Everything in Team, plus:
Key capabilities:
- Access to Devin Enterprise
- Access to Custom Devins

Security:
- Deploy in your virtual private cloud (VPC)
- SAML/OIDC SSO
- Centralized enterprise admin controls
- Teamspace isolation

数据来源：Devin 官网，东吴证券研究所

4.4. 字节跳动 Trae：立足明天，赢在当下

如果说 Devin 定义了 AI 编程“后天”的终极理想，那么字节跳动的 Trae 则选择了一条更务实的道路：聚焦并解决“明天”的核心痛点。

Devin 描绘的，是一个由完全自主的 AI 软件工程师驱动的理想场景，开发者仅需提出高级目标（如“构建一个抖音”），AI 便能独立完成从需求分析到部署上线的全部工作，人类则退居为创意提出者与最终审核者。Devin 的产品形态直指这一终局。

然而，直接跳到“后天”在当前阶段并不现实。最艰难的并非定义终局，而在于如何走好从“今天”到“后天”的过渡阶段——即“明天”。

Trae 的核心战略，正是专注于解决“明天”的问题。这一阶段，AI 能力飞速提升但尚不完美，开发者仍是流程核心，却面临着由 AI 带来的新痛点：1）**上下文的极度割裂**：开发者需在文档、IDE、终端等多个内嵌了不同 AI 的工具间手动同步信息，沟通成本高昂。2）**被忽视的非编码任务**：部署、运维、密钥管理等占开发者大量时间的“脏活累活”，现有 AI 工具赋能不足。Andrej Karpathy 的经验也证明，将项目从本地 Demo 部署上线依然痛苦。3）**低效的人机协作**：在 AI 尚不能完全自主时，开发者仍需深度介入（In the loop）。如何让这种必要的干预过程更顺畅，是核心挑战。

