Table of Contents

导出说明：本文件由 ChatGPT Deep Research 原始报告正文直接导出。仅移除了无法在站外解析的内部引用标记；正文、数据、表格、链接与结论未作编辑性改写。

AI 对软件行业的冲击与未来走向：以 WordPress 插件与主题生态为案例的科学与行业数据研究

执行摘要

截至 **2026** 年 **9** 月可核验的公开资料，证据更支持这样一个判断：**AI 正在快速替代“软件开发中的任务工时”，但尚没有证据支持“按同样比例替代软件开发岗位”**。相反，软件供给正在扩张、开发门槛正在下降、开发者数量和软件支出仍在增长，而价值正在从“把代码写出来”迁移到“定义正确问题、掌握领域知识、设计系统、验证正确性、承担安全与运营责任、获得客户与分发渠道”。2026 年 Gartner 预计全球软件支出约 **1.468 万亿美元，同比增长 15.5%**；全球 AI 支出约 **2.67 万亿美元，同比增长 49.5%**，其中 AI 软件约 **4616 亿美元**。这说明 AI 更像是在重构软件价值链，而不是让软件行业本身消失。

开发者采用率已经非常高。Stack Overflow 2025 年调查覆盖 49,000 多名受访者、177 个国家，**84% 使用或计划使用 AI 开发工具，51% 的职业开发者每天使用**；但与此同时，**46%** 不信任 **AI 输出准确性，只有 33% 表示信任，66% 最常抱怨“答案几乎正确但差一点”，45% 抱怨调试 AI 生成代码反而更耗时**。这组数据揭示了最重要的行业分化：**代码生成能力的增长速度，已经明显快于验证、责任和可靠性交付能力的增长速度。**

科学实验进一步要求我们避免“AI benchmark 高分 = 开发者可被等比例替代”这一错误推导。METR 在 2025 年对经验丰富的开源开发者进行随机对照研究，在他们熟悉、成熟的真实代码库上测试 AI 辅助开发；开发者事前预测 AI 会让自己快约 24%，事后仍认为快约 20%，但计时结果显示实际**慢了约 19%**。与此相对，Anthropic 从大量 Claude 对话推算，在被选择来交给 AI 的任务上可能出现非常大的时间节省，其模型估计平均接近 80%，但 Anthropic 自己明确指出这种方法会漏掉对话之外的检查、返工、整合等成本。因此，两者不是互相矛盾，而是在测量不同层次：**局部、可规格化任务可以被大幅加速；复杂真实工程的端到端收益仍高度依赖上下文、代码库熟悉度、测试、审查和集成成本。**

WordPress 是观察这一变化的非常好案例。WordPress 官方称其支撑全球超过 **43% 的网站**；当前插件目录页面显示可浏览 **73,000+ 免费插件**，主题目录首页称 **15,000+ 免费主题**。同时，WordPress Plugins Team 披露，2025 年插件提交量显著上升：全年审核 **12,713 个插件，同比增长 40.6%**，批准 **5,415 个**，比 2024 年的 3,259 个增加 66.2%；2025 年中期新插件提交同比一度增长约 **87%**，官方团队明确提到可以观察到 AI 对提交增长的影响。更有代表性的是，截至 2025 年 8 月，**85.3% 的首次插件审核已经由使用算法与 AI 的自动系统启动**。也就是说，AI 同时降低了“生产插件”的成本和“审核插件”的成本。

这将使 WordPress 以及更广的软件市场发生一个核心变化：**简单功能本身将越来越不稀缺。** 一个 CRUD、表单增强、短代码、简单区块、CSS/JS 定制、基础 API 包装器或普通 AI

内容生成插件，未来很难仅凭代码本身长期维持溢价。相反，越来越稀缺的是：可靠的跨插件/跨版本兼容性、支付与业务关键流程、安全与隐私、性能、迁移能力、特定行业知识、长期支持、客户关系、品牌与分发、专有数据，以及能够证明 AI 输出正确的评估与验证系统。这是本文对未来开发者策略的核心结论。该判断是基于软件支出仍增长、GitHub 开发者与项目活动仍扩大、WordPress 插件供给快速增长，以及复杂任务端到端自动化依然明显弱于局部代码生成能力所作出的综合推断。

方法论、数据来源与证据边界

本文按用户指定的 **2019—2026** 年**观察窗**口研究；严格来说它已超过五个自然年，因此本文以“2019 至今”作为实际口径。优先使用官方平台遥测、监管机构资料、原始学术研究，其次使用大型开发者调查和市场研究机构数据，再其次才是厂商自身研究。未来替代率与时间线属于本文的**情景估计**，不是已经发生的统计事实。

证据层级	来源类型	本文主要来源	用途	主要局限
最高	平台原始遥测、官方统计	WordPress.org / Make WordPress、GitHub Octoverse、美国 BLS	插件数量、审核流量、开发者平台规模、职业数据	平台用户不等于全球职业开发者
高	随机对照或原始学术研究	METR 真实开源项目实验	衡量真实工程中的生产率	样本规模较小，聚焦成熟代码库与资深开发者
高	大规模行业调查	Stack Overflow Developer Survey	AI 使用、信任、开发者主观体验	自选样本，不是全球随机抽样
高	原始产品使用数据	Anthropic Economic Index	AI 实际任务结构、自动化/增强比例	只代表 Anthropic 用户，厂商存在利益相关
中高	专业市场研究	Gartner	IT、软件、AI 市场支出	属于市场估算/预测，而非财务普查
中	委托行业研究	WP Engine WordPress Economy study	WordPress 经济规模粗略参照	“经济贡献”不是插件/主题销售收入
情景层	本文综合估计	上述证据 + benchmark + WordPress 案例	2026—2032+ 自动化率	对模型突破、监管、成本等假设敏感

