When AI Lowers Signal Costs, Do Markets Adjust?

Abstract

In digital product markets, users face uncertainty about project quality and rely on observable cues, such as product differentiation and development effort, to guide adoption and endorsement decisions. Generative AI is fundamentally changing this process by sharply lowering the cost of producing these signals at scale. However, it remains unclear whether the market discounts these now-cheaper signals or if continues to reward them as cognitive shortcuts. Using 1.13 million Hugging Face projects, we find that AI Agent Assistance increases both effort signals and idea differentiation. However, the resulting AI market premium is transmitted primarily through effort signals rather than product idea differentiation. More importantly, the market does not significantly reduce the reward attached to effort signals after AI lowers their production cost. Our findings refine signaling theory for AI-mediated digital markets and show how AI can create market advantage when production costs change faster than evaluation rules.

Keywords: generative AI, signaling theory, digital platforms, product differentiation, mediation analysis, developer productivity, market success

1. Introduction

In digital product markets, users evaluating a software project cannot directly inspect its underlying quality before adoption [1,33]. To navigate this uncertainty, they rely on cues that the producer leaves visible. Two cues have historically shaped which projects gain traction. The first is product differentiation. Projects that look distinct from existing offerings draw attention because novelty serves as a value driver in markets with limited user attention [5,8,28]. The second is effort. Drawing on signaling theory, observable development activity such as frequent commits, detailed documentation, and sustained maintenance functions as a credible quality indicator because such activity has historically been costly to produce [14,45,48]. Before the diffusion of generative AI, both cues required substantial human input to produce, and projects that displayed either or both received market reward [15,34].

Generative AI Agents change the production economics of both cues. AI agent such as Claude Code and Codex lowers the marginal cost of generating code and documentation, allowing developers to produce visible activity at a substantially higher intensity than human-only workflows could sustain [10,16,41]. Activities that once functioned as costly indicators of dedication can now be replicated at scale. This shift challenges the core premise of signaling theory: signals are informative precisely because they are costly. When effort becomes cheap, its credibility as a quality indicator should weaken. At the same time, AI may free developers to explore unfamiliar problem spaces by absorbing routine implementation tasks [32]. Models trained on similar corpora, however, may also pull outputs toward common patterns and reduce cross-developer diversity [20]. Taken together, these changes destabilize the traditional relationship between observable effort, differentiation, and perceived product quality.

Previous research has documented substantial productivity gains from AI-assisted development [10,41] but stops short of tracing how that expanded output is evaluated by the market. If evaluators recognize that effort signals have become easier to produce, they should discount these signals or shift attention toward other cues, such as idea differentiation. If not, market evaluation may remain anchored to historical heuristics, continuing to reward effort even after its informational value has declined. This tension raises an important question: when a technological shock alters the cost of producing quality signals, does the market's evaluation function adapt accordingly, or does it remain inert?

To answer this question, we collected 1.13 million projects from Hugging Face Spaces, a leading platform for digital AI applications, spanning 2024 to 2026. Leveraging the emergence of agent-based coding tools around mid-2025 as a natural inflection point, we focus on developers who create both AI and non-AI projects and exploit within-developer variation to isolate how AI adoption changes production behavior and subsequent market evaluation. We find that AI adoption substantially increases observable development activity, leading to more code commits and longer documentation. It also increases product idea differentiation. However, differentiation does not translate into additional rewards. The AI premium is carried primarily by observable effort signals, and the market continues to reward those signals at a stable rate even after AI lowers their production cost. We label this phenomenon *signal decoupling*— a condition in which the cost of producing a quality signal decline sharply, yet the market continues to reward it as if it were still costly.

Our study makes three main contributions. First, we identify signal decoupling as a boundary condition of classical signaling theory. Second, we distinguish AI's production effects

from its valuation effects. AI assistance increases both effort signals and idea differentiation, but only effort signals transmit the AI market premium. Finally, we offer practical insights into the current transitional phase of digital markets. Early AI adopters may therefore capture outsized returns during a transitional period in which production costs have changed faster than market evaluation heuristics.

2. Theoretical Background

2.1 Information Asymmetry and Product Evaluation on Digital Platforms

Digital platforms host enormous numbers of competing products. As of early 2026, GitHub indexes over 400 million public repositories, Hugging Face hosts more than 1.1 million deployable applications, and similar volumes characterize app stores, content marketplaces, and online developer communities. This volume creates a structural evaluation challenge for adopters. A potential user encountering a candidate product on such a platform cannot directly observe its underlying technical quality, robustness, or fit for purpose, and the cost of fully evaluating each candidate often exceeds the cost of moving on to the next [29,31,46]. Search and evaluation costs are therefore the central frictions through which platform users allocate their attention.

Adopters cope with this asymmetry by relying on observable cues that correlate with quality but do not require deep inspection. The reliance on such cues, often referred to as cognitive shortcuts or evaluative heuristics, is well documented across decision domains characterized by information asymmetry [50]. On digital platforms specifically, observable cues serve a dual purpose. They reduce the cognitive cost of within-product quality assessment, and

they organize attention across the broader catalog so that the most promising candidates rise to visibility.

Information systems research on digital marketplaces, open-source software communities, and content platforms has converged on two such cues as the most consistent predictors of adoption. The first is the visible effort that producers invest in their products, captured by markers such as commit activity, documentation depth, and update cadence [44,45]. The second is the conceptual distinctiveness of a product relative to existing offerings, captured by markers of originality and novelty [2,17]. Both cues have functioned reliably because they rest on a shared theoretical mechanism: producing these cues incurs significant costs, rendering them credible and difficult for lower-quality producers to imitate (or "fake"). Generative AI, however, has substantially lowered the production cost of both cues, raising the open question of whether they continue to convey quality information as they have historically. The next two subsections develop the theoretical foundations and empirical findings supporting each cue, and consider how the conditions that have made each cue informative may now be changing.

2.2 Effort Signals: From Costly Investment to Negligible Cost

Effort can operate as a signal of unobservable quality only when its production is sufficiently costly. In classical signaling theory, costly observable actions support separation because they are harder for low-quality producers to imitate than for high-quality producers to provide [48]. Management research generalizes this logic by emphasizing that credible signals must be observable to receivers, costly to producers, and interpretable across the producer–receiver gap [14]. Marketing research reaches a similar conclusion: a signal becomes informative when imitating it is more expensive than genuinely possessing the quality it represents [33]. The

shared insight across these formulations is that production cost is not a background feature of a signal; it is the condition that gives the signal its credibility.

Information systems research has extended this cost–credibility logic to digital platforms. In open-source software settings, observable development activity, project documentation, and update regularity have been linked to downstream adoption and visibility, suggesting that users treat visible effort as a cue for otherwise difficult-to-observe project quality [45]. Contributors also appear to recognize the reputational value of such cues, using observable effort to build credibility with potential users, collaborators, and evaluators [44]. Together, this work suggests that on digital platforms characterized by high product volume and evaluation uncertainty, visible effort has historically served as a credible quality cue.

Generative AI agents are beginning to change the production conditions of observable effort signals on software platforms. Prior studies report sizable but context-dependent productivity gains from AI coding assistants, ranging from faster completion of controlled programming tasks to higher task completion in enterprise field experiments [16,41]. Natural-experiment evidence further suggests that GitHub Copilot altered voluntary open-source contribution patterns [54]. Beyond task completion, emerging evidence also suggests that AI coding agents reshape the visible artifacts of software work, producing pull requests with distinctive description styles, change structures, and commit patterns [39,52]. Taken together, this evidence suggests that AI is changing the production conditions of visible effort signals. Markers that platforms have historically treated as credible cues, such as commits, pull-request activity, and documentation, can now be produced more quickly and at greater scale, raising the possibility of signal inflation.

Classical signaling logic implies that when the production cost of a signal falls, its separating value should weaken. A cue that is no longer sufficiently costly, especially for lower-quality producers, becomes less able to distinguish high-quality from low-quality types [14,48]. However, whether evaluators actually update their inferential rules in response to this cost shift is a distinct behavioral question. Evaluators may continue to rely on familiar cues because such cues offer simple heuristics under uncertainty, even when the production conditions that once made them credible have changed [24,49]. This creates a theoretical tension: do platform markets discount AI-inflated effort signals, or do familiar effort cues continue to operate as valuable heuristics? Furthermore, if AI weakens the signaling value of effort, it raises a second question: whether evaluators shift attention toward alternative dimensions of value, such as how distinct a product is from its peers.

