

# 2025-06-29 LLM 应用场景的再思考——从代码辅助的商业成功到管理变革的真正潜力-正文

Gemini

## 从代码到指令：对大型语言模型应用中的路径依赖、技术对齐和社会学障碍的分析

### 摘要

大型语言模型（LLM）的出现正深刻地重塑知识工作的边界，其中以代码辅助工具（如 GitHub Copilot）的广泛普及最为引人注目，并已成为行业主导范式。本文旨在挑战一个普遍存在的假设，即此种主导地位是技术与任务最优对齐的必然结果。我们提出一个核心论点：LLM 在编程领域的统治地位，是一个由精准的市场进入策略、随之产生的路径依赖以及深刻的社会学因素共同塑造的产物。本文通过分析揭示了一个根本性的“训练-应用悖论”：使代码成为理想训练数据的确定性结构，恰恰与 LLM 概率性输出应用于生产环境的低容错性要求相冲突。与此相对，本文论证了 LLM 的内在属性（如概率生成、信息整合）与事务性管理工作的高容错性、信息密集型特征存在更天然的技术对齐。然而，这一巨大潜力因组织内部的身份威胁、变革管理的系统性失败以及主导技术叙事的社会建构而受到系统性阻碍。本文的分析框架整合了技术采纳理论、组织社会学和技术社会建构论（SCOT），不仅解释了当前 LLM 应用路径的成因，也揭示了其战略性错配的后果。最后，本文提

出，未来的发展方向应从“替代”范式转向“增强”范式，通过构建以人为中心的、可解释的管理 AI 系统，并重新定义管理者的角色，从而解锁 LLM 在推动深层次组织变革中的真正潜力。

**关键词：**大型语言模型（LLM），代码辅助，路径依赖，技术社会建构论（SCOT），组织变革，管理自动化，技术对齐

### 1. 引言

在第四次工业革命的浪潮中，大型语言模型（LLM）已成为最具变革潜力的技术之一。从 OpenAI 的 GPT 系列到 Google 的 Gemini，这些模型在自然语言理解与生成方面的卓越能力，预示着对知识工作领域的根本性重塑。在众多潜在的应用场景中，代码辅助工具（如 GitHub Copilot）率先取得了巨大的商业成功和市场渗透，以至于在公众和企业的认知中，“AI 赋能工作”在很大程度上等同于“AI 辅助编程”。行业报告和早期研究普遍宣称这类工具能显著提升开发者生产力，例如任务完成速度提升高达 55%（GitHub, 2022）<sup>10</sup>。

然而，这种压倒性的主导地位引出一个关键的、但

常被忽视的问题：当前以代码为中心的应用路径，是反映了技术与任务之间的最优对齐，还是一个由历史、商业和组织因素决定的特定演化结果？换言之，我们正在见证的是一场由技术天性驱动的效率革命，还是一个被早期成功所锁定的、可能并非最优的战略选择？

本文旨在挑战前者这一普遍存在的、具有技术决定论色彩的假设。我们认为，将 LLM 的成功应用等同于代码辅助，是一种对现象的简化解读，它掩盖了技术本质、组织动力学和市场力量之间复杂的相互作用。本文的核心论点是，LLM 在编程领域的成功，首先是一次卓越的市场进入策略，它巧妙地利用了代码数据的独特性质和开发者群体的专业能力。但这种成功不仅掩盖了 LLM 的概率性本质与软件工程确定性要求之间的根本性张力，更重要的是，它通过强大的路径依赖和叙事建构，将大量的资本、人才和想象力锁定在了一个“渐进式改良”的领域，从而可能错失了其在其他领域引发“颠覆性变革”的更大机遇。

我们进一步论证，一个更具潜力、技术对齐更优的领域——事务性管理——却因深刻的组织和社会学障碍而未被充分开发。通过整合技术分析、组织理论和社会建构主义视角，本文旨在系统地回答以下问题：

1. 为何代码辅助会成为 LLM 商业化应用的第一个突破口？
2. LLM 的技术特性与编程及管理这两类任务的内在要求，分别存在何种程度的对齐与错位？
3. 哪些组织性及社会性力量阻碍了 LLM 在管理领域的应用？
4. 如何才能打破现有路径，释放 LLM 在重塑组织与管理方面的真正潜力？

为解答这些问题，本文将分步展开。第二部分将进行**文献综述**，梳理 AI 与工作未来、技术采纳理论和管理学研究的现有成果，以明确本研究的理论定位与贡献。第三部分将阐述本研究的**方法论**，即一种整合性的“理论综合与概念分析”框架。第四、五、六部分是本文的核心分析，将分别解构代码辅助的崛起、论证管理 AI 的潜力，并剖析阻碍其发展的社会学障碍。第七部分为**讨论**，将探讨本研究发现的广泛理论与现实意涵。最后，第八部分将**总结全文**，并为未来的经验研究指明方向。