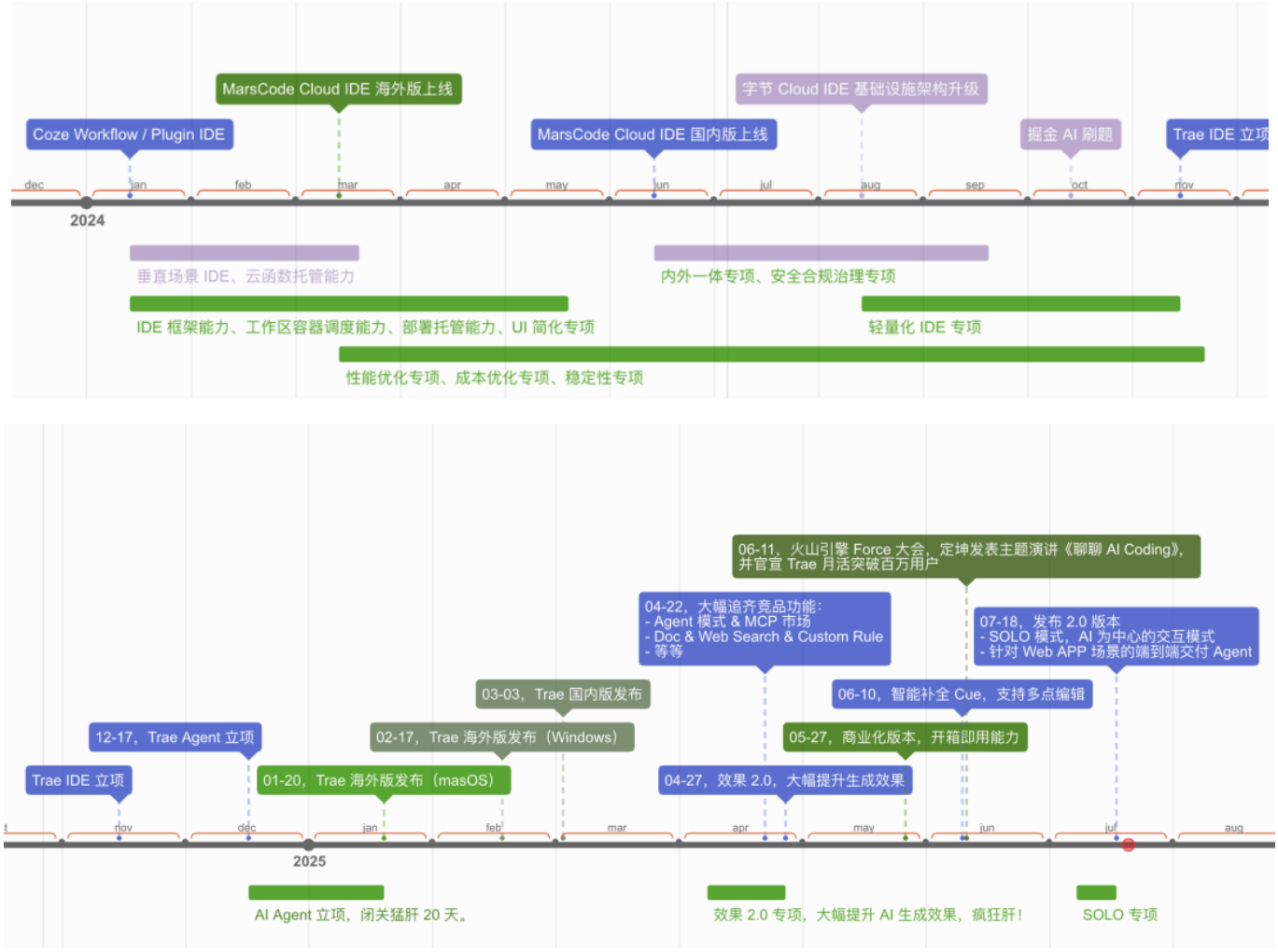
Trae 的产品矩阵：务实的“两步走”战略

Trae 的产品矩阵（Trae IDE + Trae Solo）体现了其“解决明天问题”的务实策略。

Trae IDE：极致优化“今天”，夯实用户基础。Trae 的前身是 AI 插件 Mars Code。受 Cursor 成功启发，字节意识到深度集成的重要性，转型开发了独立的 Trae IDE。Trae 首先通过深度集成 AI，将代码补全、对话等核心功能做到极致，提供顶级的类 Cursor 体验。这一步旨在服务好当下开发者，在最高频的场景建立用户信任。通过免费提供顶级 AI 模型（如 Claude 3.5 Sonnet, GPT-4o）和精美 UI，快速获取用户和开发过程数据。

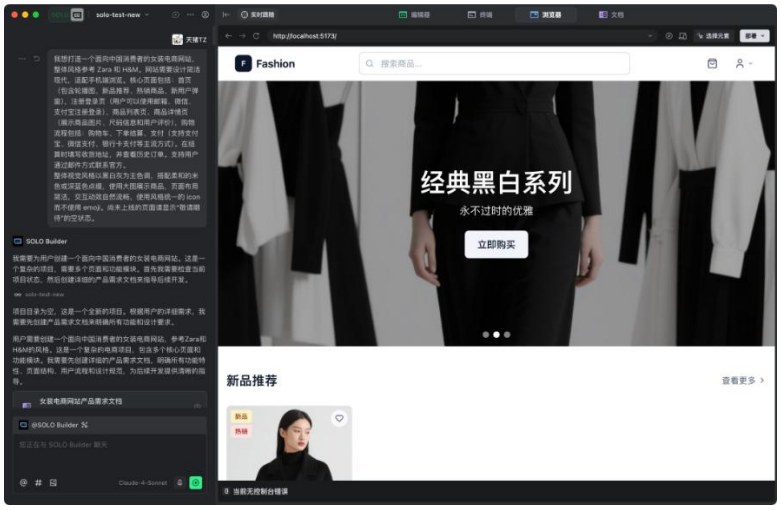
Trae Solo：迈向“后天”的第一步，是“明天”的解决方案。Trae Solo 的理念是，从“把 AI 放入工具”到“把工具放入 AI”，旨在从根本上解决工具割裂问题。它构建了一个以 AI 为核心的统一工作区，将浏览器、终端等工具作为能力集成进来，共享统一上下文，从而赋能更长的开发链路。不同于 Devin 的单个 Agent，Solo 让 Agent 群进行协同操作。由一个指挥官 Agent 拆解复杂任务，分发给多个专家 Agent（如测试、UI、数据库专家）并行处理，最终汇总成果。最重要的是，当 Solo 无法完成或用户需精细操作时，可随时无缝回退到熟悉的 Trae IDE 模式。

图19：字节 Marscode 和 Trae 发展时间轴



数据来源：founder park，东吴证券研究所

图20：Trae Solo 内置了浏览器



数据来源：founder park，东吴证券研究所

在商业化方面，Trae 现阶段不急于商业化，其战略是通过极致免费的产品体验快速扩大用户规模，最终目标是凭借对下一代开发工作流的定义，让自己成为“上下文工程”的行业标准。它的策略并非简单模仿 cursor，而是在 IDE 形态上追赶领先者，同时通过 Solo 探索并定义下一代产品的最佳形态。