2.3 Idea Differentiation: From Bottleneck to Abundant Variation

This second possibility directs attention to idea differentiation, a distinct pathway through which AI-assisted products may gain market recognition. Idea differentiation captures a different dimension of market evaluation: whether a product appears distinct from existing alternatives. Creativity research commonly defines creative output as combining novelty, or originality relative to prior work, with usefulness, or potential value to adopters [2]. Differentiation can therefore serve as a visible cue that a product embodies creative search rather than routine imitation. On digital platforms, this cue matters not only because novelty may imply substantive innovation, but also because markets with many competing products are also markets for attention. When users face many alternatives, products that depart from the prevailing distribution of offerings may be more noticeable and more discoverable [21]. Previous research

on digital platforms has also linked product differentiation to adoption outcomes. In open-source software settings, diversity within distributed project collectives has been shown to influence both community engagement and market success, with effects varying across types of diversity and stages of project development [17]. In platform-based software markets, product variety and differentiation also shape innovation incentives: differentiated applications can expand platform variety, whereas clusters of close substitutes intensify crowding and reduce incentives to innovate [8]. Historically, meaningful novelty has required cognitive and search costs: creators need domain knowledge, divergent search, and the ability to recombine existing elements in non-obvious ways [23].

Generative AI alters the production economics of idea differentiation in a more theoretically ambiguous way. On one hand, AI can expand the set of ideas an individual creator can explore by supporting brainstorming, recombination, prototyping, and implementation across unfamiliar domains [32]. This may increase the apparent novelty of individual projects. On the other hand, when many creators draw on the same underlying models, their outputs may converge toward similar patterns： AI-assisted creators tend toward similar high-quality solutions, producing algorithmic homogenization, where individual novelty rises while collective diversity falls [20]. Thus, AI may increase a project's apparent differentiation from prior offerings, but it may also make differentiation less informative if many AI-assisted projects vary in similar ways. Whether idea differentiation continues to translate into market success is therefore an open empirical question.

3. Hypothesis Development

We develop our hypotheses in five steps. We first establish the within-developer baseline relationship between AI Agent Assistance and market success (H1). We then specify the two production-side channels through which AI Agent Assistance may operate on market success: effort signals (H2a) and idea differentiation (H2b). We translate these channels into mediation predictions in H3a and H3b. Finally, we develop two directional predictions about whether the market's per-unit valuation of effort signals adjusts to AI-induced changes in their production cost: across periods (H4a) and across AI status within the post-agent period (H4b), both drawing on the classical signaling prediction of rational discounting. Together, these hypotheses test whether AI Agent Assistance affects market success, why that effect arises, and whether market evaluation rules adjust when the production cost of effort signals declines.

3.1 The AI Market Premium

We begin by establishing the baseline relationship between AI Agent Assistance and market success. Developing a digital product requires developers to move from an initial idea to a usable, market-facing project. This process is constrained by time, technical expertise, debugging capacity, and the ability to communicate the project clearly to potential users. AI agents relax these constraints by helping developers write code, resolve implementation problems, generate supporting materials, and iterate more quickly. Prior research shows that AI coding assistants can increase developer productivity, enabling developers to complete tasks faster and produce more output within the same time constraints [16,41]. In digital product markets, this increased production capacity should make AI-assisted projects more likely to reach a usable and presentable form that users are willing to try, like, and share.

AI assistance may also improve what developers are able to execute. Generative AI can help users complete tasks at higher quality, with especially large gains for less experienced or lower-performing individuals [18,38]. In software development, this implies that AI agents may help developers overcome technical bottlenecks, implement unfamiliar features, fix errors, and communicate project value more clearly. Moreover, AI agents may broaden the set of projects developers are willing to attempt. Prior work on generative AI and ideation suggests that AI can expand the range of ideas people generate and help them explore more varied solutions [20,25]. By lowering the cost of brainstorming, prototyping, and combining ideas across domains, AI assistance can help developers move beyond familiar project templates. On a crowded platform, this can increase the chance that a project offers a use case or functional combination that attracts user attention. Thus, we hypothesize:

Hypothesis 1. Within the post-agent period, AI Agent–Assisted digital products achieve higher market success than comparable non-AI products developed by the same developer.

3.2 Impact of AI on Effort Signals and Idea Differentiation

While Hypothesis 1 specifies the overall relationship between AI Agent Assistance and market success, we next unpack the production-side channels that may generate this relationship. The first channel is effort signaling. In digital product markets, visible development traces such as commits and documentation help evaluators infer whether a project is active, maintained, and worth considering [14]. AI agents change the production conditions behind these traces by assisting with coding, debugging, documentation, and routine development tasks [3]. Evidence from AI coding-assistant research shows that such tools can increase software-development productivity, although the magnitude varies across tasks and settings [16,41]. As a result, the

same developer should be able to produce more visible development artifacts when using AI assistance than when working without it. Thus, we hypothesize:

Hypothesis 2a. Within the post-agent period, AI Agent-Assisted projects exhibit stronger effort signals than non-AI projects developed by the same developer.

Beyond increasing visible production activity, AI agents may also change what developers choose to build. Idea differentiation refers to the extent to which a project's stated purpose, use case, or functional combination differs from prior projects on the platform. Prior work on generative AI and creativity suggests that AI can broaden individual ideation by helping users generate, recombine, and elaborate alternative ideas [9,20,25]. In software development, AI agents may similarly help developers consider alternative use cases, technical designs, interface structures, and application domains.

AI agents may also lower the implementation barrier for unfamiliar ideas. By assisting with boilerplate code, debugging, documentation, and adaptation of examples, coding assistants can make it easier for developers to turn ideas outside their usual domain or skill set into working projects [16,41]. This matters because developers may otherwise avoid ideas that are technically feasible but costly to execute. When AI reduces the cost of moving from idea to prototype, the same developer may be more willing to experiment with unfamiliar configurations or combine ideas across domains. Therefore, we hypothesize:

Hypothesis 2b. Within the post-agent period, AI Agent–Assisted projects exhibit higher idea differentiation than non-AI projects developed by the same developer.

3.3 Mediating Channels: Effort and Differentiation

The case for effort signals as a mediating channel builds on signaling theory and research on evaluator attention. In markets where quality is difficult to observe directly, evaluators rely on visible cues to infer producer investment, project activity, and likely value [14,48]. On digital platforms such as Hugging Face, users also face a large and continuous flow of new projects competing for attention. Under information overload, evaluators often conserve cognitive resources by relying on cues that are easy to observe and interpret before engaging in deeper inspection [29,31]. Effort signals, such as commit activity, release activity, and README depth, fit this role because they help users form an initial judgment about whether a project is active, maintained, and worth trying. Prior open-source software research similarly treats project activity and user/developer interest as central indicators and predictors of project success [12,15]. Therefore, we hypothesize:

Hypothesis 3a. Within the post-agent period, the positive effect of AI Agent Assistance on project market success is mediated by its effort signals.

Idea differentiation offers a second possible mediating channel. Differentiated products may attract attention because they appear distinct from existing alternatives and are less directly substitutable with surrounding offerings. In digital platforms with many competing projects and limited user attention, differentiation can matter because projects that address similar use cases compete directly for the same audience, while projects that present a different use case, combine familiar components in a new way, or address a less crowded problem area may face less direct comparison with existing offerings [8]. This distinctiveness may be especially salient in information-rich online environments, where users must allocate limited attention across many

competing alternatives [53]. Differentiated projects can therefore stand out more easily and may receive greater user attention and endorsement. Therefore, we hypothesize that:

Hypothesis 3b. Within the post-agent period, the positive effect of AI Agent Assistance on project market success is mediated by its idea differentiation.

3.4 The Stability of Effort-Signal Valuation

Hypotheses 1 through 3b focus on the production side of the platform. Hypotheses 4a and 4b shift attention to the evaluation side. They ask whether market evaluators adjust the per-unit valuation of effort signals after AI agents reduce the cost of producing those signals.

There are two plausible perspectives on this question. A bounded-rationality perspective suggests that evaluators on high-volume digital platforms may continue to rely on familiar and easy-to-interpret cues even after the production technology behind those cues changes. Under information overload, users often conserve cognitive effort by relying on stable heuristics to allocate attention [31,46,50]. Moreover, decision makers may update interpretive frames slowly when technological conditions shift [49]. From this view, effort signals such as commit activity and README depth may continue to matter because they remain visible, familiar, and inexpensive to process.