## 2. 文献综述

为构建一个坚实的分析框架，本研究将立足于三个交叉的学术领域：关于 AI 与工作未来的辩论、技术采纳与组织变革的经典理论，以及关于管理角色演变的文献。

**2.1. AI、自动化与工作未来的辩论** 关于 AI 对劳动力市场影响的讨论，长期以来被两大对立的观点所主导：“替代论”（Substitution）与“增强论”（Augmentation）。前者，以 Frey 和 Osborne (2017) 的开创性研究为代表，预测大量现有工作岗位将因自动化而消失。后者则认为，AI 将更多地作为人类能力的补充和增强工具，自动化重复性任务，从而使人类能专注于更具创造性、战略性和人际交往性的工作（Autor, 2015）。尽管 LLM 的出现为“增强论”提供了新的论据，但当前的主流讨论仍存在一个局限：其分析单元大多集中在“任务”（task）层面，而较少深入探讨 AI 对组织内部的**权力结构、职业身份和管理哲学**的系统性影响。本文试图超越任务

层面的分析，探讨 LLM 如何可能重塑“管理者”这一特定角色及其在组织中的功能。

**2.2. 技术采纳与组织变革的理论视角** 对新技术的采纳与扩散，社会科学提供了丰富的理论工具。传统的技术决定论（Technological Determinism）认为技术自身的发展逻辑是社会变迁的主要驱动力。然而，这一观点早已受到广泛批判。本文将主要借鉴两大社会建构主义的理论传统：

- **技术社会建构论（Social Construction of Technology, SCOT）**：由 Pinch 和 Bijker（1984）提出，SCOT 理论强调技术的形态和意义并非预设，而是在不同“相关社会群体”（Relevant Social Groups）的互动与协商中被建构出来的。通过“阐释的灵活性”（Interpretive Flexibility）阶段的竞争，最终围绕某个特定设计或意义达成“闭合”（Closure），形成稳定的技术“人造物”（Artifact）。该理论为我们理解为何“AI 用于编程”的叙事能战胜其他可能性提供了核心分析工具。
- **路径依赖（Path Dependency）与制度理论（Institutional Theory）**：经济史学家 Paul David（1985）通过经典的“QWERTY 键盘”案例阐释了路径依赖理论，指出早期的、甚至偶然的选择，会因为“递增报酬”（Increasing Returns）和高昂的转换成本而被锁定，即便存在更优的替代方案。组织社会学中的新制度理论则进一步指出，组织的决策不仅是出于效率考量，更是为了寻求合法性（Legitimacy）。DiMaggio 和 Powell（1983）提出的“制度同构”（Institutional Isomorphism）概念——即组织在压力下变得彼此相似——通过模仿性（mimetic）、强制性

（coercive）和规范性（normative）三种机制来解释这种趋同现象。这些理论共同为我们解释 LLM 应用路径的“锁定”效应提供了坚实的理论基础。

**2.3. 管理角色的演变与领导力理论** 管理学文献对管理者的角色进行了长期探讨。James MacGregor Burns（1978）和 Bernard Bass（1985）的经典著作区分了两种领导力风格：**事务性领导（Transactional Leadership）**侧重于监督、组织和基于绩效的奖惩，是维持组织稳定运转的“管理”职能；而**变革型领导（Transformational Leadership）**则通过愿景、激励和个性化关怀来驱动组织变革与创新，是引领方向的“领导”职能。近年来，随着技术发展，关于中层管理角色“空心化”（hollowing out）的讨论不绝于耳。本文的论点建立在这一区分之上，即 LLM 的当前能力主要对准了“事务性管理”而非“变革型领导”，而这恰恰是中层管理者日常工作中耗时最长的部分。

**2.4. 研究鸿沟与本文贡献** 综上所述，现有文献已从不同角度探讨了 AI、技术采纳和管理变革。然而，鲜有研究将这三个领域的洞见整合在一个统一的分析框架内，以系统性地解释 LLM 这一特定技术在组织中应用路径的形成、锁定及其潜在的战略错位。本文的独创性贡献在于：通过结合技术特性分析、组织社会学理论和 SCOT 框架，不仅为“代码辅助为何先行”这一现象提供了多层次的社会学解释，更重要的是，它识别并论证了“管理 AI”这一被忽视的、但可能更具变革潜力的应用方向，并深入剖析了阻碍其发展的深层原因，从而为未来的研究和实践开辟了新的议程。

### 3. 研究方法：一个理论综合与概念分析的框架

鉴于本文旨在构建和论证一个新的理论视角，而非检验一个既有的经验性假设，我们采用的研究方法是“**理论综合与概念分析**”（Theoretical Synthesis and Conceptual Analysis）。此方法不涉及一手数据的收集，而是通过对现有知识体系的系统性梳理、批判性整合和概念性重构，来生成新的洞见和理论命题。

具体而言，我们的分析框架包含三个层面：

1. **技术层面**：我们深入分析 LLM 的核心技术特性，特别是其基于概率的生成机制、信息整合能力以及其固有的局限性（如幻觉、缺乏因果推理等）。
2. **组织层面**：我们将技术特性与不同工作任务的内在要求进行匹配分析。我们借鉴管理学和组织行为学理论，区分编程任务的“确定性”和管理任务的“容错性”，以评估技术-任务的对齐程度。
3. **社会学层面**：我们运用 SCOT、路径依赖和制度同构等理论，来解释特定技术应用路径如何被社会性地建构、选择和锁定，以及组织内部的权力动态和身份认同如何成为技术采纳的障碍。