4.5. 巨头入场：“模型即产品”的降维打击风险

现在有一种观点是“模型即产品”：一个底层 AI 大模型本身的能力已经强大到可以直接作为完整产品交付给最终用户，而不再仅仅是作为一种**服务或能力接口（API）**被开发者调用。为了更好地理解，我们可以将它与之前的“模型即服务”（Model as a Service, MaaS）进行对比：

图21：模型即服务 vs 模型即产品

模型即服务 (MaaS)

模型公司将AI模型的能力通过API接口封装起来，像提供水电煤一样，提供给其他开发者。开发者基于这些API，去构建自己面向特定场景的、独立的应用程序。模型公司在后端提供动力，应用公司在前台负责产品体验和交互，最终用户使用的是应用公司的产品。

例如：一家创业公司调用GPT-3 API，专门开发一个“AI小红书文案生成器”应用。

模型即产品 (MaaP)

在MaaP新模式下，模型本身变得极其通用和强大，以至于用户可以直接通过一个简单的界面与模型交互，来完成过去需要多个专门应用才能完成的任务。模型本身，就是用户体验的核心。一个好的Agent模型，似乎就能取代之前很多Agent应用。

例如：用户直接对GPT-4o说：“帮我写一篇关于xx产品的小红书推广文案”，模型直接完成任务。

数据来源：medium，东吴证券研究所

我们看到，越来越多科技巨头开始推出自己的 AI 编程产品，例如谷歌的 Jules 和 Gemini CLI，Anthropic 的 Claude Code，亚马逊的 Kiro。

表4：Gemini CLI、Claude Code 和 Amazon Kiro 对比

特性/工具	Gemini CLI	Claude Code	Amazon Kiro
发布时间	2025 年 6 月	2025 年 6 月	2025 年 7 月
核心定位	命令行界面 AI 编程助手	智能代码辅助，支持深度代码库理解与自动化多任务。	以规范驱动(spec-driven)的集成开发环境(IDE)为特色
主要优势	免费试用，调用次数多，快速响应；100 万 tokens 上下文	支持跨项目重构、依赖跟踪和隐私合规	自动生成需求、设计文档和任务列表，GUI 体验好
集成环境	主要命令行界面，轻量、快速	支持终端及主流 IDE 插件，兼顾灵活性和生产力	基于 VS Code 定制，具备更完善的界面
定价情况	免费：每分钟可发送 60 次模型请求，每日约 1000 次调用限制，超额按 token 计费	17 美元/月（sonnet 4 模型） 100 美元/月（opus 4.1 模型） 200 美元/月（更多额度）	免费版（50 次任务） 20 美元/月（225 次任务） 40 美元/月（450 次任务）

数据来源：Kiro，Anthropic，机器之心，东吴证券研究所

“模型即产品”会有什么影响？

对应用层创业公司的降维打击：如果创业公司产品的核心功能只是对大模型能力的简单“套壳”或“封装”，那么护城河就非常浅。一旦底层大模型（如 GPT 系列、Claude 系列）通过迭代，免费或以极低成本提供了同样的能力，这些“套壳”产品就会瞬间失去价值。例如，Claude Code 虽然推出时间晚于 Cursor，但其 ARR 达 4 亿美元，在快速追赶（Cursor 最新 ARR 是 5 亿美元）。

价值链的重塑：模型公司不再满足于只做“卖水人”或“电力公司”，它们希望直接拥有最终用户，掌握用户数据和交互入口，从而捕获整个价值链中最大的利润。这是从“To B/To D (Developer)”向“To C”的战略延伸。

对应用创新的要求更高：这并不意味着应用层没有机会，而是**对创新的要求被提到了新的高度**。未来的 AI 应用想要成功，必须提供超越底层通用大模型的能力，例如极致的垂直领域优化、无与伦比的用户体验、强大的生态和网络效应。

总结来说，“模型即产品”标志着 AI 竞赛的核心正从“谁的应用做得好”转向“谁的底层模型能力更强、更通用”。对于用户而言，这意味着更强大、更简洁的体验；而对于创业者而言，这意味着必须构建更深、更独特的价值，才能在巨头的阴影下生存和发展。