尤其需要强调三个测量陷阱。

第一，**代码生成率不等于端到端替代率**。模型可以写出一个函数的绝大部分代码，但工程师仍需定义需求、确认 API、处理边缘条件、审查安全性、迁移数据库、测试旧版本、发

布和维护。因此本文后面给出的“替代率”主要指**可由 AI 与传统自动化完成并通过生产验收的任务工时比例**，而不是“AI 生成了多少行代码”。

第二，**benchmark** 不等于生**产环**境。OpenAI 已在 2026 年公开指出，SWE-bench Verified 随着训练数据污染、测试质量等问题，区分前沿模型的价值下降，并推动使用更困难、更接近真实工程的 SWE-Bench Pro。这也是为什么本文不会把某模型在 SWE-bench 上的 80% 分数解释为“80% 软件工程师可以被替代”。

第三，WordPress 的某些经济数据并不存在统一官方口径。尤其是 **WordPress 开发者数量、插件/主题总销售收入、收入分布、年度插件下架率、年度主题下架率**，均没有发现 WordPress 官方公开的统一统计。本文严格标记为 **“未指定/无公开数据”**，而不是用第三方估算填空。

核心原始/权威资料可直接查阅：

来源	直接链接	关键用途
Gartner 全球 IT 支出	Gartner 2026 IT Spending Forecast	全球软件市场规模
Gartner 全球 AI 支出	Gartner 2026 AI Spending Forecast	AI 软件、AI 基础设施规模
WordPress 插件目录	WordPress Plugins	当前插件规模
WordPress 主题目录	WordPress Themes	当前主题规模
WordPress 插件团队	Make WordPress Plugins	提交、批准、AI 审核数据
Stack Overflow 开发者调查	2025 Developer Survey	AI 使用、信任、Agent 使用情况
METR 开发者生产率实验	Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity	随机对照真实开发实验
Anthropic Economic Index	Economic Index, January 2026	AI 任务结构与自动化模式
GitHub Octoverse	Octoverse 2025	开发者与开源活动规模
欧盟 AI Act	European Commission AI transparency guidance	监管风险监测

软件行业正在发生什么

最重要的宏观数据并不是“软件市场萎缩”，恰恰相反，是**软件和 AI 投入同时增长，而生产软件的边际成本正在下降**。

指标	时间	数据	变化/解释	证据
全球软件支出	2025	**1.271 万亿美元**	Gartner 最新口径	
全球软件支出	2026	**1.468 万亿美元**	**+15.5%**	

指标	时间	数据	变化/解释	证据
全球 IT 总支出	2026	**6.369 万亿美元**	**+14.2%**	
全球 AI 总支出	2025	**1.787 万亿美元**	Gartner AI 口径包含基础设施等	
全球 AI 总支出	2026	**2.670 万亿美元**	**+49.5%**	
全球 AI 软件	2025	**2882 亿美元**	—	
全球 AI 软件	2026	**4616 亿美元**	约 **+60.2%**，据官方数值计算	
全球 AI 软件	2027 预测	**6564 亿美元**	仍快速增长	
GitHub 开发者账户	2023	突破 **1 亿**	平台开发者 proxy，不等于职业人口	
GitHub 开发者账户	2025	**1.8 亿+**	单年新增 **3600 万+**	
美国软件开发/QA/测试职业组	当前 BLS 页面	约 **190.5 万岗位**	BLS 仍给出高于总体平均的增长展望	

GitHub 的数据尤其值得注意。2025 年超过 **1.8 亿开发者**在平台上构建软件，全年新增超过 **3600 万**；GitHub 同时报告 2025 年约 **10 亿次 commit**，月均合并约 **4320 万个 pull request**。这并不能证明 AI 创造了这些增长，但至少与“AI 已导致软件开发活动系统性崩塌”的假说不一致。GitHub 还报告，接近 **80% 的新开发者在加入 GitHub 第一周使用 Copilot**，说明 AI 工具正从“高级外挂”变成新开发者默认工具链的一部分。

从供需角度看，更合理的模型是：

AI 降低单位软件功能的生产成本 → 更多人可以生产软件 → 软件和功能供给暴增 → 普通功能价格下降 → 用户预期提高 → 软件需求进一步扩展 → 价值向复杂问题、信任、数据、集成、运维和分发迁移。

这是一种基于 Gartner 软件支出、GitHub 开发活动和 WordPress 插件供给变化的经济推断，而不是已经得到充分验证的宏观因果定律。

AI 对开发效率的证据则明显比营销口径复杂。

证据	结果	真正说明什么
Stack Overflow 2025	**84%** 使用或计划使用 AI；职业开发者 **51% 每天使用**	AI 开发已进入主流
Stack Overflow 2025	**46%** 不信任准确性，**33%** 信任	adoption ≠ trust
Stack Overflow 2025	**66%** 抱怨结果“几乎正确”；	验证成为新瓶颈