我们的**证据来源**（Data Sources）是多样的，包括：

- **学术文献**：计算机科学、组织社会学、管理学和科技研究（STS）领域的同行评议论文。
- **高质量行业报告**：来自 Gartner、GitHub、Microsoft 等机构的白皮书和研究报告。

- **技术文档与专业讨论**：来自顶尖 AI 会议、开源社区和专业博客的技术分析。
- **权威媒体分析**：来自《哈佛商业评论》、《麻省理工科技评论》等媒体对 AI 应用的深度报道。

通过对这些来源进行批判性解读和交叉验证，本研究旨在构建一个逻辑严谨、论证扎实且具有理论深度的概念模型，为理解和引导 LLM 在组织中的应用提供一个全新的、更具综合性的视角，并为未来的经验研究奠定基础。

### 4. 第一部分：代码辅助的崛起——一项战略与社会学分析

本部分旨在解构 LLM 在编程领域的统治地位，论证其更多是市场进入策略和路径依赖的产物，而非技术与任务完美对齐的结果。

**4.1. 起源：作为理想初始数据集的代码** 源代码作为 LLM 的初始训练语料库，具有无可比拟的优势，这构成了其市场进入策略的坚实基础。首先，代码拥有**形式化的语法、严格的句法和逻辑结构**，与自然语言的模糊性形成鲜明对比。这种确定性结构为模型学习模式、关系和长程依赖提供了一个清晰、一致的框架（Li et al., 2025）<sup>1</sup>。其次，开源软件的爆炸式增长，尤其是在 GitHub 等平台上，创造了一个**规模庞大（PB 级别）且公开可用的训练语料库**，例如 The Stack v2 数据集就包含超过 30 亿个文件（Mockus, 2025）<sup>3</sup>。最后，代码具有**内在的可验证性**。一段代码要么能够编译并通过单元测试，要么不能，这为模型评估和强化学习提供了清晰、自动化的反馈信号，这一特性在主观性更强的

领域（如战略分析）中是缺失的（Stanford CS224g, 2025）<sup>2</sup>。这些特性共同使得在编程领域训练出一个“可用”的模型，其技术路径远比在其他知识领域更为清晰和直接。

#### 4.2. 训练-应用悖论：概率模型与确定性任务的根本性张力

然而，深入分析揭示了一个根本性的“训练-应用悖论”：那些使代码成为理想训练数据集的特性（结构、逻辑、可验证性），恰恰与 LLM 输出的核心性质（概率性、非确定性、常有瑕疵）不匹配。LLM 的生成过程本质上是统计性的，它通过预测序列中下一个最有可能的词元（token）来构建输出，而非进行逻辑推导或符号推理（Grow, 2025）<sup>7</sup>。这导致其输出天然地具有不确定性。

而在现实世界的应用中，软件工程是一个要求近乎完美精确度的领域。一个错位的分号、一个错误的逻辑判断，都可能导致系统性失败或引入难以察觉的安全漏洞（Scrum.org, 2025）<sup>6</sup>。因此，我们正在用一个**概率系统**，在一个**确定性的数据集**上训练，以期应用于一个**确定性的任务**。这种根本性的错位，是“最佳对齐”论点的核心弱点。训练过程的成功（模型学会了语法和常见模式）掩盖了其在应用过程中的脆弱性（模型不理解逻辑，随时可能犯下致命错误）。

#### 4.3. 路径依赖的形成：“可衡量性”驱动的制度锁定

最初专注于代码的战略选择，因其早期可量化的成功而被放大，从而创造了一种强大的路径依赖。企业采纳昂贵的高新技术，迫切需要用清晰的投资回报率（ROI）来证明其合理性（Masood, 2025）<sup>17</sup>。开发者生产力虽然复杂，但存在既定的（尽管有争议的）衡量指标，如任务完成时间、代码接受率、

拉取请求数量等（GitHub, 2022）<sup>10</sup>。这些指标易于量化和报告，为管理者采纳提供了看似确凿的依据。相比之下，管理工作的成效涉及判断力、沟通和激励，这些都难以轻易衡量。因此，选择编程作为起点，不仅是数据问题，更是**成果可衡量性**的问题。在展示价值方面，阻力最小的路径是通过编程，即便技术契合度存在根本矛盾。

这一发展轨迹被两大理论所强化。**路径依赖**理论告诉我们，早期的成功会带来递增报酬（网络效应、学习曲线），并形成高昂的转换成本，使整个生态（人才、资本、产品）被“锁定”在这条道路上（David, 1985）<sup>13</sup>。同时，**制度同构**理论解释了这种现象的快速扩散：当微软/GitHub 这一行业领袖通过 Copilot 取得成功后，其他科技公司在不确定性面前，会倾向于**模仿**（mimetic isomorphism）这一成功策略；同时，一个“现代开发者就应该使用 AI 助手”的**规范**（normative isomorphism）开始形成，并通过教育和专业网络传播开来，最终使得“AI 辅助编程”成为行业标准（DiMaggio & Powell, 1983）<sup>15</sup>。

