Generative AI and the division of labor in Knowledge Work: Evidence from Open Source Ecosystems

[Preliminary: Do not share without authors' consent]

Vincent Lefrere* Léna Poinsignon* Antonio Russo* Maximilian Schäfer*

March 16, 2026

Abstract

Open source development is a key input to collective innovation and production of digital commons. Generative AI coding assistants are reshaping how developers write and maintain code, while major digital firms have expanded their open source footprint, altering the balance between corporate and community participation. This paper studies how the release of ChatGPT-3.5 in November 2022 modified the patterns of open-source activity and whether these changes differed between institutional (firm-linked) and independent developers. Using data on 1M GitHub developers and their activity on the platform, we classify contributors as institutional or independent and estimate a difference-in-differences model with two-way fixed effects. We find that public open source activity declines after the AI shock, as measured by the number of public repositories per developer, while surviving projects tend to be of higher quality: they receive more stars, are more likely to be outgoing forks, and show higher contribution volumes. Institutional developers reduce their public footprint more intensely but experience larger gains in visibility and quality.

*Institut Mines-Télécom Business School

1 Introduction

Open source software (OSS) has become a core element of contemporary digital production. It provides essential infrastructure for firms and users, constitutes a significant part of the current software stack, and offers a lens to understand the dynamics of collective innovation (Lerner and Tirole, 2002; Lakhani and Von Hippel, 2003). The Open Source Initiative (OSI) defines open source through a set of criteria that guarantee free redistribution, access to source code, the right to create derivative works under similar conditions, and non-discrimination across users and fields of use. In practice, these features lower coordination and verification costs, create a stock of reusable code, and facilitate the diffusion of knowledge across projects. Once produced, code is non-rival and, under an open license, non-excludable. Its duplication is almost costless and improvements can spread quickly. Governance is typically delegated to communities and maintainers, often mediated by platforms such as GitHub. Through these properties, large OSS projects such as Linux, LibreOffice, or Mozilla Firefox have emerged as credible alternatives to proprietary software and as central digital commons.

The structure of the OSS ecosystem has, however, changed profoundly over the past decade. Early narratives emphasized volunteer coding driven by intrinsic motives such as learning, reputation, ideology, or enjoyment. More recent evidence points to increasing professionalization and firm involvement. Major digital players now sponsor key maintainers, employ dedicated open source teams, and design "open core" business models that combine public repositories with proprietary extensions and value-added services (Li *et al.*, 2025). A growing share of contributions to flagship projects is made by salaried developers. For example, estimates for the Linux kernel suggest that volunteers now account for only a small fraction of commits. Firms also use open source strategically, to shape ecosystems, commoditize complements, and sell services such as cloud hosting, support, or training. This corporate turn brings investment capacity, professional workflows, and long-term maintenance, but it also raises concerns about governance and value capture. Increased firm control can concentrate decision rights, redefine project priorities, and alter community norms (Bitzer *et al.*, 2007; Meissonier *et al.*, 2010; Zhang *et al.*, 2024).

In parallel, the rise of large language models (LLMs) and generative AI (genAI) coding assistants has introduced a new shock to the software development process. Tools such as GitHub Copilot and conversational models like ChatGPT can generate code and documentation from natural language descriptions, suggest fixes for error messages, refactor existing code, and help write tests or translate between programming languages. They lower the cost of many routine tasks and change how developers search for information. Instead of asking peers in issues, forums, or chat, developers can increasingly ask the AI. Large firms report that a substantial share of their internal code is now written with the assistance of such tools, suggesting that genAI is becoming embedded in standard professional workflows rather than remaining a marginal tool. The emerging empirical literature shows that AI-assisted coding accelerates development and can improve code quality, especially for certain types of tasks (Peng *et al.*, 2023; Hoffmann *et al.*, 2025; Yeverechyahu *et al.*, 2024; Cui *et al.*, 2025). It also appears from previous work that genAI does not affect all contributors in the same way. Some studies emphasize larger gains for more experienced users who can better integrate AI into existing routines, while others highlight its role in helping less experienced participants compensate for missing skills. Developers using AI tools tend to shift effort toward refinement and integration tasks such as debugging, testing, or writing documentation rather than pure exploration, suggesting that genAI may better complement already formalized practices. Other articles study the impact of genAI in online communities and collaborative platforms, documenting changes in participation and content quality (Burtch *et al.*, 2024; Quinn and Gutt, 2025).