证据	结果	真正说明什么
	45% 认为调试 AI 代码更耗时	
Stack Overflow Agent 用户	约 **69%** 认为 Agent 提升个人生产率；约 70% 认为特定任务耗时下降	已采用者主观收益明显，但存在选择偏差
METR RCT	经验丰富开发者在熟悉代码库上使用早期 2025 AI，实际约 **慢 19%**	复杂成熟工程存在上下文、检查、整合成本
Anthropic 任务估算	被提交给 Claude 的任务模型估算平均时间下降可接近 **80%**	更接近“适合交给 AI 的局部任务”上限；非端到端因果结果
Anthropic Economic Index	2025 年 11 月 Claude.ai 使用中约 **52% 为增强、45% 为自动化**	实际使用仍不是“纯自动替代”

Anthropic 的实际使用数据还显示，计算机和数学类工作在 Claude.ai 中占约 **34%**、在第一方 API 使用中占约 **46%**；其中“修改软件以修复错误”本身就是重要任务类别。与此同时，Anthropic 观察到任务越复杂、按人类完成时间估计越长，模型成功率整体越低。这与 METR 的真实代码库实验方向一致：**规模扩大、上下文增加、反馈周期变长之后，AI 的错误累积和验证成本会迅速变重要**。

前沿 Code LLM 的能力进步又是真实且非常快的。下面不是“模型排行榜”，而是几个有代表性的公开里程碑：

时间	模型/报告	软件工程 benchmark	结果	解读
2024-08	GPT-4o	SWE-bench Verified	**33.2%**	当时官方 scaffold 结果
2025-01	Claude 3.5 Sonnet	SWE-bench Verified	**49%**	Repo-level debugging 明显进步
2025-05	Claude Opus 4	SWE-bench Verified	**72.5%**	Agentic coding 快速提升
2025-08	GPT-5	SWE-bench Verified	**74.9%**	同一代基准继续逼近饱和
2026-02	Claude Opus 4.6	SWE-bench Verified	**81.42%**（特定 prompt 设置）	老 benchmark 已接近失去区分度
2026-04	GPT-5.5	SWE-Bench Pro	**58.6%**；Terminal-Bench 2.0 **82.7%**	换更难基准后重新出现明显余量

因此，最值得开发者牢记的是：**benchmark 曲线可以指数式上升，而真实端到端工程生产率不会按相同比例上升。** OpenAI 自己关于 SWE-bench Verified 污染与测试质量的警告，就是为什么必须同时观察真实任务实验，而不能只看 headline score。

WordPress 插件与主题生态：一个提前发生的未来

WordPress 的价值在于，它同时具备成熟平台、低开发门槛、巨大安装基数、开放市场和大量中小开发团队，因此比大型企业软件更早暴露“AI 降低软件供给成本”后的竞争形态。WordPress 官方称其运行于超过 **43% 的 Web**。

生态规模与数据可得性

指标	公开数据	判断
WordPress 覆盖	**>43% 的 Web**	巨大的潜在用户池
WordPress.org 当前插件目录	**73,000+ 免费插件**	当前目录首页口径
2025-08 Plugins Team “published”口径	**60,187 个插件**	与当前目录数字口径明显不同，不能直接相减计算净增量
当前主题目录宣传口径	**15,000+ 免费主题**	官方目录页面
同一主题页面当前结果栏	约 **8,651 themes**	与“15,000+”存在口径差异，因此不宜当作同口径存量
WordPress 专职/兼职开发者总人数	**未指定/无公开数据**	不应由 GitHub/PHP 人口直接外推
插件+主题年度直接销售额	**未指定/无公开数据**	WordPress.org 不披露商业插件销售
插件/主题收入分布	**未指定/无公开数据**	安装量不能等同收入
插件年度下架率	**未指定/无公开数据**	无统一年度官方 churn 序列
主题年度下架率	**未指定/无公开数据**	无统一年度官方 churn 序列

WP Engine 委托的 WordPress Economy 研究曾估计 WordPress 经济生态在 **2020 年约 5967 亿美元、2021 年约 6360 亿美元**。这是一项有价值的生态经济影响估算，但必须避免将它写成“插件/主题收入”，也不能与 Gartner 的软件销售额直接相加；两者测量对象完全不同。

插件安装量显示出明显的头部现象：WordPress.org 当前热门插件中，Elementor、Yoast SEO、Contact Form 7 等达到千万级活跃安装，WooCommerce 达数百万级。不过，免费活跃安装量无法推导厂商 ARR、利润或市场收入集中度，所以本文不制造所谓“Top 10 占 X% 收入”的数字。

插件供给正在加速

WordPress Plugins Team 的数据是本研究最有价值的案例证据之一。

指标	2024/2025 数据	含义
2025 全年审核插件	**12,713**	同比 **+40.6%**
2025 批准插件	**5,415**	2024 为 **3,259**，同比约 **+66.2%**
2025 参与后续审核流程的插件	**7,882**	作者实际继续响应的项目池
批准 / 全部被审核	**约 42.6%**	本文按 5,415 ÷ 12,713 计算
批准 / 继续参与审核者	**约 68.7%**	本文按 5,415 ÷ 7,882 计算；官方文字给出的比例约 69.5%，存在轻微口径差异
被审核后作者不再回复	**38.7%**	说明提交量并不等于可维护产品量
2025-08 YTD 新提交	**7,670**	同期同比约 **+87.3%**
新提交周均	约 **235/周**，上年约 124/周	当时供给增速接近翻倍
2025 年末附近	约 **330/周**，前一年约 150/周	高提交量继续维持
首次审核由算法/AI 自动启动	**85.3%**	审核端本身也被 AI 自动化