#### 4.4. 最优对齐的幻觉：被忽视的“专家在环”隐性成本

LLM 在编程领域的普及，并非其与任务完美契合的证据，而是一个根本上不完整的技术需要持续、专业的人工干预才能变得可行的明证。

大量研究表明，LLM 生成的代码通常存在缺陷、不安全且难以维护。一项研究发现，Copilot 的解决方案中只有 28.7% 是正确的（Sahota, 2025）<sup>9</sup>。另一项研究发现，使用 AI 辅助的工程师更有可能编写不安全的代码（Perry et al., 2022）<sup>9</sup>。这表明，人类开发者的角色并非简单的协作，而是必不可少的监督、调试和质量控制。开发者必须充当“专

家在环” (Expert-in-the-Loop, EITL)，运用深厚的领域知识来指导、验证和修正 AI 的输出 (GitHub, 2024; AI@93276, 2025)<sup>18 19</sup>。

为了更生动地理解这一点，我们可以进行一个**迷你案例**的设想：一位初级开发者和一位资深开发者使用 Copilot 完成同一个涉及复杂 API 调用的任务。初级开发者可能会迅速接受 AI 生成的代码片段，因为它们“看起来能用”，但却可能忽略了其中潜在的内存泄漏和安全隐患。而资深开发者则会审慎地检查每一行代码，质疑其逻辑，重构其结构以符合项目规范，并编写额外的单元测试来验证其边缘情况。最终，虽然两人都使用了 AI，但代码的质量和长期可维护性却天差地别。

这揭示了“生产力”的**隐性成本**。当前对生产力的衡量大多关注初始代码生成的速度，却未能计入资深开发者为修复 AI 产生的低质量、不安全代码所投入的审查、调试和重构时间的“净”成本。“专家在环”的劳动是大多数生产力计算中未被计入成本的依赖项。因此，“最佳对齐”更像是一种通过忽视专家修复的隐性成本而得以维持的幻觉。

## 5. 第二部分：未开发的疆域——LLM 在事务管理中的应用

本部分旨在论证，LLM 的内在属性与高容错性的事务管理领域存在更优的技术对齐，但这一潜力仍未被充分挖掘。

**5.1. 定义领域：事务管理与组织的内在容错性** 如前所述，**事务性领导 (Transactional Leadership)** 关

注的是组织日常运转的结构与流程，其任务通常是重复性和信息密集型的，如起草周报、汇总团队进展、分配任务、安排会议和确保流程合规 (Burns, 1978)<sup>21</sup>。

我们将此与工程学中的**容错性 (Fault Tolerance)** 概念相联系，即系统在部分组件发生故障时仍能继续运行的能力 (FasterCapital, 2025)<sup>24</sup>。组织，特别是其管理系统，天然就是一个高度容错的系统。这种容错性通过**组织惰性 (Organizational Slack)**——即组织可用资源与维持运营所需资源之间的差额——得以实现 (Cyert & March, 1963)<sup>26</sup>。组织惰性充当缓冲器，吸收了次优决策和沟通失误带来的成本。一封措辞稍有不当的邮件、一个稍微错过的截止日期，或一次次优的资源分配，很少会导致整个组织系统的崩溃。组织的设计就是为了通过后续的澄清、追问、人为纠正和冗余流程来吸收这些“故障”。

这引出一个核心观点：管理领域与编程不同，其内在设计就是为了处理一定程度的错误、模糊性和次优结果。它是一个“概率性”的领域，这使其与 LLM 的“概率性”本质更为天然地契合。将一个概率性工具 (LLM) 应用于一个脆弱、确定性的领域 (编程)，会产生巨大摩擦；而将其应用于一个有弹性、概率性的领域 (管理)，则创造了更自然的对齐，其中系统固有的容错性可以吸收工具的不完美之处。

**5.2. 更优的技术-任务对齐：LLM 优势与管理任务的映射** LLM 的核心优势与事务管理任务高度对应。其**总结与综合能力**可用于从冗长的邮件链和多个报告中提炼关键信息，为管理者节省大量时间；**信息提取与模式识别**能力可用于扫描绩效数据和客户反馈，以增强监控能力；**内容生成能力**可

用于高效起草邮件、项目更新、会议议程和绩效评估初稿；**任务分解**能力则可用于将一个高层目标分解为可执行的工作流（Tookitaki, 2025; ListedKit, 2025）<sup>22 23</sup>。

对于许多事务性管理任务，LLM 提供的“**足够好**”（good enough）的输出远比在编程领域更有价值。在管理领域，一个能为管理者节省 80% 起草时间的、内容连贯但可能需要微调的报告初稿，就是一个重大的胜利。这个“足够好”的门槛远低于编程领域“必须完美无瑕才能编译运行”的要求。这意味着，由于其高容错性和对“成功”输出的较低标准，管理领域中价值生成的“命中率”在本质上更高。

**5.3. 反论：算法管理的局限性与不可简化的人类因素** 然而，将 LLM 应用于管理并非没有挑战。一个强有力的反论是：管理，即使是事务性的，也具有当前 LLM 无法处理的、不可简化的复杂性，特别是涉及伦理和人际互动的层面。

LLM 存在诸多有据可查的弱点，如**缺乏上下文理解**和**常识推理**、“**黑箱**”问题导致**缺乏可解释性**、以及从训练数据中继承的**内在偏见**，这些在管理情境下尤其有害（Brown, 2025; Sen, 2025）<sup>27 28</sup>。例如，依赖 AI 进行绩效评估，极有可能因为数据偏见而对某些群体的员工做出不公平的判断。