Classical signaling theory, however, provides a sharper directional benchmark. A signal is informative when its production cost is correlated with the producer's underlying ability, effort, or commitment [14,48]. In digital product markets, effort signals have historically carried informational value because they required developer time and attention to produce. AI agents weaken this cost-based logic by lowering the marginal cost of generating code, documentation,

and other visible traces of development activity. As a result, the same level of commits or README depth should convey less information about human effort or underlying project quality than it did before.

Under this rational-discounting logic, evaluators should reduce the weight they place on effort signals once AI agents make those signals cheaper to produce. This prediction has two implications. First, if evaluators recalibrate at the market level after the diffusion of AI agents, the relationship between effort signals and market success should be weaker in the post-agent period than in the pre-agent period. Second, if evaluators selectively discount effort signals when AI assistance is detectable, the relationship between effort signals and market success should be weaker for AI-assisted projects than for non-AI projects within the post-agent period.

Hypothesis 4a. The slope of effort signals on market success is smaller in the post-agent period than in the pre-agent period.

Hypothesis 4b. Within the post-agent period, the slope of effort signals on market success is smaller for AI-assisted projects than for non-AI projects.

[Insert Figure 1 about here]

4. Research Design

4.1 Data

We test our hypotheses using data from Hugging Face Spaces (https://huggingface.co/spaces), a large open-source platform for hosting deployable AI applications. A Space is a web-deployable application, or app, that runs on Hugging Face's shared infrastructure. Each Space typically

combines an executable application, built with frameworks such as Gradio, Streamlit, or Docker, a public Git repository with its development history, and a README file describing the project. Visitors can browse, launch, discuss, and like Spaces directly from their public landing pages. The like count is displayed prominently and serves as a visible indicator of community endorsement. Figure 2 shows a Space example in Hugging Face.

[Insert Figure 2 about here]

Hugging Face Spaces provides a suitable empirical setting for our study because it combines product-level market feedback with detailed production traces. The platform hosts a large and active community of developers and users, generating substantial variation in both application design and market engagement. At the same time, each Space is linked to a public repository, allowing us to observe development activity, documentation, source code, dependency files, and textual descriptions at the project level. This structure enables us to examine how AI assistance shapes both the production of observable project signals and their association with market success.

We collected a full snapshot of all publicly available Spaces on March 20, 2026, in Hugging Face. For each Space, we obtained the complete commit history, including timestamps, authors, and commit messages; the README source; the application source code and dependency files; engagement counters, including likes and discussions; and developer profile metadata, including account type, account creation date, and portfolio size. As of March 20, 2026, the full platform snapshot contains 1,134,208 Spaces created by 600,059 developers. Among these projects, 3,229 Spaces are classified as AI-assisted based on repository-level indicators, distributed across 2,477 developers.

Before estimation, we apply two data-quality filters that are unrelated to our hypotheses. First, we exclude projects that appear to be abandoned or placeholder content, including empty repositories, broken builds, and projects with fewer than two commits and no documentation. Second, we require each Space to have a README of at least 50 characters, ensuring that the sentence embeddings used to measure idea differentiation are based on substantive project descriptions rather than text fragments. These filters do not condition on either AI assistance or market success.

Our main identification strategy relies on within-developer variation in AI assistance. Therefore, we mainly focus on developers who created both AI-assisted and non-AI Spaces. In a developer fixed-effects design, developers whose projects are all AI-assisted or all non-AI do not contribute within-developer variation in the treatment indicator and are absorbed by the developer fixed effect. After applying these filters, the within-developer panel contains 13,299 Spaces from 1,334 dual-status developers across the full observation window. Each Space contributes one observation.

For our main mediation tests of Hypotheses 1 through 3, we further restrict the analytical sample to the post-agent period, defined as Spaces created on or after May 1, 2025, when agent-based coding tools (Claude Code, Codex, and Cursor agents) had become broadly available to developers. This restriction yields 8,121 Spaces from 1,237 dual-status developers (AI share = 20.6%; 1,669 AI-assisted versus 6,452 non-AI Spaces). Restricting the main analysis to the post-agent period ensures that our mediation tests compare AI-assisted and non-AI Spaces created within the same competitive environment, in which AI agents are already a familiar production

technology. The pre-agent period (N = 5,178) is reserved for our test of Hypothesis 4 in Section 4.3.3, which compares the market's evaluation of effort signals across periods.

4.2 Variable

Market Success: We measure market success using the number of likes a Space has accumulated by the snapshot date. Likes are a visible platform-level indicator of community endorsement: users can like a Space from its public landing page after viewing the application, its description, and related project information. We use likes because they capture intentional positive engagement with a project and are prominently displayed as a summary signal of community interest on Hugging Face Spaces.

This measure is analogous to GitHub stars, which prior studies have used as project-level indicators of success and community attention in open-source software settings [7,15]. It is also consistent with broader platform research that treats visible community endorsements as indicators of market response [12,45].

AI Agent indicator： Our treatment variable is a binary indicator equal to one if a Space shows repository-level evidence of being built with the assistance of an AI coding agent, and zero otherwise. We define AI coding agents broadly as AI tools that assist developers in generating, modifying, or co-authoring software code based on natural language instructions.

We identify AI assistance using three complementary sources of repository-level evidence. First, we detect tool-specific configuration files and artifacts, such as CLAUDE.md, AGENTS.md,.cursorrules, and .devin/. Second, we identify commit-level evidence, including co-authorship attributions such as “Co-Authored-By: Claude” and bot author tags such as github-

copilot[bot]. Third, we search for README statements that explicitly attribute the development process to AI coding tools, such as “built with Cursor.” The detection procedure covers fourteen AI coding tools, including IDE-integrated assistants such as Cursor, Antigravity, and Windsurf; command-line agents such as Claude Code, OpenAI Codex CLI, and Aider; autonomous agents such as Devin; and AI application builders such as Bolt.new and Lovable.

A Space is coded as AI-assisted if any of these three sources indicates AI-agent involvement in the development process. We exclude automated CI/CD bots, such as Dependabot, because they reflect routine dependency maintenance rather than human-AI collaborative development. This artifact-based measurement strategy is consistent with recent software-engineering studies that identify AI-agent involvement from observable development traces, such as agent-authored pull requests, tool markers, and commit-level records, rather than relying solely on self-reports [52].

In this paper, the AI Agent Assistance indicator captures projects that leave repository-level traces of AI use, rather than the full population of projects whose development may have been influenced by AI tools. This approach may produce false negatives and make our estimates conservative if such unobserved AI use is present in the comparison group. It may also produce false positives if developers mention AI tools without using them substantively, although our requirement of artifact-level evidence reduces this risk. We assess the robustness of our results to alternative detection sources in Appendix E.

Effort signals: We measure effort signaling through two project-level production traces: commit count and README length. Commit count is the total number of commits in a Space’s repository, capturing development activity. Each commit represents a discrete, time-stamped act

of development; it is a widely used operationalization of development effort in open-source software research [37,47]. README length is the character count of the README.md body text (excluding YAML metadata), capturing developer investment in documentation. Documentation is an important form of production work because it helps structure the project, clarify its value proposition, and support adoption and reuse. Prior research similarly treats documentation as an effort-based artifact that reduces information asymmetry and facilitates adoption in software platforms [22,42]. We interpret commit count and README length as visible signals of effort rather than direct measures of human effort. AI agents can affect both signals through more developer effort and through transformed workflows (for example, more granular commit boundaries or AI-drafted boilerplate). Our claims concern how these visible signals predict market success, not whether each unit of signal corresponds to a fixed unit of human effort.

Idea Differentiation: We measure idea differentiation based on how semantically distinct a Space's README is from the existing population of Spaces on the platform at the time of its creation. We use the README because it provides a concise textual description of the project's purpose, functionality, and intended use [36]. For each Space, we encode the README using sentence-transformers/all-MiniLM-L6-v2, a 384-dimensional sentence embedding model commonly used to represent semantic similarity in text [43,51]. We then L2-normalize each README embedding and compute its cosine similarity to the running centroid of all Spaces created strictly before the focal Space. This historical reference set ensures that differentiation is measured relative to the alternatives visible at the time of creation and avoids look-ahead bias. We define idea differentiation as 1 − cosine similarity, so that higher values indicate greater conceptual distinctiveness relative to the existing project landscape.

Control Variables: We include several project-level controls to account for differences in community activity, technology stack, and content domain. First, we control for the number of discussion threads opened on a Space. Discussion activity captures community engagement around a project and is likely correlated with both development effort and likes. Prior open-source software research has similarly used discussion activity as a control for community engagement [27].