WordPress 团队在 2025 年的官方文章中明确说，新插件提交同比增长约 **87%**，从此前数月已经观察到上升，并提到可以观察到 AI 的影响；增长类别包括聊天机器人、内容生成、WooCommerce、电商、SEO、多媒体生成、表单、摘要、FAQ、翻译、分类与 metadata 等。

这很可能预演了软件行业更广泛的供给曲线：**当一个以前需要三天的插件原型可以在数小时内完成时，市场并不会只得到“原来的插件数量、但更便宜”，而会得到大量过去根本不值得开发的插件。**

其后果是两面的：

积极面是长尾需求被满足。一个只有数百名用户的微型需求，以前不值得专业团队投入，现在个人开发者借助 AI 就可能盈利。

负面面是供给竞争极大增强。“一个功能”本身的护城河急剧下降；安全、维护、兼容性和品牌验证成本相对上升。WordPress Plugins Team 自身不得不持续强化自动 scanner、Plugin Check 和 AI 辅助审核，这恰好说明**代码产生得越便宜，验证和治理越重要**。

上架、下架与主题数据

插件的“上架率”最接近的官方可计算指标，是审核后的批准率。按 2025 年全年数据，批准数占所有被审核插件约 **42.6%**；若只计算真正继续完成审核流程的作者，约 **68.7%**。两者适合回答两个不同问题：前者描述“提交进入审核后最终成功发布的机会”，后者描述“愿意完成整改流程的作者成功率”。

但**插件年度下架率没有公开统一官方数据**。WordPress 的 reviewer 文档同时指出，现有插件通常允许保留，并不会因为普通原因直接删除，除非出现极端情况，因此不能把目录数量变化简单等同为“下架数量”。

主题方面，Make WordPress Themes 的一个 2026 年周度样本显示：7 天内关闭 **1,001 个 ticket**，其中 **957 个 made live**，包括 **37 个新主题和 920 个主题更新**，另有 **44 个 ticket 未批准**。如果机械地用 957/1,001，可得到约 95.6%，但由于绝大部分是**更新而非新主题**，这个数字**绝不能称为新主题上架率**。年度新主题批准率与下架率仍应标注 **“未指定/无公开数据”**。

因此，一张严格的数据缺口表反而比伪精确数字更有意义：

WordPress 指标	是否可可靠获得	本报告处理
插件存量	是，但不同时点/页面可能口径不同	报告原始值，不强行做净增长
插件审核/批准量	是	使用 Make WordPress 原始统计
插件上架成功率	可近似计算	同时给“全部审核”和“参与整改”两个分母
插件下架率	否	**未指定/无公开数据**
主题目录规模	有，但页面存在不同口径	明确口径冲突
主题新上架率	无完整年度口径	**未指定/无公开数据**
主题下架率	否	**未指定/无公开数据**
WordPress 开发者数量	否	**未指定/无公开数据**
插件/主题总收入	否	**未指定/无公开数据**
插件/主题收入集中度	否	**未指定/无公开数据**

自动化替代能力、工具分工与演进时间线

必须先定义“替代”。这里将其定义为：

> 任务工时自动化率：在需求足够明确、有可执行测试/验收标准、允许 AI 使用代码库与开发工具、且仍保留必要人工责任签核的情况下，原本由开发者完成的某类工程任务工时中，可由 AI + 自动化系统完成并达到生产验收水平的比例。

它**不等于岗位替代率**，也不等于 AI 写出的代码行比例。

按复杂度估算

下表是本报告结合 METR、Anthropic、Stack Overflow、SWE-bench 演进和 WordPress 实际审核自动化所做的**情景估计，不是已观测统计值**。

任务层级	WordPress / 软件例子	2026 端到端任务工时可自动化	2027–2028 情景	2029–2031 情景	2032+ 情景	当前主要瓶颈
简单功	shortcod	**45–70%**	**60–85%**	**75–95%**	**85–98%**	规格、

任务层级	WordPress / 软件例子	2026 端到端任务工时可自动化	2027–2028 情景	2029–2031 情景	2032+ 情景	当前主要瓶颈
能	e、简单 block、CRUD、CSS、表单字段、API wrapper、测试样板、文档					规范条件、验收
中等复杂	WooCommerce 扩展、多文件功能、第三方 API、支付流程、复杂 Gutenberg block、数据迁移	**20–40%**	**30–55%**	**45–75%**	**60–90%**	跨文件状态、兼容性、真实环境测试
架构级	大型重构、权限体系、高可用、安全模型、性能设计、复杂迁移、长期架构	**5–20%**	**10–30%**	**20–45%**	**30–65%**	隐性需求、长期权衡、责任、系统历史
生态级	产品战略、插件组合、合作伙	**0–10%**	**5–15%**	**10–25%**	**15–40%**	人类偏好、信任、政治/商业判断、