更重要的是，即使是事务管理也需要独特的、不可简化的人类技能。这包括在信息不完全的模糊情况下做出**判断（Judgment）**（Deloitte, 2025）<sup>36</sup>，运用关于组织文化和人际关系的**隐性知识（Tacit Knowledge）**（Nonaka, 1994）<sup>37</sup>，以及建立信任和激励员工的“**连接性劳动（Connective Labor）**”（Pugh, 2022, as cited in UVA Today, 2022）<sup>38</sup>。

让我们通过另一个**迷你案例**来对比两种不同的实施路径。A 公司尝试用一套“**算法管理**”系统来替代中层经理：系统根据 KPI 自动给销售团队分配任务、监控进度并生成绩效排名。结果，团队士气低落，员工感到自己被非人化的数字监控，并开始寻找规避系统监控的方法，创新和协作精神荡然无存。B 公司则为销售经理提供一个 AI 增强工具：该工具帮助经理分析销售数据以发现潜在机会，自动起草每周的进展报告，并提醒经理关注表现异常的团队成员。经理得以从繁琐的文书工作中解放出来，将更多时间用于与团队成员进行一对一的辅导、解决他们遇到的实际困难和进行战略思考。最终，B 公司的业绩和员工满意度都得到了提升。

这个案例生动地展示了“**替代**”与“**增强**”的区别，并揭示了反论的核心：自动化管理任务（如报告）是可行的且有价值的，但这并不能消除对管理者角色的需求。相反，它要求管理者将工作重心转移到 AI 所不具备的、更高级的人类技能上。任何基于离散任务自动化而试图替代管理者角色的尝试，都是一种忽略了管理整体性的范畴错误，并可能导致类似零工经济中“**算法暴政**”的负面后果。

## 6. 第三部分：人类的障碍——社会学与组织性阻力

即使管理 AI 的技术对齐更为优越，其应用也受到强大的人类和系统性因素的阻碍。

**6.1. 机器中的幽灵：中层管理者的抵制与身份威胁** 大量文献证实，中层管理者是抵制新技术，尤其是颠覆性数字技术的关键来源（Cogent Business & Management, 2025）<sup>4 5</sup>。然而，将这种现象简单

归咎于非理性的恐惧或“卢德主义”，是一种表面的、且具有误导性的诊断。更深入的分析表明，这种抵制通常是对角色、权力和身份所受到的多重感知威胁的理性反应。

首先，最直接的是**对角色和工作安全的威胁**。当 Gartner 等权威机构预测，到 2026 年，AI 将可能淘汰超过一半的现有中层管理职位时（Deloitte, 2025）<sup>36</sup>，这在管理者群体中引发了深刻的“生存焦虑”（Schein, 1996）<sup>4 6</sup>。其次，AI 自动化威胁到他们的**权力和控制范围**。在传统层级结构中，中层管理者通过控制信息流、分配资源和监督下属来行使其权力。而管理 AI 系统则可能将这些权力向上集中到高层管理者手中，或横向分散给技术专家，从而削弱中层管理者的权威。最后，也是最根深蒂固的，是**对职业身份的威胁**。许多管理者的职业身份与他们作为经验丰富的决策者、信息中继站和问题解决者的角色紧密相连。AI 有可能使这些长期积累的经验和技能变得过时，从而引发一种深刻的“学习焦虑”——即对自身无法适应新技能要求的恐惧，以及对放弃既有工作习惯的抵制（Schein, 1996）<sup>4 6</sup>。

然而，一个更具批判性的观点是，中层管理者的抵制并非技术的固有属性，而是“技术中心主义”变革管理失败的必然症状。研究表明，AI 项目中高达 70% 的挑战源于人员和流程问题，而非技术本身（Agility at Scale, 2024）<sup>5 0</sup>。当领导层在推行新技术时，仅仅强调其效率和成本节约，却未能为管理者在一个 AI 增强的新世界中阐明一个有价值的、受人尊敬的新角色时，抵制就成了对管理不善、充满威胁的变革的最理性反应。因此，问题的根源往往不在于“管理者的抗拒”，而在于“领导层在管理技术转型的人类维度上的失败”。

**6.2. 规范的力量：社会建构的技术叙事** 除了个体和群体的心理及政治因素外，代码辅助工具的统治地位更是一个被系统性力量所塑造的社会建构结果。**技术社会建构论（SCOT）**为此提供了强大的解释框架。SCOT 理论认为，技术的意义并非内在于技术本身，而是在不同“相关社会群体”的互动中被协商和塑造的（Pinch & Bijker, 1984）<sup>5 2</sup>。

在 LLM 应用的初期，存在着巨大的“**阐释的灵活性**”。这项技术本可以被定义为解决许多问题的方案，例如法律文书分析、科学研究辅助，或是管理流程优化。然而，最终胜出的叙事是由几个强大的“**相关社会群体**”共同推动的：