4.6. 中国力量：国产模型的追赶与开源战略

国产模型（Kimi, Qwen，智谱等）快速迭代，Hugging Face 上中国模型基本垄断了热门榜。

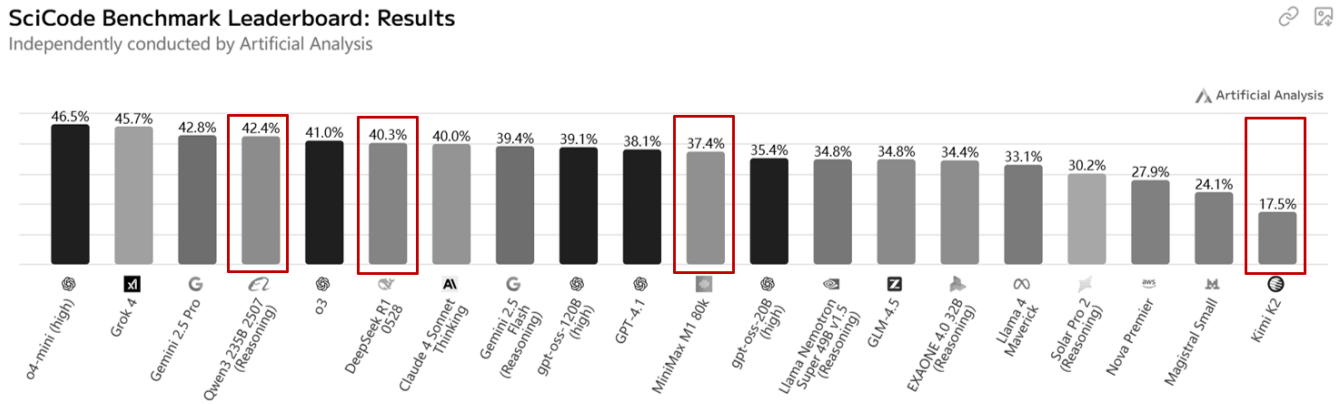
图22：huggingface 上的 big code model leaderboard

Model	Win Rate	humaneval-python	java	javascript	cpp
Qwen2.5-Coder-32B-Instruct	59.17	83.2	73.69	76.05	81.95
Qwen2.5-Coder-32B	56.67	57.1	65.49	65.07	64.35
OpenCodeInterpreter-DS-33B	56.25	75.23	54.8	69.06	64.47
Nxcode-CQ-7B-orpo	55.92	87.23	60.91	71.69	68.04
CodeQwen1.5-7B-Chat	55.67	87.2	61.04	70.31	67.85
CodeFuse-DeepSeek-33b	54.67	76.83	60.76	66.46	65.22
DeepSeek-Coder-33b-instruct	52.25	80.02	52.03	65.13	62.36
Artigenz-Coder-DS-6.7B	51.67	70.89	56.84	66.16	59.75
DeepSeek-Coder-7b-instruct	50.58	80.22	53.34	65.8	59.66
OpenCodeInterpreter-DS-6.7B	49.75	73.2	51.41	63.85	60.01
Phind-CodeLlama-34B-v2	49.42	71.95	54.06	65.34	59.59

数据来源：Huggingface，东吴证券研究所

在代码领域，中国大模型已具备全球竞争力。从 SciCode 代码能力基准测试排行榜可以看出，来自阿里巴巴的 Qwen 和深度求索的 DeepSeek 模型已经跻身全球第一梯队，其性能评分与谷歌的 Gemini 2.5 Pro、Claude 4 Sonnet 等模型非常接近。

图23：中国公司的大模型的 coding 能力快速追赶



数据来源：Artificial Analysis，东吴证券研究所

注：SciCode Benchmark 是一个由科研人员构建的评分基准，测试集包括 16 个科学学科、80 个真实实验的 338 项任务。不同于传统的编程任务评分基准，SciCode 要求大模型将科学知识与编程技能融会贯通，旨在解决现实中的问题，而非解答抽象的问题。

图24：OpenRouter 编程领域的 Tokens 消耗量：按照模型拆分

2025/3/11~3/17		2025/5/20~5/26		2025/7/29~8/4	
Gemini 1.5 Flash	12%	Claude Sonnet 4	30%	Claude Sonnet 4	29%
openAI o3 mini	20%	Claude Sonnet 3.7	18%	Horizon Beta	20%
qwen 2.5 72B	9%	Gemini 2.5 pro	13%	Qwen 3 Coder	14%
GPT-4o-mini	7%	Gemini 2.5 flash	16%	GLM 4.5	6%
Gemini 2.0 pro	6%	GPT-4o-mini	7%	Kimi K2	6%
qwen QwQ 32B	5%	GPT-4.1	3%	Gemini 2.5 pro	4%
GPT-4.5	5%	Claude Opus 4	2%	Gemini 2.5 flash	4%
Grok beta	5%	Gemini 2.0 flash	1%	Claude Sonnet 3.7	4%
其他	30%	其他	10%	其他	14%