任务层级	WordPress / 软件例子	2026 端到端任务工时可自动化	2027–2028 情景	2029–2031 情景	2032+ 情景	当前主要瓶颈
	伴、平台政策、定价、客户发现、社治理					分发

这些区间故意比“AI 写代码速度”保守得多。原因是同一时期，我们既观察到了 Claude 对被选择任务可能产生极大时间节省的模型估计，也观察到了 METR 的成熟代码库 RCT 中 **19% 的负生产率结果**；Stack Overflow 又显示高采用与高不信任同时存在。因此，2026 年合理的判断不是“AI 能不能写”，而是**“AI 写完后，还剩多少昂贵工作”**。

简单任务的代码生成比例实际上可以明显高于表中的端到端替代率。例如，一个 AI 可能生成某插件 90% 的初始 PHP/JS 代码，但开发者仍需要处理 nonce、capability、输入 sanitization、数据升级路径、WP/PHP 版本兼容、插件冲突、性能、错误恢复和发布责任。因此“90% AI 代码”完全可能最终只对应“50% 工时减少”。

各类工具真正适合替代什么

开发环节	Copilot / ChatGPT 类助手	Repo/Code Agent	测试、静态分析、CI/CD	2026 判断
Boilerplate / CRUD	**强**	**强**	验证	高度商品化
CSS/前端小修改	**强**	**强**	UI regression 可辅助	高度商品化
API 使用与样例	**强**	**强**	Contract test	高度加速
单元测试生成	**强**	**强**	**核心执行层**	测试生成便宜，测试设计仍重要
Bug 定位	中强	**强**	Logs/tests 提供 oracle	越有可复现测试越适合 AI
多文件 feature	中	中强	**重要**	已实用，但需要审查
Legacy refactor	中	中	**非常重要**	隐式行为和回归风险高
安全修复	中	中强	SAST/DAST/scanner 必须配合	不宜让模型独立签核
数据迁移	中	中	Migration test/rollback	成本高
架构	可做方案生成	可做 repo 分析	无法决定商业	AI 是顾问，不

开发环节	Copilot / ChatGPT 类助手	Repo/Code Agent	测试、静态分析、CI/CD	2026 判断
			权衡	宜是最终责任者
CI/CD	生成配置很强	可修 pipeline	**传统自动化本身最强**	大量低价值人工操作会消失
产品发现	可总结	较弱	不适用	客户接触仍是核心
定价与定位	可模拟分析	较弱	不适用	AI 无法替你拥有市场
生态合作	辅助沟通	弱	不适用	人际信任难自动化

GitHub 早在 2019 年已经把 Actions 扩展到 CI/CD；这提醒我们，软件工程自动化并不始于大模型。真正强大的未来工作流更可能是 **LLM 负责生成/推理 + 确定性工具负责验证 + CI/CD 负责执行 + 人类负责目标和责任**。GitHub 2024 年还报告，其生态的 secret scanning 检测到超过 **3900 万次 secret 泄漏**，说明传统静态与规则化自动化在 AI 时代并没有过时，而是成为 AI 的“硬约束层”。

演进路径

以下时间线中，2021—2026 属于观测到的行业演进；2027 之后是本文情景而非确定预测。AI 使用率、代码 benchmark、GitHub 与 WordPress 供给变化的依据分别来自 Stack Overflow、模型厂商官方 benchmark、GitHub 和 WordPress。

```
timeline
  title AI 驱动的软件开发演进
  2019 : CI/CD 与规则型自动化已经成熟
  2021 : AI Pair Programmer 进入开发工作流
  2022-2023 : 对话式 LLM 降低编程知识获取门槛
        : 自然语言开始成为新的开发接口
  2024 : Repo 级 Bug 修复能力明显增强
     : AI 开源项目与开发者活动快速增长
  2025 : AI 开发工具进入大众采用阶段
     : Coding Agent 开始执行多文件任务
     : WordPress 新插件供给显著加速
  2026 : Agent + Terminal + CI 开始形成闭环
     : 旧代码 Benchmark 趋于饱和
     : 验证、安全、评估成为主要瓶颈
  2027-2028 : 简单功能开发大面积自动化
        : 小团队吞吐量显著提高
        : 普通功能价格继续下降
  2029-2031 : Agent 可持续处理更长的软件任务
        : 中等复杂功能逐步 Agent-first
```

：架构与治理仍以人类为中心
2032+：编码本身趋向基础设施化
：价值集中于问题、数据、信任、分发与责任

这条演进路线最重要的分水岭并不是“AI 会不会写完整插件”，因为这件事已经越来越常见；真正的分水岭是：

AI 能否独立证明自己写的东西在长期、复杂、开放环境中是正确的。

一旦一个任务拥有非常好的自动化测试、清晰 specification、稳定 sandbox、可逆部署以及明确 reward，Agent 的自动化速度会很快；反之，如果“正确”需要用户判断、历史背景、跨团队协调、市场洞察或承担法律责任，自动化进展就会慢很多。Anthropic 观察到复杂任务成功率随任务长度下降、METR 真实工程实验出现负生产率，正好说明这个边界。