Second, we include SDK dummies to account for differences in development frameworks, such as Gradio, Streamlit, and Docker. We also include task-category dummies, such as natural language processing, computer vision, and audio, to absorb broad content-domain heterogeneity across Spaces.

All regressions include developer fixed effects and creation-month fixed effects. Developer fixed effects absorb time-invariant developer characteristics, including unobserved ability, technical style, audience network, prior reputation, and stable differences in account characteristics. Creation-month fixed effects absorb calendar-time variation in platform growth, user attention, the diffusion of AI coding tools, and exposure time arising from the fixed March 2026 snapshot date. Because older projects mechanically have more time to accumulate likes, creation-month fixed effects also account for project age at the monthly level. This fixed-effects structure allows us to compare AI-assisted and non-AI projects created by the same developer, while accounting for common shocks and platform conditions at the time of project creation.

Table 1 reports descriptive statistics for the main variables in their raw, untransformed form. As shown in Table 1, several count-based variables, including likes, commits, README length, and discussion count, are highly right-skewed. Therefore, we log transformed those

variables. Table 2 reports the pairwise correlations among the variables as they enter the regression models.

[Insert Table 1 about here]

[Insert Table 2 about here]

4.3 Models

4.3.1 Baseline Model

We estimate the relationship between AI assistance and project-level market success using a within-developer fixed-effects specification [6,11]:

$$Success_i = \beta_1 AI_i + \gamma' X_i + \alpha_{d(i)} + \delta_{t(i)} + \varepsilon_i$$

where $Success_i$ denotes market success for project i, $AI_i$is a binary indicator equal to one if project i contains detected AI-agent markers and zero otherwise; $X_i$is the vector of project-level controls described in Section 4.2; $\alpha_{d(i)}$is a developer fixed effect; and $\delta_{t(i)}$is a creation-month fixed effect. The developer fixed effect absorbs time-invariant differences across developers, including unobserved ability, technical style, audience network, and prior reputation. The creation-month fixed effect absorbs cohort effects (when the project was launched) and exposure-time effects (how long the project has had to accumulate likes). Standard errors are clustered at the developer level.

The coefficient $\beta_1$captures the within-developer difference in market success between AI-assisted and non-AI projects, net of project-level controls and creation-month effects. This coefficient corresponds to the total effect, or c-path, in the mediation framework and tests Hypothesis 1.

4.3.2 Mechanism Models

We test the proposed mediation channels in three steps. First, we estimate the relationship between AI assistance and each proposed mediator, corresponding to the a-paths in the mediation framework. These models test whether AI assistance increases observable effort signals and product differentiation, as predicted in Hypotheses 2a and 2b. Second, we progressively introduce the mediators into the market-success equation following Baron and Kenny [4], allowing us to examine how much of the AI market premium is absorbed by each channel. Third, we implement the causal mediation framework [26,30] as a complementary decomposition of indirect and direct effects.

(a) A-paths: AI to Each Mediator

For each candidate mediator $M_i \in$ {log commit count, log README length, idea differentiation}, we estimate a separate within-developer fixed-effects regression:

$$M_i = aAI_i + \gamma' X_i + \alpha_{d(i)} + \delta_{t(i)} + \varepsilon_i$$

The coefficient a captures the within-developer difference in the mediator between AI-assisted and non-AI projects, net of project-level controls and creation-month effects. Each a-path regression uses a single-mediator specification and does not include the other candidate mediators as controls. The coefficient a tests Hypothesis 2a when $M_i$is log commit count or log README length, and Hypothesis 2b when $M_i$is idea differentiation.

(b) Mediation: Baron-Kenny Stepwise (H3a, H3b)

To examine whether the AI market premium operates through the proposed mediators, we progressively introduce the candidate mediators into the following equation:

$$Success_i = \beta_1 AI_i + \beta_2' Effort_i + \beta_3 Diff_i + \gamma' X_i + \alpha_{d(i)} + \delta_{t(i)} + \varepsilon_i$$

where $Effort_i$ is the vector of observable effort signals, including log commit count and log README length, and $Diff_i$denotes idea differentiation. We estimate four nested specifications: AI assistance only, AI assistance plus log commit count, AI assistance plus log commit count and log README length, and the full model adding idea differentiation.

The coefficient $\beta_1$captures the remaining direct association between AI assistance and market success after accounting for the included mediators. We calculate the reduction in $\beta_1$relative to the baseline model as an estimate of the share of the AI market premium absorbed by each mediation channel. This stepwise approach tests whether effort signals and idea differentiation explain the positive association between AI assistance and market success, corresponding to Hypotheses 3a and 3b.

(c) Causal Mediation

Following recent applications of the causal mediation framework [19,26], we further implement the approach developed by Imai et al. [30] to estimate the average causal mediation effect (ACME) and the average direct effect (ADE) for each mediator. This analysis provides a complementary decomposition of the indirect effect through each proposed mediator while holding the other mediator constant. Confidence intervals are constructed using developer-clustered bootstrap resampling. As with all mediation analyses using observational data, we interpret these estimates as mechanism evidence conditional on the model specification rather than as definitive proof of causal mediation.

4.3.3 Signaling Theory Test

To assess whether the market's evaluation of effort signals has changed in response to the increased supply of such signals enabled by agent-based AI, we estimate two pooled interaction models. The first model uses the combined pre-agent and post-agent sample to test hypothesis 4a:

$$Success_i = \beta_1' Effort_i + \beta_2 Post_{t(i)} + \beta_3' (Effort_i \times Post_{t(i)}) + \beta_4 Diff_i + \gamma' X_i + \alpha_{d(i)} + \varepsilon_i$$

where $Post_{t(i)}$ equals one if project i was created on or after May 2025 and zero otherwise. The interaction vector $\beta_3$captures whether the association between effort signals and market success changes between the pre-agent and post-agent periods. Specifically, if the market discounts effort after agent-based AI makes effort signals easier to produce, we would expect $\beta_3$to be negative.

The second model uses the post-agent sample only to test hypothesis 4b:

$$Success_i = \beta_1 AI_i + \beta_2' Effort_i + \beta_3' (Effort_i \times AI_i) + \beta_4 Diff_i + \gamma' X_i + \alpha_{d(i)} + \varepsilon_i$$

The interaction vector $\beta_3$tests whether effort signals are evaluated differently for AI-assisted and non-AI projects within the same post-agent period. Specifically, if users discount effort when it is associated with AI assistance, we expect these interaction terms to be negative.

We assess Hypothesis 4 using a directional inferential framework that matches the rational-discounting prediction derived from signaling theory. First, we conduct one-sided t-tests on each interaction coefficient in the predicted direction, $\beta < 0$. Second, because Hypothesis 4 concerns the joint discounting of multiple effort signals, we apply an intersection-union test (IUT), which requires both effort-signal interactions within a hypothesis to be individually significant in the predicted direction. The joint p-value is therefore the maximum of the two one-sided p-values. Finally, we report a two-sided joint Wald $\chi^2(2)$test as an auxiliary test of

whether the effort-signal slopes differ in any direction. We interpret a null IUT result as failure to detect the rational-discounting prediction, not as affirmative evidence that the slopes are equal.

5. Results

Before turning to regression results, we present model-free evidence on the trends underlying our main hypotheses. Figure 3 reports four monthly time series over the visualization window (June 2024 to February 2026), separately for AI Agent-Assisted and non-AI projects in the post-agent period and combined (non-AI) for the pre-agent period. Panel A plots the share of projects with at least one like; Panel B reports idea differentiation; Panel C plots the monthly mean of log(1 + commits); Panel D plots the monthly mean of log (1 + README length).

[Insert Figure 3 about here]

Three patterns are visible. First, Panels C and D show that AI-assisted projects exhibit substantially higher commit counts and longer README documentation than contemporaneous non-AI projects in the post-agent period, while non-AI projects in the post-agent period are largely indistinguishable from pre-agent projects. Second, Panel B shows that AI-assisted projects are also more semantically differentiated from prior projects on the platform. Third, Panel A shows that AI-assisted projects achieve a higher share of likes than contemporaneous non-AI projects, but the difference is more modest than the production-side gaps in Panels B-D, motivating the within-developer regression analysis that follows.