数据来源：OpenRouter，东吴证券研究所

4.6.1. Qwen 3 Coder：Agent 化的工程能力

阿里巴巴的 Qwen 3 Coder 能力惊艳。2025 年 7 月，阿里巴巴在发布 Qwen 3 Coder 代码大模型，是阿里迄今为止最强大的代码模型。它的定位超越了简单的代码助手，是一个具备强大自主能力的 AI Agent。在性能上已非常接近全球顶尖的闭源模型 Claude 4，并超越了 GPT-4 的编码能力，成为国产代码模型领域的领军者。

图25：Qwen3 Coder 在 SWE-Bench 排名第四（截至 2025/8/7）

Model	% Resolved	Org	Date	Logs	Trajs	Site	Release
Claude 4 Opus (20250514)	67.60		2025-08-02	✓	✓		1.0.0
Claude 4 Sonnet (20250514)	64.93		2025-05-21	✓	✓		1.0.0
o3 (2025-04-16)	58.40		2025-05-21	✓	✓		1.0.0
Qwen3-Coder 480B/A35B Instruct	55.40		2025-08-02	✓	✓		1.0.0
Gemini 2.5 Pro (2025-05-06)	53.60		2025-05-21	✓	✓		1.0.0
Claude 3.7 Sonnet (20250219)	52.80		2025-05-21	✓	✓		0.0.0
o4-mini (2025-04-16)	45.00		2025-05-21	✓	✓		1.0.0
GPT-4.1 (2025-04-14)	39.58		2025-05-21	✓	✓		1.0.0

数据来源：SWE Bench，东吴证券研究所

关键技术优势：Qwen 3 Coder 的能力主要来源于独特的训练方式和高质量数据。最核心的优势在于利用了阿里云强大的基础设施，部署了大规模的仿真编码环境来进行强化学习训练，这使得模型能在接近真实世界的开发场景中不断试错和提升，是其他公司难以复制的。此外，它的训练也基于海量的、以代码为核心的高质量数据集以及 Qwen 3 的基础模型。

从“写代码”到“解决问题”。Qwen 3 Coder 的设计核心是“Agent 化”，其关键能力是“长过程推理”。它能像一个初级软件工程师一样自主工作，在接收任务后，可以在仿真环境中自主搜索、调用工具、安装依赖、读取代码库文件并进行长达数小时的持续纠错和推理。它还展现出优秀的规划能力，能在编码前详细规划应用架构，并熟练调用浏览器、终端等外部工具来完成复杂任务。

开源战略与生态影响：阿里为 Qwen 3 Coder 选择了与海外巨头截然不同的开源战略。它采用了非常宽松的 Apache 2.0 开源协议，允许免费商用，旨在为因海外工具封禁而困扰的中国开发者提供一个强大的国产替代方案。通过开源，阿里希望快速构建一个围绕 Qwen 3 Coder 的开发者社区，挑战海外闭源模型的主导地位，并通过魔搭社区等平台提供免费 API 调用，极大地降低了使用门槛。

4.6.2. Kimi：以长文本优势，攻克上下文管理难题

2025 年 7 月，月之暗面发布 Kimi K2 模型，在极致的长文本处理能力（这是 Kimi 的立身之本）之上，在通用模型能力、代码能力、agent 能力都有明显提升，在主流基准测试上表现出色，且价格只有 Claude Sonnet 4 的约 20%。