未来真正稀缺的技能、角色、产品与商业模式

未来软件行业的一个反直觉趋势是：**会写代码的人可能继续增加，但“仅仅会写代码”的市场稀缺性会下降。** GitHub 2025 年新增超过 3600 万开发者账户，同时 AI 让自然语言成为越来越有效的开发入口。

稀缺性的排序可以这样理解：

优先级	未来稀缺资产	为什么越来越值钱	WordPress 对应机会
极高	**领域知识 + 问题所有权**	AI 可生成实现，却很难知道真正值得解决的问题	医疗、教育、电商运营、会员业务、本地服务等垂直插件
极高	**客户关系与分发**	软件供给爆炸后，“被发现”和“被信任”比“被开发”更难	品牌、SEO、合作伙伴、Agency、Hosting 渠道
极高	**验证与可靠性工程**	AI 生成越多，错误检测成为瓶颈	PHPUnit、Playwright、Plugin Check、compatibility farm、监控
极高	**安全与责任能力**	攻击面和 AI 生成代码同步增加；错误成本不下降	支付、身份、权限、数据保护、安全插件
高	**系统架构与长期维护**	局部最优代码很容易，跨年度演进困难	大型 WooCommerce、会员、LMS、企业 WordPress
高	**复杂生态集成能力**	多系统边界很难通过单一 prompt 商品	Woo、CRM、ERP、支付、物流、邮

优先级	未来稀缺资产	为什么越来越值钱	WordPress 对应机会
		化	件、CDP
高	**数据资产与工作流资产**	模型本身日益普及，独有上下文更稀缺	客户 workflow、行业 taxonomy、benchmark/eval 数据
高	**AI Agent / Eval 工程**	企业真正缺的是可控自动化，而非 demo	WordPress 运维 Agent、迁移 Agent、QA Agent
高	**产品判断和定价**	用户最终买的是结果，不是代码量	垂直 SaaS + plugin、按站点/用量收费
中高	**性能与成本优化**	AI 更容易制造“能运行但昂贵”的系统	Hosting、缓存、数据库、Core Web Vitals
中高	**开源维护与社区治理**	生成代码容易，维护公共依赖难	WP Core、Woo、公共框架贡献

未来更危险的产品类型是：**单一小功能、可在一段 prompt 中完整描述、几乎没有专有数据、没有网络效应、没有客户迁移成本、没有长期服务要求的插件。**

例如，单纯“新增一个 shortcode”“给后台加一个字段”“生成 meta description”“把一种公开 API 包装成 WordPress 页面”“新增若干 Gutenberg blocks”等产品，不会完全消失，但其价格和利润率更容易受到 AI 供给冲击。WordPress 2025 年插件提交量快速增长正是这一趋势的早期信号。

相对更强的产品是：

插件 + 持续服务。 例如 WordPress connector 只是入口，真正产品是云端搜索、备份、监控、合规、AI workflow 或数据同步。

插件 + 专有数据。 例如行业目录、税务规则、商品数据、风险规则、用户行为 benchmark。

插件 + 深度集成。 同时可靠支持 WooCommerce、支付、CRM、ERP、物流和多语言。

插件 + SLA/责任。 用户购买的不是“代码存在”，而是“这个关键业务流程明天仍然工作”。

插件 + 社区/网络。 供需市场、agency 网络、模板生态、第三方 extensions。

插件 + 迁移/运营服务。 AI 可以生成代码，但将一个运行五年的真实客户系统安全迁移，仍具有大量隐式知识和责任成本。

从商业模式看，长期更防御性的排序大致是：

商业模式	AI 时代防御力	原因
托管服务 + 订阅 + Plugin connector	**很高**	持续提供运行、数据、模型、监控与结果
垂直 SaaS + WordPress 集成	**很高**	价值来自业务 workflow，

商业模式	AI 时代防御力	原因
		不是插件代码
企业订阅 + SLA + 实施	**很高**	客户购买可靠性与责任
API / 数据订阅	**高**	专有数据和实时服务难以复制
Pro Plugin 年订阅	**中高**	必须通过更新、支持、兼容性不断提供价值
Freemium → Pro	**中高**	免费层承担获客，但需要强差异化
单次 Lifetime License	**中低**	持续维护成本与一次性收入错配
一次性简单插件	**低**	克隆成本趋近于零

Gartner 2026 年也观察到软件厂商正把 Agentic AI 嵌入既有产品，以防御跨功能 Agent；这进一步支持"AI 不是单独的新产品类别，而会逐渐成为每一个软件产品的基础能力"这一方向。

个人开发者与小团队的可操作路线图

对个人开发者和 2—10 人的小团队来说，最大的错误不是"不用 AI"，而是**只用 AI 提高代码产量，却没有同步提高验证能力、产品判断和获客能力**。AI 会让这种团队生产更多代码，也可能让它们更快制造技术债。