1. **大型科技公司（如微软、OpenAI）**：他们拥有技术和资本，需要一个清晰、可快速商业化的应用场景来展示其技术的价值。
2. **开发者社区**：这是一个庞大、技术接受度高、且对其工作中的“辛劳”（toil，如编写样板代码）有明确痛点的用户群体。
3. **风险投资生态系统**：他们青睐具有清晰用户、明确痛点和可量化增长指标（如用户采纳率）的项目。

这些群体共同将待解决的“问题”成功地定义为“开发者的辛劳与生产力”。GitHub Copilot 这一产品的推出，成为了一个关键的“人造物”，它有效地结束了关于 LLM 在企业环境中首要用途的争论，实现了“**闭合（Closure）**”。在这个过程中，其他社会群体（如中层管理者、人力资源专家）基本处于沉默或被排斥的状态，他们所面临的问题（如管理效率低下、会议泛滥）未被置于技术议程的中心。

因此，我们观察到的路径依赖，并不仅仅是一个技术或经济的产物；它是一场非常成功的社会学



营销活动的结果。最成功的市场进入策略不仅仅是产品本身，更是围绕它构建的、能够被广泛接受的叙事。将“AI 用于编程”进行社会建构，是一种刻意的市场创造行为，它在公众和企业的想象中定义了该技术的主要目的，从而在其他可能性甚至还未被认真考虑之前就将其排挤出局。

## 7. 讨论：理论与现实意涵

本研究分析不仅解释了 LLM 当前的应用格局，更对其背后更广泛的理论和现实意涵提出了新的思考。

**7.1. 理论意涵：颠覆性技术的“战略性错位”** 我们的分析揭示了颠覆性技术在早期商业化过程中可能存在的一种普遍模式——“**战略性错位**” (Strategic Misalignment)。即，技术的第一个大规模成功应用场景，往往是基于**市场进入的便利性**（如**数据可用性、用户同质性、成果可衡量性**），而非基于**技术与任务本质的最高度对齐**。代码辅助的案例完美地展示了这一点：它是一个绝佳的“滩头阵地” (beachhead)，但可能并非 AI 能创造最大价值的“主战场”。

这一发现对传统的“技术扩散”模型提出了挑战。它表明，市场力量本身可能不会自发地将技术引导至其最具变革潜力的应用领域。相反，早期的成功可能会通过路径依赖和制度同构，创造出一种“局部最优”的陷阱，从而抑制了对更激进、更具颠覆性应用的探索。这提示我们，在分析任何新兴技术的社会影响时，必须超越技术本身的属性，深入考察其商业化路径背后的社会建构过程和权力动态。

**7.2. 管理与实践意涵：从“拥抱 AI”到“批判性部署 AI”** 对于企业管理者和战略制定者而言，本研究最重要的启示是：必须从被动地“拥抱 AI”转向主动地“**批判性部署 AI**”。这意味着，领导者需要警惕并挑战由市场主导者设定的技术叙事。他们不应仅仅追问“我们如何能用 AI 让现有流程变得更快？”，而应提出更根本的问题：“我们组织中哪些高价值、高容错性的领域，可以被这种概率性技术从根本上重塑？”

具体而言，这意味着：

- **重新评估 AI 投资组合**：企业应有意识地将资源从过度饱和的“渐进式改良”应用（如为每个开发者购买 AI 助手），分配一部分到更具探索性的“颠覆性变革”应用（如开发内部管理增强工具）上。
- **赋能而非替代**：在部署 AI 时，尤其是在管理领域，核心原则应是**赋能 (Augmentation)**而非**替代 (Replacement)**。目标应是自动化管理者角色内的低判断力任务，以解放他们从事更高价值的领导工作，如战略思考、团队激励和文化建设。这不仅能降低组织抵制，更能真正发挥人机协同的优势。
- **培养双重素养**：未来的管理者不仅需要具备基本的 AI 工具使用能力，更需要培养一种“**AI 批判性素养**”——即理解 AI 的能力边界、潜在偏见以及何时应该质疑或否决 AI 建议的判断力。

**7.3. 政策与教育意涵：引导健康的 AI 生态** 本研究的发现对政策制定者和教育机构也具有深远意义。如果市场的自发力量可能导致 AI 的应用出现“战略性错位”，那么就需要适当的外部引导。政策

制定者可以考虑设立专项基金或激励措施，鼓励 AI 技术在教育、公共管理、医疗协调等具有高度社会价值但商业化路径不明确的“高容错性”领域进行探索和应用。

对于教育机构，尤其是商学院而言，传统的管理学课程需要进行深刻变革。未来的课程体系不应再将技术视为一个独立的、外生的变量。相反，需要将对 AI 等技术的社会学和伦理学分析，深度整合到领导力、组织行为学和战略管理的核心课程中。培养未来的领导者，不仅要教他们如何读懂财务报表，更要教他们如何读懂算法、理解技术背后的社会动力学，并做出负责任的、以人为本的战略决策。