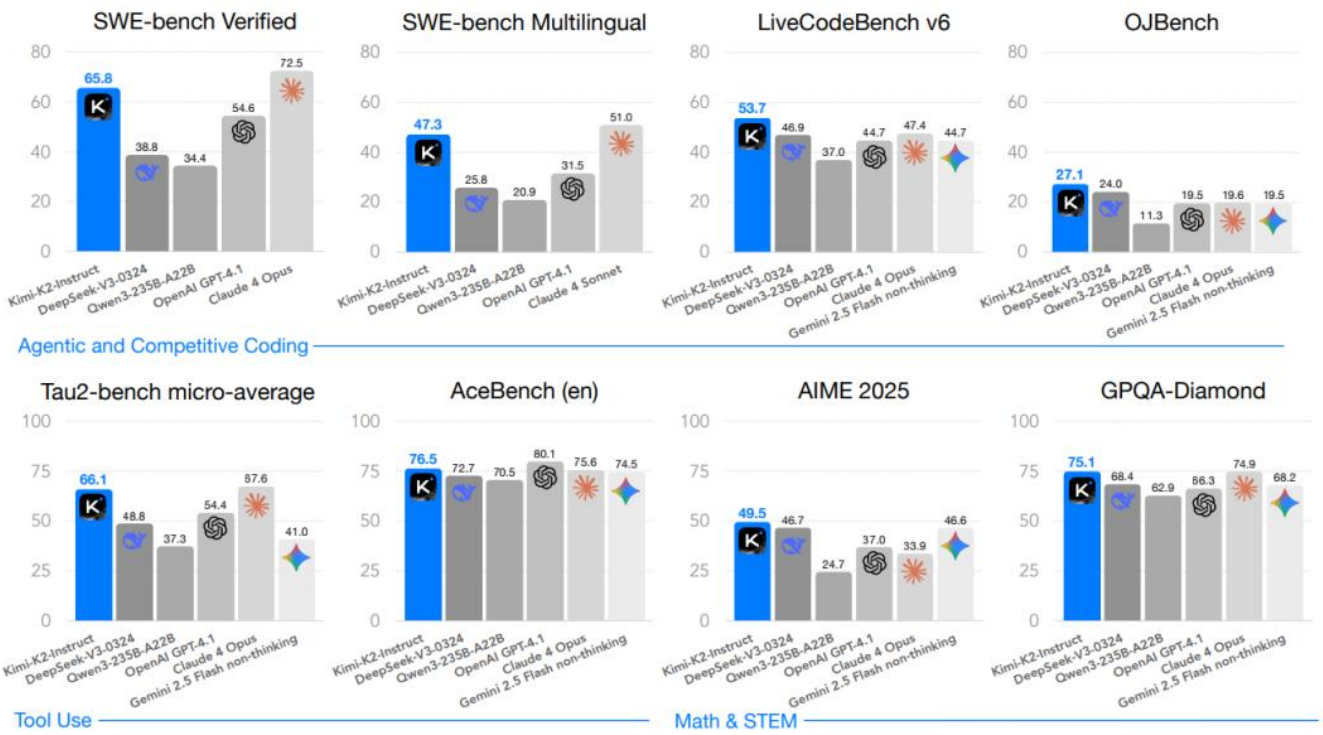
在众多 AI 编程的竞争者中，Kimi 是一个特殊的存在。虽然它并未像 Qwen Coder 或 Devin 那样推出专门的编程产品，但其底层模型的核心能力，尤其是行业领先的长文

本处理技术，恰好直击当前 AI 编程从“辅助工具”迈向“自主智能体”的最大瓶颈——**上下文管理**。

Kimi 的核心壁垒在于超长上下文能力。AI 编程最大的挑战已不再是生成单段代码，而是在数万甚至数十万行的真实项目中，形成对代码库的全局认知。Kimi 的核心优势正在于此。其 K2 模型支持的 200 万字超长上下文窗口，并非简单的“读得长”，而是为 AI 提供了构建完整“记忆宫殿”的能力。它能理论上一次性“读入”整个项目，理解不同模块间的复杂依赖、数据流向和设计模式，这是实现跨文件重构、修复深层 Bug 和遵循项目架构规范的基础。当开发者提出模糊需求时，Kimi 可以基于对代码库、文档、甚至历史提交记录的全面理解，更精准地推断其真实意图，而不仅是进行字面上的代码翻译。

综合能力跃升：从“特长生”到“全能型选手”。如果说早期 Kimi 依靠长文本建立护城河，2025 年 7 月发布的 K2 模型则标志着它已从“特长生”进化为“全能型选手”。其在 MMLU、MATH、HumanEval 等主流基准测试上的表现，证明了其通用推理和代码生成能力已达到或接近 GPT-4o 和 Claude 3.5 Sonnet 等世界模型的水平。这意味着 Kimi 不仅能“看得全”，更能“想得深”、“做得对”，其编程能力不再是短板，而是与长文本优势相辅相成的强项。