推荐采用"AI 提升吞吐 + 测试守住质量 + 垂直领域建立壁垒 + 经常性收入承接价值"的路线。

周期	技能与工程	产品定位	商业与生态	建议量化里程碑
未来 30–90 天	熟练掌握至少一种 Agent/IDE；建立 PHPUnit、WP-Env、Playwright、PHPCS/PHPStan、Plugin Check、CI	删掉"一句话 prompt 就能复制"的 idea；锁定一个垂直痛点	做客户访谈，不先堆 feature	完成 **20 次客户访谈**；拿到 **3 个 design partner**；所有 PR **100% 过 CI**
未来 3–6 个月	建立 AI-assisted issue→code→test→review 流程；记录 AI 接受率、返工率	从 feature 变 workflow，至少解决完整业务流程的一段	建立官网内容、affiliate/agency/hosting 合作	Critical path 自动测试覆盖 **>80%**；PR lead time 比基线下降 **30%** 且 escaped defects 不增加
未来 6–12 个月	学 security、性	建立 2–3 个难复	从单次授权转	**30–50 个付费客**

周期	技能与工程	产品定位	商业与生态	建议量化里程碑
	能、可观测性、数据迁移与 AI eval	制的深度 integration	annual subscription / service	**户**或达到自身可持续 MRR；月 churn **<3–5%** 为目标；合作渠道贡献 **>10%** 新线索
未来 12–24 个月	Agent orchestration、sandbox、回归测试矩阵、模型 fallback	plugin → platform / managed service	建 partner ecosystem，降低单一 WordPress.org/SEO 渠道依赖	**>50% 重复运维工作自动化**；合作渠道带来 **>20%** leads；关键事故 MTTR 持续下降
未来 24–36 个月	成为“AI 系统负责人”而非纯 coder	专有 workflow、数据、企业能力形成护城河	企业计划、SLA、API、数据业务	经常性收入占比目标 **>70%**；任何单一获客渠道占比尽量 **<60%**；建立至少一种模型/vendor fallback

这里的百分比是**管理目标而非行业 benchmark**，应根据产品成熟度调整。真正值得追踪的不是“AI 写了百分之多少代码”，而是：

[= {}]

一个团队如果 AI 生成代码比例从 20% 上升到 80%，但生产事故翻倍、support tickets 上升 50%，那不是 AI transformation，而是把编码瓶颈转移成了 QA 和客服瓶颈。Stack Overflow 发现 66% 的开发者都遇到“几乎正确”的 AI 输出，正说明应把**返工率和 defect escape rate** 纳入 AI ROI，而不是只统计 completion 或 token 数。

对于 WordPress 开发者，我建议把技能栈从传统的：

> PHP + JS + CSS + WordPress API

升级为：

> **WordPress 深度能力 + AI Agent + 自动化测试 + Security + Cloud/API + Product/Domain + Distribution**

其中最容易被忽略的是测试。未来 Agent 能不能真正取代任务工时，很大程度取决于系统有没有“机器可判定的正确答案”。测试不是 AI 时代的附属技能，而是**把 AI 从聊天助手升级为自动开发者的前提条件**。METR 的真实工程结果与 Anthropic 的复杂度结果都说明，上下文和验证才是 repo 级自动化的关键约束。

个人开发者可以采用下面的具体开发闭环：

```
flowchart LR
  A[客户问题/生产数据] --> B[人工定义需求与验收标准]
  B --> C[AI 拆任务与生成方案]
  C --> D[Agent 实现代码]
  D --> E[单元/集成/E2E 测试]
  E --> F[静态分析/安全扫描]
  F --> G{全部通过?}
  G -- 否 --> C
  G -- 是 --> H[人工 Review 关键差异]
  H --> I[Staging/Canary]
  I --> J[生产监控]
  J --> K[真实用户反馈]
  K --> A
```

这个流程中的长期竞争优势不是 D“写代码”本身，而是 **A、B、E、F、J、K**：真实客户问题、正确 specification、可靠测试、安全约束、生产数据和反馈闭环。代码模型越强，这些环节相对越值钱。

对于已经经营 WordPress 插件的团队，还应立即给产品组合做一次“AI 商品化压力测试”：

问题	危险信号	強势信号
用户能否用一个 prompt 描述整个产品？	能	需求有大量隐式行业规则
模型是否能在一天内做出 80% clone？	能	真正价值在后台服务/数据/网络
产品价值是否主要是 UI 和 CRUD？	是	产品管理关键业务 workflow
是否依赖持续更新？	很少	必须持续适配平台/API/法规
是否有专有数据？	没有	有历史数据、benchmark、模型 feedback
用户切换成本？	极低	数据迁移、workflow、团队习惯形成粘性
用户为何付钱？	“省了写代码”	“降低收入损失/风险/人工成本”
AI 写错后谁承担责任？	没人在乎	客户愿意为可靠供应商付费

若左栏危险信号占大多数，产品未来最好的战略通常不是“继续增加 20 个小功能”，而是把产品向 **workflow、service、integration、data、trust** 中至少两项迁移。

风险、不确定性、监测指标与结论

任何到 2030 年以后软件自动化率的预测，都存在很大误差。最重要的不确定性不是“下一代模型 benchmark 会不会涨”，而是模型能力能否从短任务迁移到**长周期、开放世界、高责任任务**。