5.1 Baseline AI Premium

Table 3 reports four nested within-developer fixed-effects regressions of market success on AI Agent Assistance and the candidate mediators. Based on Model 1, the coefficient on AI Agent

Assistance is +0.060 (SE = 0.020, p = 0.002), indicating an upward shift of 0.060 in log(1 + likes) for AI-assisted projects relative to non-AI projects by the same developer. Therefore, Hypothesis 1 is supported.

[Insert Table 3 about here]

5.2 AI and Production Behavior

Table 4 reports three within-developer fixed-effects regressions, one for each mediator as the dependent variable. AI Agent Assistance is associated with a +0.456 (SE = 0.034, $p < 0.001$) increase in log commit count in Model 5 and a +0.701 (SE = 0.049, $p < 0.001$) increase in log README length in Model 6 — approximately 58% more commits and 101% longer README documentation in AI-assisted projects. Thus, hypothesis 2a is supported. Based on result in Model 7, the coefficient of AI-Assisted is +0.140 (SE = 0.013, $p < 0.001$), indicating that AI-assisted projects are systematically more differentiated from prior projects on the platform. Thus, hypothesis 2b is supported.

[Insert Table 4 about here]

5.3 Mediation: Effort versus Differentiation

We decompose the AI premium using two complementary methods. Table 3 reports result of Stepwise Baron-Kenny regression and Table 5 reports results of causal mediation mechanism, formally separating the average direct effect (ADE) and the average causal mediation effect (ACME) using cluster-bootstrap inference.

In Table 3, Model 2 reduces the AI coefficient from 0.060 (Model 1) to 0.031. Model 3 further reduces it to 0.016, at which point the AI coefficient is no longer statistically significant.

The Imai decomposition in Table 5 confirms this pattern: the ACME through commits is +0.028 (95 percent CI [+0.012, +0.042], $p < 0.001$), accounting for 46.0 percent of the total effect; the ACME through README is +0.023 (95 percent CI [+0.001, +0.041], $p < 0.05$), accounting for 38.7 percent. Therefore, hypothesis 3a is supported.

According to Table 3 Model 4, the AI coefficient changes only marginally and the coefficient on idea differentiation itself is small and not statistically significant ($b = -0.049$, $SE = 0.048$, n.s.). The Imai decomposition reaches the same conclusion: the ACME through idea differentiation in Table 5 is small and not statistically distinguishable from zero (−0.007, 95 percent CI [−0.017, +0.004]), and the implied share mediated is −11.3 percent. Therefore, hypothesis 3b is not supported. Although AI increases differentiation, the market does not reward differentiation as a driver of market success.

To assess the robustness of the mediation results to alternative specifications of the causal mediation framework, we re-estimate the analysis under the natural direct/indirect effects (NDE/NIE) framework with a treatment-by-mediator interaction, following Li, Grahl, and Hinz [35]. The results are consistent with the Imai decomposition reported in Table 5. Detailed results are reported in Online Appendix Table A.1.

[Insert Table 5 about here]

5.4 Signaling Theory Test

After establishing effort as the primary channel through which AI's market premium operates, we next examine whether the market's pricing of effort signals changes when agent-based AI expands the supply of such signals.

Table 6 reports the pooled slope-stability tests. Model 8 tests the temporal slope adjustment predicted in H4a. The coefficient for Commits × Post-Agent is negative but not statistically significant in the predicted direction ($\beta = -0.016$, SE = 0.016, one-sided p = 0.162). The coefficient for README × Post-Agent is also negative but does not reach conventional significance ($\beta = -0.026$, SE = 0.020, one-sided p = 0.092). The intersection-union test for H4a yields a joint one-sided p-value of 0.162, failing to reject the null at the 10% level. The auxiliary two-sided Wald test is also not significant, $\chi^2(2) = 2.922$, p = 0.232. Thus, H4a is not supported.

Model 9 tests the AI-specific slope discounting predicted in H4b. The coefficient for Commits × AI is negative and marginal under the one-sided test ($\beta = -0.019$, SE = 0.013, one-sided p = 0.076). The coefficient for README × AI is also negative but not significant ($\beta = -0.012$, SE = 0.015, one-sided p = 0.207). The intersection-union test for H4b yields a joint one-sided p-value of 0.207, which does not provide sufficient evidence to support the hypothesis. The auxiliary two-sided Wald test is also not significant, $\chi^2(2) = 2.218$, p = 0.330. Thus, H4b is not supported.

[Insert Table 6 about here]

Taken together, we do not find evidence that the market discounts effort signals after agent-based AI lowers their production cost. We refer to this absence of detectable market-side AI-specific discounting as *signal decoupling*: a gap between production-side signal expansion and market-side adjustment, in which AI agents substantially expand the supply of effort signals, yet the market does not detectably reduce the reward attached to them.

5.5 Robustness Checks

We conduct a series of robustness checks to assess whether our findings depend on sample construction, AI-detection rules, mediator measurement, period definitions, model specification, outcome distribution, or unobserved selection. Table 7 summarizes the concern addressed by each check, the corresponding robustness strategy, and the main finding. Detailed results are reported in the online appendices.

[Insert Table 7 about here]

Together, these checks support the main empirical pattern. The AI premium is not driven by the dual-status sample restriction or a single AI-detection rule. The effort-mediation result is not explained by reverse causality or mediation specification. The idea-differentiation result is robust to alternative semantic reference sets. And the signal-decoupling result is supported by alternative cutoff dates and non-AI slope comparisons.

6. Conclusion

This paper asks whether digital markets recalibrate when AI agents lower the cost of producing visible effort signals. Classical signaling theory would suggest that they should: if a signal becomes easier to produce, evaluators should discount it [14,48]. Yet digital platforms are also environments of limited attention, where users often rely on familiar and easy-to-read cues when judging projects [50].

Our findings show that AI changes production faster than it changes evaluation. AI Agent Assistance increases both effort signals and idea differentiation, but the market premium operates mainly through effort signals. Idea differentiation's limited role in transmitting market success sharpens the core finding: the market continues to reward visible effort signals even after

AI makes those signals easier to generate. We call this pattern *signal decoupling***.** Signal decoupling occurs when the production cost of a market signal declines, but the market's valuation of that signal remains anchored to older assumptions about what the signal means. The signal still works, but its market success becomes partly detached from the cost-based credibility that originally made it informative.

This matters because AI does not simply increase productivity. It can also create a gap between what producers can now generate cheaply and what markets still treat as costly and meaningful. As AI-assisted production becomes routine, this gap may shape how digital platforms reward work, how developers allocate effort, and how users interpret the signals they see.

6.1 Theoretical Contributions

This paper makes three theoretical contributions. First, we identify signal decoupling as a boundary condition of classical signaling theory. Signaling theory argues that signals are informative because they are costly to produce. When the cost of producing a signal falls, rational evaluators should discount the value of that signal. Our findings show a more nuanced pattern. Detectable AI assistance substantially expands visible effort signals, and these signals continue to predict market success and mediate the AI premium. Yet we do not detect a significant AI-specific reduction in the reward attached to effort signals when they are associated with researcher-detected AI-assisted projects. This suggests that the market valuation of a signal can become partially decoupled from the production economics that originally made it credible. The boundary condition is therefore not that effort signals stop mattering. Rather, it is that

familiar effort cues may continue to receive market rewards even when the cost structure behind those cues has changed.

Second, we extend AI productivity research by distinguishing production effects from valuation effects. Prior work has shown that AI can increase output by helping workers write code, generate text, and complete tasks more efficiently. We show that the market consequences of AI depend not only on what AI helps producers create, but also on how evaluators respond to those outputs. In our setting, AI assistance increases both effort signals and idea differentiation, but only effort signals transmit the AI premium to market success. The AI market premium therefore reflects the joint operation of production expansion and market valuation. This distinction matters because productivity gains and market gains are not the same construct. AI can change what producers generate, but whether those changes become market success depends on the evaluation rules used by platform users.

Third, we contribute to research on digital platform evaluation by showing that markets do not reward all AI-enabled outputs equally. Research on digital platforms and innovation often treats differentiation as a route to market attention because differentiated products stand apart from existing alternatives. Our findings qualify this logic in a high-volume digital platform setting. Although AI assistance increases idea differentiation, idea differentiation does not mediate the AI premium. Instead, the premium operates through effort signals that are visible, familiar, and easy for users to interpret. This suggests that when evaluators face many competing projects and limited attention, legible cues of project activity may carry more market weight than substantive novelty. The finding helps explain why AI-enabled production may generate market advantage through visible effort rather than through differentiated ideas alone.

6.2 Practical Implications

Our findings offer practical implications for both developers and platform designers.