图26：Kimi K2 在主流基准测试（MMLU、MATH、HumanEval 等）上表现出色

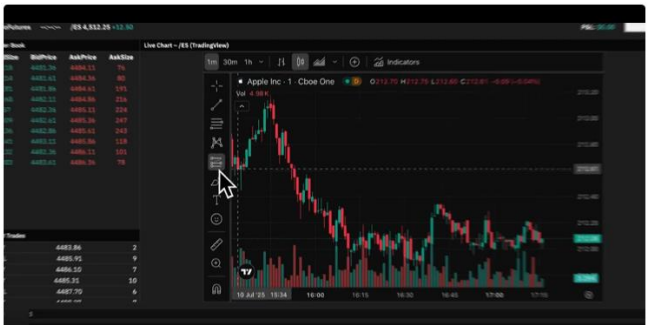


数据来源：量子位，东吴证券研究所

图27：Kimi K2 Instruct 生成的网页版虚拟钢琴

数据来源：机器之心，东吴证券研究所

图28：Kimi K2 Instruct 生成的期货交易系统



数据来源：月之暗面，东吴证券研究所

战略定位：隐藏的“黑马”与极致的性价比。Kimi 的独特性在于，它选择将顶尖的编程能力作为一种基础设施通过 API 开放，而非封装成独立的编程产品。开发者和企业可以直接调用 Kimi 强大的底层能力，构建自己的、深度定制化的编程助手或 Agent，而不是被限定在某个特定产品的框架内。Kimi K2 模型的 API 定价极具竞争力，据公开数据，其价格仅为同级别模型 Claude 4 Sonnet 的约 20%。这种激进的定价策略极大地降低了开发者使用顶级 AI 能力的门槛，对高价的闭源模型构成了强大的市场压力。

表5：Kimi K2 Instruct 的价格比 Claude 4 Sonnet 便宜~80%

模型	输入价格（美元/每百万 tokens）	输出价格（美元/每百万 tokens）
Kimi K2 Instruct	0.58	2.29
Claude 4 Sonnet	3	15
GPT-4.1	2	8

数据来源：硅基流动，东吴证券研究所

总结而言，Kimi 凭借“**极致长文本 + 顶级通用能力 + 颠覆性价格**”的组合拳，虽然在产品形态上保持低调，却在 AI 编程最核心的能力层建立起了难以复制的优势，是未来竞争格局中一股不可忽视的强大力量。

4.6.3. 开源策略：成为 AI 时代的 Android

开源，一场成为 AI 时代 Android 的阳谋。过去，市场对国产大模型的期待，逻辑多为“追赶并替代 OpenAI 在国内的市场”——这是一套防御性的叙事。然而，随着以 Qwen、Kimi、智谱为代表的国产模型性能追至世界前沿，一种更宏大、更具进攻性的全球战略浮出水面：**通过开源，成为 AI 时代的 Android**。这并非简单的免费策略，而是一场旨在构建全球性技术生态的“阳谋”。正如安卓系统为全球手机厂商提供了开放、免费的底层操作系统，催生了繁荣的应用生态一样，中国开源大模型正试图扮演类似角色。

全球市场，特别是欧美之外的广大发展中国家和地区，正处于数字化转型浪潮中，但普遍缺乏自研先进基础模型的能力和资源。这形成了一个巨大的市场空白。中国开源模型以其“世界级性能 + 极低成本”的组合，提供了一个开箱即用的解决方案。

一旦海外的开发者、创业公司和企业基于中国的开源模型进行应用开发，强大的网络效应与路径依赖便开始形成。代码、人才、解决方案、行业 Know-how 都将围绕这个技术基座沉淀。未来，从上游的 AI 芯片、云服务，到下游的行业解决方案，都可以围绕“中国标准”的生态展开。这不仅是赢得用户，更是锁定一个时代的开发者，其护城河比单个闭源产品更深。

5. 投资建议：短期看入口，中期看数据，长期看生态

第一阶段：投资 killing app。在行业发展的初期，市场格局混沌，最确定的投资机会来自于那些能够解决特定、高频痛点，并提供极致产品体验的 killing app。它们能够通过“产品力”这一最直接的方式，快速获得用户增长和商业化验证。寻找展现出强大产品驱动增长（PLG）势头的公司，其用户增长主要依靠社区口碑传播。重点关注前端开发、特定语言/框架的深度优化、以及自动化测试与调试等细分赛道。这类公司是赛道中最具爆发力的群体，增长曲线陡峭，并且是科技巨头最青睐的“收购标的”。