## 8. 结论与未来研究

**8.1. 结论** 大型语言模型在代码辅助领域的巨大成功，是其技术普及道路上一个高明且必要的开端，它成功地完成了市场的初步启蒙。然而，本文通过整合技术分析、组织社会学理论和技术社会建构论的视角，系统地论证了：将这条成功的初始路径误认为终极方向，将使我们错失 AI 时代最重大的组织变革机遇。

我们的分析揭示，LLM 在编程领域的统治地位，是市场进入策略、路径依赖和社会叙事建构的共同产物，而非技术与任务最优对齐的必然结果。其核心在于一个“训练-应用悖论”：代码的确定性结构使其成为理想的训练数据，但软件工程的低容错性却与 LLM 的概率性输出存在根本矛盾。与之相对，事务性管理工作的高容错性、信息密集特性，与 LLM 的能力本质形成了更天然、更具潜力的技术对齐。

然而，这一潜力却因中层管理者的身份威胁、失败的变革管理以及被锁定的技术叙事而受到系统性阻碍。

未来的挑战，已不再仅仅是提升模型的技术能力，更是要克服组织内部的路径依赖、利益冲突和根深蒂固的认知范式。能够率先认识到这一点，并勇敢地将战略重心从单纯辅助执行者（如开发者）转向全面赋能和重塑协调者（如管理者）的组织，将有可能在下一阶段的 AI 变革中，获得最核心、最持久的竞争优势。这不仅是一场技术革命，更是一场深刻的组织与管理哲学革命。

**8.2. 本研究的局限性** 作为一项基于理论综合与概念分析的研究，本文的主要局限性在于其非经验性。我们的论点和模型是基于对现有文献和公开资料的批判性解读构建的，虽然逻辑上力求严谨，但仍需通过实证研究来检验和修正。我们提出的“战略性错位”和“管理 AI 的潜力”等概念，目前仍属于理论命题，其在真实世界中的具体表现和影响程度，有待进一步的经验证据支持。

**8.3. 未来研究方向** 为了弥补上述局限性并推动该领域的研究，我们提出以下几个具体的、可操作的未来研究方向：

- 1. 民族志研究 (Ethnographic Studies)**：未来的研究可以通过深入的田野调查和民族志方法，长期观察开发团队和管理团队与 LLM 工具的真实互动过程。这将为我们理解“专家在环”的隐性成本、管理者的真实焦虑以及 AI 如何被实际地“挪用”或“抵抗”提供丰富、细致的质性数据。

2. **大规模问卷调查 (Large-Scale Surveys)**: 可以通过设计和分发大规模问卷, 跨行业、跨地区地量化不同职业群体 (特别是开发者与管理者) 对各类 AI 工具的接受度、使用模式、感知效益和职业焦虑感。这有助于检验本文关于路径依赖和职业群体差异的宏观论点。
3. **比较案例研究 (Comparative Case Studies)**: 可以选择数家正在尝试推广“管理 AI”工具的公司, 进行深入的比较案例研究。通过对比成功与失败的案例, 可以识别出影响管理 AI 采纳的关键组织文化、领导力风格和实施策略等因素, 从而将本文提出的战略建议具体化、情境化。
4. **话语分析 (Discourse Analysis)**: 可以系统地分析科技媒体、公司财报和政策文件中关于 AI 应用的“话语”, 追踪“AI 用于编程”和“AI 用于管理”等不同叙事的演变、竞争和相互影响, 从而为本研究的 SCOT 分析提供更扎实的经验证据。

通过这些未来的研究, 我们有望更全面、更深刻地理解并引导这场由大型语言模型驱动的、席卷全球的组织与社会变革。

---

## 参考文献

1. Li, Y., et al. (2025). Large Language Models (LLMs) for Source Code Analysis: applications, models and datasets. arXiv preprint arXiv:2503.17502.
2. Stanford CS224g. (2025). Code Generation with LLMs. Stanford University.
3. Mockus, A. (2025). Cracks in The Stack: Hidden Vulnerabilities and ... Technical Report.
4. Masood, A. (2025). Inside the Great AI Data Grab. Medium.
5. Masood, A. (2025). Open Source Licensing Modalities in Large Language Models. Medium.
6. Scrum.org. (2025). Can AI Generate High-Quality Code?. Scrum.org Blog.
7. Grow, J. (2025). The Strengths and Limitations of Large Language Models in Reasoning, Planning, and Code Integration. Medium.
8. Bivens, J. (2023). Exploring the impact of automation and new technologies on the future of U.S. workers and their families. Washington Center for Equitable Growth.
9. Sahota, N. (2025). AI Code Generation: Faster, Smarter, But Is It Reliable?. neilsahota.com. (Note: Perry et al., 2022, is referenced within this source for the security claim).
10. GitHub. (2022). Research: quantifying GitHub Copilot's impact on developer productivity and happiness. The GitHub Blog.
11. Kalliamvakou, E. (2025). Experience with GitHub Copilot for Developer Productivity at Zoominfo. arXiv preprint arXiv:2501.13282.
12. Brynjolfsson, E., Li, D., & Raymond, L. R. (2023). The Productivity Effects of Generative AI: Evidence from a Field Experiment. MIT Generative AI.
13. Verhoeven, P. (2025). Falling behind the adoption curve: Local journalism's struggle for innovation in the AI transformation. Journalistica.
14. Liebowitz, S. J., & Margolis, S. E. (1995). Path De-