关键风险与需要监控的指标

风险	可能使 AI 冲击加速	可能使冲击放缓	应监控指标
模型能力	长任务可靠性突然突破、Agent 自我纠错增强	长上下文仍频繁失效、错误相关性高	SWE-Bench Pro、Terminal-Bench、真实内部任务成功率、一次通过率
验证能力	自动测试/eval 成本下降	oracle 本身难建立	AI PR revert rate、escaped defects、人工 review 分钟/PR
模型成本	inference/token/agent 成本继续下降	GPU、电力、供应限制推高成本	每个“验收通过任务”的模型成本，而非每百万 token
开源模型	开源逼近闭源，私有部署成熟	训练成本/可限制	同一内部 eval 的开源/闭源质量差、推理成本
监管	规则清晰，合规框架标准化	地区碎片化、数据与版权约束增强	EU/US/日本 AI 规则、数据跨境要求、AI disclosure
WordPress 平台政策	自动审核、API、Agent 接口进一步开放	审核规则、商业政策、目录政策变化	submission queue、approval rate、Plugin Check 新规则、目录政策
安全	自动扫描和自动修复成熟	AI 大幅提高攻击规模	CVE/事故率、secret leak、依赖漏洞、AI 生成代码事故率
开发者劳动力	单人吞吐不断上升	软件需求同步扩大吸收生产率	GitHub 开发者增长、招聘数、junior/senior 比例、BLS 数据
市场竞争	AI 导致插件数量快速增长	用户对品牌/安全的要求提高	同类新插件/季度、CAC、价格、免费→付费转化
平台集中度	Agent 直接绕过 UI/传统插件	平台生态继续保留入口权力	AI Agent 流量 vs search/direct traffic、WordPress.org 获取占比

监管已经不再只是远期风险。欧盟 AI Act 自 2024 年生效后分阶段适用，**Article 50 的特定透明度义务从 2026 年 8 月 2 日起适用**，涉及部分交互式/生成式 AI 系统及 AI 生成内容的透明度要求。开发 AI WordPress 插件的厂商需要根据自身是 provider、deployer 还是下游集成方判断具体责任，而不应假定“只是调用 OpenAI/Claude API，所以法规与自己无关”。具体适用性需要法律专业判断。

平台风险也值得重视。WordPress Plugins Team 已经把算法和 AI 放进审核流程，并不断扩展 Plugin Check 与扫描能力；这对生态安全是好事，但也意味着一个完全依赖 WordPress.org 免费流量的开发商，本质上把分发权交给单一平台。

模型风险则需要防止另一个极端：因为 METR 目前曾发现负生产率，就认为 AI Coding 是泡沫。这个结论同样不成立。SWE-bench 从 2024 到 2026 的能力变化、WordPress 新插件供给和 GitHub 的 AI 使用数据都显示技术边界正在迅速移动。合理策略不是押注“AI 永远做不到”，而是建立一个即使模型继续快速变强也会升值的职业资产组合。

最终可以把软件行业未来的价值链浓缩为：

[+++]

AI 时代逐渐变成：

[+ + + + + +]

其中粗体部分恰恰是未来最值得开发者投资的方向。

对 WordPress 插件/主题行业而言，我的基准判断是：**插件不会因为 AI 而消失，但“插件代码”会越来越像一种廉价原材料。** AI 会制造更多插件，而不是更少；因此目录更加拥挤、简单产品生命周期缩短、免费替代品增加，代码本身更难形成护城河。与此同时，可靠、安全、业务关键、数据丰富、深度集成、有品牌和持续服务的 WordPress 产品，其相对价值反而会上升。2025 年插件提交量与审核自动化同时激增，已经是这一未来的早期实证。

对软件职业本身，最可能发生的不是某一天“AI 替换所有程序员”，而是**同等规模团队需要更少的纯实现工时，却可以管理更多软件、更多功能和更多客户**。当软件市场仍以两位数速度增长、GitHub 开发人口仍高速扩张时，岗位数量究竟下降还是需求扩张吸收生产率提升，仍无法由现有数据确定。最稳健的个人策略 therefore 不是预测哪一年 AGI 到来，而是让自己的价值不再与“手写一段代码需要多少小时”绑定。

可以立即执行的五件事：

- **本周就建立 AI 开发基线。** 记录当前 feature lead time、PR review 时间、生产缺陷、support tickets 和每月研发工时；未来所有 AI 工具都用这些指标证明 ROI，而不是用“生成代码行数”证明。Stack Overflow 的数据已经说明返工是 AI 使用中的主要隐性成本。
- **未来一个月完成至少 20 次目标客户访谈。** 不要再优先寻找“我能写什么插件”，而要寻找“哪些昂贵、反复发生的业务问题即使代码免费，客户仍愿意付钱解决”。这将领域知识和客户渠道变成你的护城河。
- **在继续提高 AI 代码占比之前，把测试、静态分析、安全扫描和 CI 变成强制 gate。** 对 WordPress 项目至少建立关键路径的自动化测试、Plugin Check、代码规范/静态分析和真实版本兼容测试。模型越强，验证层的投资回报越高。

- 把品 “ 功能件 ”向“果 **+ 持续服务”移动。** 优先选择支付、数据同步、迁移、安全、性能、合规、垂直 workflow、深度 API integration 等需要持续维护的领域，并尽可能建立订阅、托管服务、数据/API 或 SLA 收入。
- **同时减少平台和模型单点依赖。** 至少拥有自己的邮件/客户列表和官网获客渠道，并为核心 AI 功能保留第二模型或可降级路径。WordPress 政策、模型供应商、价格和 AI 法规都在快速变化；真正稳健的小团队应该能够在其中任何一个发生变化时继续服务客户。