For developers, the results suggest that AI assistance creates market success when it strengthens the cues that users already rely on. In our setting, effort signals continue to predict market success even after AI agents make those signals easier to produce. Developers may therefore benefit from using AI not only to generate more code, but also to make projects easier to evaluate through clearer documentation, visible development activity, and a stronger sense that the project is active and maintained. At the same time, our findings caution against assuming that AI-enabled novelty automatically translates into market success. AI increases idea differentiation, but differentiation alone does not explain the AI premium.

For platform designers, signal decoupling raises a design challenge. If users continue to reward effort signals while AI makes those signals easier to produce, traditional cues such as commits and README length may become noisier indicators of project quality or developer commitment. Platforms may therefore need to supplement these cues with additional indicators, such as AI-use disclosure, code-quality measures, maintenance histories, usage-based reliability metrics, or community-vetted reputation signals.

6.3 Limitations and Future Research

This study has several limitations that open opportunities for future research. First, our empirical setting is Hugging Face Spaces, a platform for deployable AI applications. The signal-decoupling pattern may differ in settings where evaluators have stronger expertise, greater incentives, or more time to update their inference rules. Professional code review, scientific

publishing, enterprise software procurement, and app stores may respond differently to AI-assisted signal production. Future research could examine whether signal decoupling generalizes across platforms with different evaluator populations and evaluation processes.

Second, our observation window captures the early phase of AI agent diffusion. We document stability in effort-signal valuation during the first months after agent-based coding tools became widely available, but we cannot rule out longer-term adjustment. Evaluators may eventually learn to discount AI-amplified effort signals, platforms may change how signals are displayed, or new quality indicators may emerge. Future work could examine whether signal decoupling is a temporary transitional condition or a more persistent feature of AI-saturated digital markets.

Third, our AI Agent Assistance measure relies on observable artifacts, including marker files, commit attributions, and README mentions. This approach allows us to identify AI assistance at scale, but it may miss AI-assisted projects that leave no detectable trace and may include projects that mention AI tools without using them substantively. Relatedly, we do not directly observe users' perceptions of AI assistance. Future research could combine artifact-based detection with developer surveys, platform logs, or tool-level telemetry to measure AI assistance more directly.

Fourth, our effort signals are observable proxies for developer investment. Commit count and README length capture visible traces of activity, but they do not capture all dimensions of effort or quality. Code architecture, testing quality, usability, model performance, and long-term maintenance may also shape market evaluation. Future research could examine whether signal

decoupling varies across signals that differ in visibility, production cost, and technical informativeness.

More broadly, future research could investigate the mechanisms that accelerate or slow evaluator adjustment. Some platforms may rebuild the cost-ability link quickly by introducing better quality indicators, verification systems, or disclosure rules. Others may continue to reward familiar but increasingly noisy cues. Understanding when markets recalibrate, and when they remain anchored to old evaluation rules, will be central to explaining how AI reshapes digital competition.

References

1. Akerlof, G.A. The market for “lemons”: Quality uncertainty and the market mechanism. In *Uncertainty in economics*. Elsevier, 1978, pp. 235–251.
2. Anderson, N.; Potočnik, K.; and Zhou, J. Innovation and creativity in organizations: A state-of-the-science review, prospective commentary, and guiding framework. *Journal of management*, *40*, 5 (2014), 1297–1333.
3. Baird, A. and Maruping, L.M. The Next Generation of Research on IS Use: A Theoretical Framework of Delegation to and from Agentic IS Artifacts. *MIS quarterly*, *45*, 1 (2021).
4. Baron, R.M. and Kenny, D.A. The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of personality and social psychology*, *51*, 6 (1986), 1173.
5. Berger, J. and Heath, C. Where consumers diverge from others: Identity signaling and product domains. *Journal of consumer research*, *34*, 2 (2007), 121–134.
6. Bertrand, M.; Duflo, E.; and Mullainathan, S. How much should we trust differences-in-differences estimates? *The Quarterly journal of economics*, *119*, 1 (2004), 249–275.
7. Borges, H.; Hora, A.; and Valente, M.T. Understanding the factors that impact the popularity of GitHub repositories. In *2016 IEEE international conference on software maintenance and evolution (ICSME)*. IEEE, 2016, pp. 334–344.
8. Boudreau, K.J. Let a thousand flowers bloom? An early look at large numbers of software app developers and patterns of innovation. *Organization science*, *23*, 5 (2012), 1409–1427.
9. Boussioux, L.; Lane, J.N.; Zhang, M.; Jacimovic, V.; and Lakhani, K.R. The crowdless future? Generative AI and creative problem-solving. *Organization Science*, *35*, 5 (2024), 1589–1607.
10. Brynjolfsson, E.; Li, D.; and Raymond, L. Generative AI at work. *The Quarterly Journal of Economics*, *140*, 2 (2025), 889–942.
11. Cameron, A.C. and Trivedi, P.K. *Regression analysis of count data*. Cambridge university press, 2013.
12. Chengalur-Smith, I.; Sidorova, A.; and Daniel, S.L. Sustainability of free/libre open source projects: A longitudinal study. *Journal of the Association for Information Systems*, *11*, 11 (2010), 5.
13. Colin Cameron, A. and Miller, D.L. A practitioner’s guide to cluster-robust inference. *Journal of human resources*, *50*, 2 (2015), 317–372.
14. Connelly, B.L.; Certo, S.T.; Ireland, R.D.; and Reutzel, C.R. Signaling theory: A review and assessment. *Journal of management*, *37*, 1 (2011), 39–67.
15. Crowston, K.; Wei, K.; Howison, J.; and Wiggins, A. Free/Libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR)*, *44*, 2 (2008), 1–35.
16. Cui, K.Z.; Demirer, M.; Jaffe, S.; Musolff, L.; Peng, S.; and Salz, T. The effects of generative AI on high-skilled work: Evidence from three field experiments with software developers. *Management Science*, (2026).
17. Daniel, S.; Agarwal, R.; and Stewart, K.J. The Effects of Diversity in Global, Distributed Collectives: A Study of Open Source Project Success. *Information Systems Research*, *24*, 2 (June 2013), 312–333.
18. Dell’Acqua, F.; McFowland III, E.; Mollick, E.R.; et al. Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity

and quality. *Harvard business school technology & operations mgt. Unit working paper*, 24–013 (2023).
19. Dimitriadis, S. and Koning, R. Social skills improve business performance: Evidence from a randomized control trial with entrepreneurs in Togo. *Management science*, *68*, 12 (2022), 8635–8657.
20. Doshi, A.R. and Hauser, O.P. Generative AI enhances individual creativity but reduces the collective diversity of novel content. *Science advances*, *10*, 28 (2024), eadn5290.
21. Evans, D.S. Attention rivalry among online platforms. *Journal of Competition Law & Economics*, *9*, 2 (2013), 313–357.
22. Fan, Y.; Xia, X.; Lo, D.; Hassan, A.E.; and Li, S. What makes a popular academic AI repository? *Empirical Software Engineering*, *26*, 1 (2021), 2.
23. Fleming, L. Recombinant uncertainty in technological search. *Management science*, *47*, 1 (2001), 117–132.
24. Gigerenzer, G. and Brighton, H. Homo heuristicus: Why biased minds make better inferences. *Topics in cognitive science*, *1*, 1 (2009), 107–143.
25. Girotra, K.; Meincke, L.; Terwiesch, C.; and Ulrich, K.T. Ideas are dimes a dozen: Large language models for idea generation in innovation. *Available at SSRN*, *4526071*, (2023).
26. Gong, J.; Liang, Y.; and Ramasubbu, N. Software-vendor diversification: A source of organizational rigidity in adversity? *Journal of Management Information Systems*, *40*, 2 (2023), 338–365.
27. Grewal, R.; Lilien, G.L.; and Mallapragada, G. Location, location, location: How network embeddedness affects project success in open source systems. *Management science*, *52*, 7 (2006), 1043–1056.
28. Hauser, J.; Tellis, G.J.; and Griffin, A. Research on innovation: A review and agenda for marketing science. *Marketing science*, *25*, 6 (2006), 687–717.
29. Hilbert, M. Toward a synthesis of cognitive biases: how noisy information processing can bias human decision making. *Psychological bulletin*, *138*, 2 (2012), 211.
30. Imai, K.; Keele, L.; and Tingley, D. A general approach to causal mediation analysis. *Psychological methods*, *15*, 4 (2010), 309.
31. Iyengar, S.S. and Lepper, M.R. When choice is demotivating: Can one desire too much of a good thing? *Journal of personality and social psychology*, *79*, 6 (2000), 995.
32. Jia, N.; Luo, X.; Fang, Z.; and Liao, C. When and how artificial intelligence augments employee creativity. *Academy of Management journal*, *67*, 1 (2024), 5–32.
33. Kirmani, A. and Rao, A.R. No pain, no gain: A critical review of the literature on signaling unobservable product quality. *Journal of marketing*, *64*, 2 (2000), 66–79.
34. Lerner, J. and Tirole, J. Some simple economics of open source. *The journal of industrial economics*, *50*, 2 (2002), 197–234.
35. Li, X.; Grahl, J.; and Hinz, O. How do recommender systems lead to consumer purchases? A causal mediation analysis of a field experiment. *Information Systems Research*, *33*, 2 (2022), 620–637.
36. Lindberg, A.; Schecter, A.; Berente, N.; Hennel, P.; and Lyytinen, K. The Entrainment of Task Allocation and Release Cycles in Open Source Software Development. *MIS Quarterly*, *48*, 1 (March 2024), 67–94.
37. Malgonde, O.S.; Saldanha, T.J.; and Mithas, S. Resilience in the open source software community: How pandemic and unemployment shocks influence contributions to others' and one's own projects. *MIS quarterly*, *47*, 1 (2023), 361–390.