- pendence, Lock-In, and History. *Journal of Law, Economics, & Organization*, 11(1), 205-226.
15. Schildt, H., et al. (2023). New Institutional Theory and AI: Algorithms as a Driving Force of Organizational Isomorphism. ResearchGate. (Note: Original source is DiMaggio, P. J., & Powell, W. W. (1983). The Iron Cage Revisited: Institutional Isomorphism and Collective Rationality in Organizational Fields. *American Sociological Review*, 48(2), 147-160.)
  16. Emerald Insight. (2024). Adoption and integration of AI in organizations: a systematic review. Kybernetes.
  17. Masood, A. (2025). AI in Organizational Change Management. Medium.
  18. GitHub. (2024). Why developer expertise matters more than ever in the age of AI. The GitHub Blog.
  19. AI@93276. (2025). Expert in the Loop: Why AI Needs Specialists, Not Just Humans. Medium.
  20. Mozilla Foundation. (2024). Towards Best Practices for Open Datasets for LLM Training.
  21. The Decision Lab. (2025). Transactional Leadership. (Note: Original source is Burns, J. M. (1978). *Leadership*. Harper & Row.)
  22. Tookitaki. (2025). Automated Transaction Monitoring: A New Era.
  23. ListedKit. (2025). 7 Steps to Build a Transaction Workflow.
  24. FasterCapital. (2025). Fault Tolerance: Uninterrupted Transactions.
  25. Limble CMMS. (2025). What is Fault Tolerance?.
  26. CORE. (2014). Organizational Slack, Structure, and Learning. (Note: Original source is Cyert, R. M., & March, J. G. (1963). *A Behavioral Theory of the Firm*. Prentice-Hall.)
  27. Brown, E. (2025). The Strengths and Limitations of Large Language Models. ericbrown.com.
  28. Sen, P. (2025). LLM Management Risks, Challenges And Opportunities for Businesses. Medium.
  29. Standard Beagle. (2025). The Future of Work Is Now: AI Project Management for Product Teams.
  30. Gradient Ascent. (2025). AI as a Disruptive Technology.
  31. Magnimind Academy. (2025). Unlocking the Mystery of Emergent Capabilities in LLMs.
  32. Deepchecks. (2025). Exploring the Emergent Abilities of Large Language Models.
  33. Dhiria. (2025). LLMs: emerging abilities or statistical artifacts?. (Note: Refers to Schaeffer, R., et al. (2023). Are Emergent Abilities of Large Language Models a Mirage?.)
  34. arXiv. (2025). Emergent Abilities in Large Language Models: A Survey.
  35. Schaeffer, R., et al. (2023). Are Emergent Abilities of Large Language Models a Mirage?. OpenReview.
  36. Deloitte. (2025). Is there still value in the role of managers?. Deloitte Insights.
  37. Annual Reviews. (2024). Automation and Augmentation: Artificial Intelligence, Robots, and the Future of Work. (Note: Original source on Tacit Knowledge is Nonaka, I. (1994). A Dynamic Theory of Organizational Knowledge Creation. *Organization Science*, 5(1), 14-37.)
  38. UVA Today. (2022). Q&A: Is Automation Making

- Your Life Worse?. (Citing research by Prof. Lora Pugh).
39. Galaxy.ai. (2025). Navigating the Future of Project Management in the Age of AI.
  40. Project Management Institute. (2025). How AI Will Transform Project Management.
  41. New Jersey Innovation Institute. (2024). Project Management Tips and Tricks for 2024.
  42. ZDNet. (2019). Whither project managers? AI will take 80 percent project management tasks, says Gartner.
  43. EMSNow. (2025). Gartner Identifies Six Trends for Chief Human Resources Officers.
  44. IMD Business School. (2025). Automation and the future of middle management.
  45. Cogent Business & Management. (2025). A scoping review of middle managers in the digital transformation.
  46. INSEAD. (2020). Disruption from the Mysterious Middle. (Note: Original source is Schein, E. H. (1996). Kurt Lewin's Change Theory in the Field and in the Classroom: Notes Toward a Model of Managed Learning. *Systems Practice*, 9(1), 27-47.)
  47. Easy Sociology. (2025). The Negative Effect of Automation.
  48. Sociologica. (2023). The Ghost of Middle Management: Automation, Control, and Relational Work in the Platform Economy.
  49. Sociologica. (2023). (Duplicate of 48).
  50. Agility at Scale. (2024). The Human Side of AI Transformation.
  51. SlideShare. (2025). Social constructionism of technology.
  52. IGI Global. (2025). What is Social Construction of Technology (SCOT). (Note: Original source is Pinch, T. J., & Bijker, W. E. (1984). The Social Construction of Facts and Artefacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other. *Social Studies of Science*, 14(3), 399-441.)
  53. ResearchGate. (2025). Human and Artificial Intelligence Interaction from the Perspective of Social Construction of Technology.
  54. UWE Repository. (2023). Developing Industry 4.0 applications - A social construction of technology approach.
  55. Journal of Science Communication. (2025). Negotiating AI(s) futures.