第二阶段：投资超级工作台。随着市场进入整合期，单纯的工具将面临增长天花板。竞争的核心将从“单个工具的好用性”转向“整个工作流的流畅性”。能够将各种工具链整合起来，为开发者提供一站式超级工作台的公司，将开始构建起真正的平台级护城河。它们正在构建的是开发者高度依赖、难以迁移的基础设施，一旦平台地位确立，将享有极高的用户粘性、强大的定价权以及广阔的延伸空间。

第三阶段：投资垂直行业冠军&生态玩家。当通用的 AI 编程平台能力趋于“商品化”时，最高的、最可持续的利润将不再产生于“如何写代码”，而是产生于“用代码解决了什么高价值的行业问题”。1）寻找那些将 AI 编程能力与深厚的 Know-how 相结合，在垂直领域构建起绝对优势的公司。2）围绕繁荣的开源 AI 编程生态，提供企业级支持、安全保障和私有化部署的服务商。3）产业链上游的“终极卖铲人”：例如 AI 芯片公司和头部大模型厂商。

6. 风险提示

1. 技术路径与实现风险：自主的 AI 智能体在真实世界的复杂任务中仍面临巨大挑战。AI 能否从“辅助驾驶”真正跃迁到可靠的“自动驾驶”，存在技术不确定性。

2. 商业模式与盈利风险：随着 AI 能力向 Agent 化演进，单次任务消耗的 Token 数量呈指数级增长，导致服务的单位成本非但没有改善，反而可能在持续恶化。

3. 市场竞争与生态风险：随着“模型即产品”趋势的兴起，底层大模型厂商正将越来越多的能力直接集成到其核心产品中，这对应用层创业公司构成“降维打击”的风险。

东吴证券 SOOCHOW SECURITIES

免责声明

东吴证券股份有限公司经中国证券监督管理委员会批准，已具备证券投资咨询业务资格。

本研究报告仅供东吴证券股份有限公司（以下简称“本公司”）的客户使用。本公司不会因接收人收到本报告而视其为客户。在任何情况下，本报告中的信息或所表述的意见并不构成对任何人的投资建议，本公司及作者不对任何人因使用本报告中的内容所导致的任何后果负任何责任。任何形式的分享证券投资收益或者分担证券投资损失的书面或口头承诺均为无效。

在法律许可的情况下，东吴证券及其所属关联机构可能会持有报告中提到的公司所发行的证券并进行交易，还可能为这些公司提供投资银行服务或其他服务。

市场有风险，投资需谨慎。本报告是基于本公司分析师认为可靠且已公开的信息，本公司力求但不保证这些信息的准确性和完整性，也不保证文中观点或陈述不会发生任何变更，在不同时期，本公司可发出与本报告所载资料、意见及推测不一致的报告。

东吴证券投资评级标准

投资评级基于分析师对报告发布日后 6 至 12 个月内行业或公司回报潜力相对基准表现的预期（A 股市场基准为沪深 300 指数，香港市场基准为恒生指数，美国市场基准为标普 500 指数，新三板基准指数为三板成指（针对协议转让标的）或三板做市指数（针对做市转让标的），北交所基准指数为北证 50 指数），具体如下：

公司投资评级：

买入：预期未来 6 个月个股涨跌幅相对基准在 15%以上；

增持：预期未来 6 个月个股涨跌幅相对基准介于 5%与 15%之间；

中性：预期未来 6 个月个股涨跌幅相对基准介于-5%与 5%之间；

减持：预期未来 6 个月个股涨跌幅相对基准介于-15%与-5%之间；

卖出：预期未来 6 个月个股涨跌幅相对基准在-15%以下。

行业投资评级：

增持： 预期未来 6 个月内，行业指数相对强于基准 5%以上；

中性： 预期未来 6 个月内，行业指数相对基准-5%与 5%；

减持： 预期未来 6 个月内，行业指数相对弱于基准 5%以上。

我们在此提醒您，不同证券研究机构采用不同的评级术语及评级标准。我们采用的是相对评级体系，表示投资的相对比重建议。投资者买入或者卖出证券的决定应当充分考虑自身特定状况，如具体投资目的、财务状况以及特定需求等，并完整理解和使用本报告内容，不应视本报告为做出投资决策的唯一因素。

东吴证券研究所
苏州工业园区星阳街 5 号
邮政编码：215021
传真：（0512）62938527
公司网址： http://www.dwzq.com.cn