38. Noy, S. and Zhang, W. Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, *381*, 6654 (2023), 187–192.
39. Ogenrwot, D. and Businge, J. Patchtrack: A comprehensive analysis of chatgpt's influence on pull request outcomes. *Empirical Software Engineering*, *31*, 5 (2026), 136.
40. Oster, E. Unobservable selection and coefficient stability: Theory and evidence. *Journal of Business & Economic Statistics*, *37*, 2 (2019), 187–204.
41. Peng, S.; Kalliamvakou, E.; Cihon, P.; and Demirer, M. The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. 2023. http://arxiv.org/abs/2302.06590.
42. Prana, G.A.A.; Treude, C.; Thung, F.; Atapattu, T.; and Lo, D. Categorizing the content of github readme files. *Empirical Software Engineering*, *24*, 3 (2019), 1296–1327.
43. Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 2019, pp. 3982–3992.
44. Roberts, J.A.; Hann, I.-H.; and Slaughter, S.A. Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management science*, *52*, 7 (2006), 984–999.
45. Setia, P.; Bayus, B.L.; and Rajagopalan, B. The takeoff of open source software: A signaling perspective based on community activities. *MIS quarterly*, *44*, 3 (2020), 1439–1458.
46. Simon, H.A. Bandwagon and underdog effects and the possibility of election predictions. *Public Opinion Quarterly*, *18*, 3 (1954), 245–253.
47. Singh, P.V.; Tan, Y.; and Mookerjee, V. Network effects: The influence of structural capital on open source project success. *MIS Quarterly*, *35*, 4 (2011), 813–829.
48. Spence, M. Job market signaling. In *Uncertainty in economics*. Elsevier, 1978, pp. 281–306.
49. Tripsas, M. and Gavetti, G. Capabilities, cognition, and inertia: Evidence from digital imaging. *The SMS Blackwell Handbook of Organizational Capabilities*, (2017), 393–412.
50. Tversky, A. and Kahneman, D. Judgment under Uncertainty: Heuristics and Biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *science*, *185*, 4157 (1974), 1124–1131.
51. Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; and Zhou, M. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, *33*, (2020), 5776–5788.
52. Watanabe, M.; Li, H.; Kashiwa, Y.; Reid, B.; Iida, H.; and Hassan, A.E. On the use of agentic coding: An empirical study of pull requests on github. *ACM Transactions on Software Engineering and Methodology*, (2025).
53. Wu, F. and Huberman, B.A. Novelty and collective attention. *Proceedings of the National Academy of Sciences*, *104*, 45 (2007), 17599–17601.
54. Yeverechyahu, D.; Mayya, R.; and Oestreicher-Singer, G. The impact of large language models on open-source innovation: Evidence from github copilot. *arXiv preprint arXiv:2409.08379*, (2024).

Table 1. Descriptive Statistics

Variable	Measurements	N	Mean	Median	SD	Min	Max
Market Success	Number of likes	13,299	5.929	0.000	178.2	0.000	16,441
AI Agent Assistance	Indicator (0/1)	13,299	0.146	0.000	0.353	0.000	1.000
Commit Count	Number of commits	13,299	28.3	10.0	66.9	1.000	1,000
README Length	Character count of README	13,299	1,796	288.0	6,027	60.0	450,833
Idea Differentiation	1 − cosine similarity to centroid	13,299	0.342	0.034	0.388	0.002	1.124
Discussion Count	Number of discussions	13,299	1.043	0.000	74.1	0.000	8,426

Table 2. Correlation Matrix

Variable	(1)	(2)	(3)	(4)	(5)	(6)
Market Success	1.000					
AI Agent Assistance	-0.028	1.000				
Commit Count	+0.175	+0.152	1.000			
README Length	+0.095	+0.335	+0.214	1.000		
Idea Differentiation	+0.041	+0.284	+0.146	+0.680	1.000	
Discussion Count	+0.595	-0.017	+0.155	+0.048	+0.023	1.000

Notes. Pearson correlations on the within-developer panel (N = 13,299). Variables enter the correlation in the form used in regressions: log(1 + x) transformations applied to likes, commits, README length, and discussion count

Table 3. Mediation: AI Premium and Mediators

	Model 1 (AI only)	Model 2 (+ Commits)	Model 3 (+ Commits + README)	Model 4 (full + Diff.)
AI Agent Assistance (T)	+0.060**	+0.031†	+0.016	+0.016
	(0.020)	(0.018)	(0.022)	(0.022)
Commit Count (log)		+0.064**	+0.060***	+0.061***
		(0.020)	(0.018)	(0.018)
README Length (log)			+0.024*	+0.033*
			(0.012)	(0.016)
Idea Differentiation (1 − Idea Similarity)				-0.049
				(0.048)
Discussion (log)	+0.663***	+0.642***	+0.641***	+0.641***
	(0.107)	(0.112)	(0.113)	(0.113)
Developer FE	Yes	Yes	Yes	Yes
SDK Dummies	Yes	Yes	Yes	Yes
Task Category Dummies	Yes	Yes	Yes	Yes
Creation-Month FE	Yes	Yes	Yes	Yes
N	8,121	8,121	8,121	8,121
Within-R^2	0.190	0.202	0.204	0.204

*Notes. Dependent variable = log(1 + likes). Post-agent panel, N = 8,121 projects from 1,237 developers. Model 1 reports the total AI Premium (c-path). Models 2 through 4 progressively add candidate mediators. Within-developer FE with cluster-robust SE at developer level [13]. *** p < 0.001, ** p < 0.01, * p < 0.05, † p < 0.10.*

Table 4. A-Paths: Effect of AI Agent Assistance on Each Mediator

	Model 5 DV: Commits (log)	Model 6 DV: README (log)	Model 7 DV: Idea Differentiation
AI Agent Assistance (T)	+0.456***	+0.701***	+0.140***
	(0.034)	(0.049)	(0.013)
Discussion (log)	+0.326***	+0.123*	+0.023*
	(0.031)	(0.051)	(0.011)
Developer FE	Yes	Yes	Yes
SDK Dummies	Yes	Yes	Yes
Task Category Dummies	Yes	Yes	Yes
Creation-Month FE	Yes	Yes	Yes
N	8,121	8,121	8,121
Within-R^2	0.168	0.227	0.235

*Notes. Each column is a separate within-developer FE regression where the listed mediator is the dependent variable. Within-developer FE with cluster-robust SE at developer level. *** p < 0.001, ** p < 0.01, * p < 0.05, † p < 0.10.*

Table 5. Causal Mediation Analysis: ACME Decomposition

Mediator	Indirect effect (ACME) [95% CI]	Direct effect (ADE) [95% CI]	% of Total Effect Mediated
Commit count (log)	+0.028*** [+0.012, +0.042]	+0.016 [-0.021, +0.059]	+46.0%
README length (log)	+0.023* [+0.001, +0.041]	+0.016 [-0.021, +0.059]	+38.7%
Idea differentiation	-0.007 [-0.017, +0.004]	+0.016 [-0.021, +0.059]	-11.3%
Outcome (DV)	log(1 + likes)	log(1 + likes)	—
Treatment (T)	AI Agent Assistance	AI Agent Assistance	—
Sample size (N)	8,121	8,121	8,121

*Notes. Quasi-Bayesian causal mediation following Imai, Keele, and Tingley (2010), implemented on within-developer demeaned data with creation-month FE; 95% CIs and p-values from B = 200 cluster bootstrap by developer. Outcome (DV) = log(1 + likes); Treatment (T) = AI-Assisted indicator. Each row reports a single-mediator decomposition with the other two mediators included as controls. % of Total Effect Mediated = ACME / Total Effect × 100. 95% confidence intervals in brackets are percentile bootstrap intervals; p-values are obtained from CI inversion. *** p < 0.001, ** p < 0.01, * p < 0.05.*

Table 6. Pooled Slope-Stability Tests

	Model 8 Effort × Post (Pooled)	Model 9 Effort × AI (Post-agent)
AI Agent Assistance	—	+0.155
		(0.108)
Post-Agent (May 2025+)	+0.117	—
	(0.123)	
Commit Count (log)	+0.067***	+0.065***
	(0.014)	(0.017)
README Length (log)	+0.057**	+0.037*
	(0.019)	(0.017)
Commits × Post-Agent	-0.016	—
	(0.016)	
README × Post-Agent	-0.026	—
	(0.020)	
Commits × AI	—	-0.019
		(0.013)
README × AI	—	-0.012
		(0.015)
Idea Differentiation	-0.076†	-0.053
	(0.040)	(0.046)
Discussion (log)	+0.914***	+0.639***
	(0.111)	(0.113)
Developer FE	Yes	Yes
SDK Dummies	Yes	Yes
Task Category Dummies	Yes	Yes
Creation-Month FE	No	Yes
Joint Wald $\chi^2(2)$ on	2.922	2.218
Effort × {Post or AI}	(p = 0.232)	(p = 0.330)
N	13,299	8,121

*Notes. Both columns are within-developer FE OLS with cluster-robust SE. The Wald test row reports the joint test of the two interaction terms. *** p < 0.001, ** p < 0.01, * p < 0.05, † p < 0.10.*

Table 7. Summary of Robustness Checks

Concern	Robustness Check	Finding	Appendix
Mediation specification	NDE/NIE mediation model with treatment-by-mediator interaction	Effort remains the dominant channel, while idea differentiation remains unsupported as a mediator.	A
Mediation sensitivity	Imai sensitivity analysis for mediator-outcome confounding	Mediation evidence is consistent with an effort channel	B
Sample restriction and cross-developer selection	Full-platform reproduction using within-developer FE and cross-developer specifications with user controls	The AI premium and effort channel reproduce on the full platform	C
Reverse causality	Early-window effort signals measured 7, 14, and 30 days after project creation	Effort mediation remains when effort is measured early, reducing concern that later market success drives the effort channel.	D
AI-detection validity	Separate AI indicators based on marker files, commit attributions, and README mentions	Main findings hold across single-source AI-detection variants.	E
Idea-differentiation measurement	Alternative semantic reference sets	Main findings hold across different measurements	F
Post-agent cutoff choice	Alternative cutoff dates around the May 2025 baseline	The within-developer H4a null is not driven by a single cutoff date.	G
Pre/post composition shift	Non-AI-only pre/post slope comparison	Effort-pricing slopes among non-AI projects remain stable across periods.	H
Post-treatment engagement control concern	Re-estimate main models without log(1 + discussion)	The AI premium and effort channel remain robust after removing a potentially post-treatment engagement control.	I
Outcome-distribution concern	Two-part hurdle model for zero-heavy likes	AI-assisted projects are more likely to receive any user attention, but do not receive disproportionately more likes once noticed.	J
Selection on unobservables	Oster coefficient-stability bounds [40]	The AI premium is robust to bounded unobserved selection.	K

Figure 1: Conceptual Framework

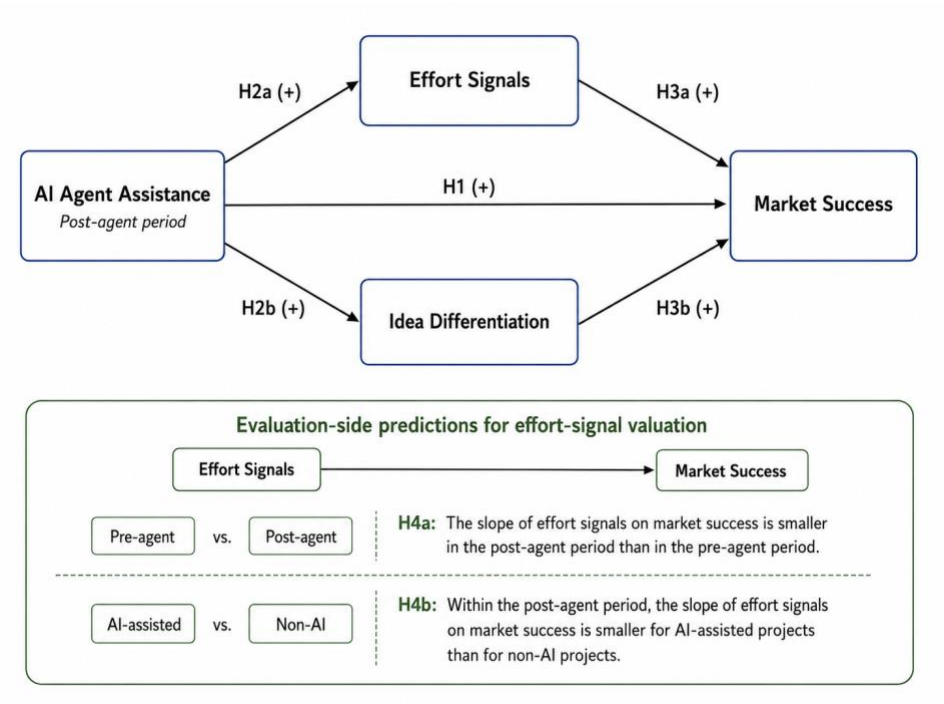


Figure 2. Example of a Space in Hugging Face

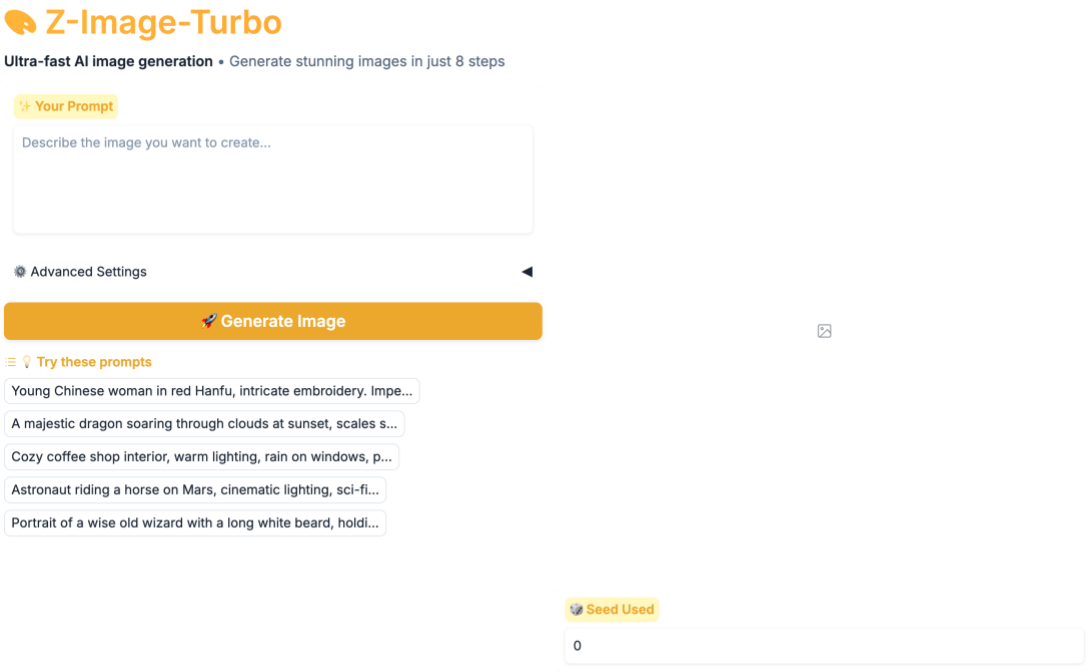


Figure 3. Model-Free Evidence: Monthly Means by AI Status