From an economic perspective, the corporate turn in OSS and the diffusion of genAI tools raise important questions about how these new technological resources interact differently with intrinsic motivations and extrinsic rewards, who captures the productivity gains enabled by AI, and what is the impact on the volume and nature of contributions, the orientation of projects, and the quality of the resulting code. To explore these issues, we distinguish between two types of contributors: institutional developers and independent developers. Institutional developers are those who are paid by an organization (a company, foundation, university, or public agency) to work on open source softwares at least part of the time. They participate as part of their professional activity and often have access to better infrastructure, complementary tools paid by their organization, and structured workflows. This could make them early and intensive adopters of genAI and allow them to scale up productivity gains on large codebases. At the same time, if AI-generated code dilutes the signaling content of commits or if firms can internalize more production on private repositories, the relative value of maintaining a broad public footprint may decline for this type of developers. By contrast, independent developers contribute mainly in their free time and are not directly remunerated for their contributions. Their incentives are intrinsic (learning, enjoyment, ideology, sense of community belonging...) and their time and resources are more constrained. They may rely on free versions of code tools and have less formal practices, which could limit their effective use of AI or reorient it towards simple production tasks rather than complex integration work. For independent developers, genAI may simultaneously lower entry barriers and raise concerns about appropriation. Indeed, if they perceive that their unpaid contributions are used to train proprietary models or support commercial services, without commensurate returns, their willingness to invest in high-effort public goods may decline.

This paper uses the public release of ChatGPT-3.5 in November 2022 as genAI shock to study how it reshaped public OSS activity on GitHub, with a particular focus on the differential adjustment of institutional and independent developers. We chose the release of

ChatGPT 3.5, which constitutes the starting point for mainstream AI use with 1 million users in only 5 days. While other programming assistants such as GitHub Copilot were already available, ChatGPT-3.5 stands out for its simple interface using natural language that is accessible to everyone. While Github Copilot was mainly likely to impact the supply of code, ChatGPT-3.5 is also capable of impacting the demand for code, as some users who were used to consuming open source projects could now replace some of them with AI. Developers producing substitutable content could then withdraw from the market or redirect their activities toward fewer, more visible, and higher-quality projects. Alongside the empirical analysis, we develop a theoretical model whose purpose is to bring to the foreground the key mechanisms suggested by the literature and our data, and to provide a formal interpretation of the empirical patterns we document.

2 Literature Review

2.1 Impact of Corporate Involvement and Sponsorship Programs on Incentives and OSS Contributions

Early work on OSS emphasized that a substantial share of production came from community efforts carried out by volunteers, typically highly educated developers in IT and computer science (Ghosh *et al.*, 2002). Incentives to contribute are usually divided into intrinsic and extrinsic motivations (Bitzer *et al.*, 2007; Ryan and Deci, 2000). Intrinsic motives include the enjoyment of programming, curiosity and learning, skill improvement, and the desire to engage in a form of innovative cooperation or to belong to a community (Lakhani and Von Hippel, 2003; Carillo *et al.*, 2014). Extrinsic motives are linked to external rewards: peer recognition, career opportunities, or monetary rewards. In particular, public contributions can act as a signal of ability on the job market, where commits and visible projects serve as a portfolio of skills (Lerner and Tirole, 2002; Orman, 2008). Despite this rich set of motivations, participation is fragile. Many projects are abandoned, and surveys reveal that time is the main constraint: most volunteers devote fewer than ten hours per week and receive no monetary compensation for their contributions (Ghosh *et al.*, 2002; Meissonier *et al.*, 2010). The cost of contributing limits sustained engagement and raises concerns about the long-run viability of a purely volunteer-based production model for critical digital infrastructure.

As the strategic and economic value of OSS has become clearer, firms have progressively moved from skepticism to active participation. Companies now pay developers to contribute to existing projects, initiate new open source components, and build business models around them. The literature documents several models: "open core", where a basic version is open but advanced features are proprietary; service-based models centered on support, customization, or managed hosting; and consortium arrangements where multiple firms jointly fund a shared code base. (Jullien *et al.*, 2025) show that sustainable business models revolve around assurance, adaptation and assistance services that require active upstream participation. This professionalization deeply reshapes community dynamics. Case studies, such as the development of the Rust programming language, show that paid contributors have higher commit frequencies and are more likely to add new features than volunteers, and that remuneration increases the probability of becoming a long-term contributor (Zhang *et al.*, 2024). Volunteers often attribute differences mainly to the greater time and budgets of paid developers, and often hold prejudices or misunderstandings about the work and motivations of their paid colleagues, which could hinder collaboration as evidenced by the demission of volunteer teams following the increased involvement of large companies. (Haese and Peukert, 2025) analyse Microsoft's migration of Edge to the Chromium engine and find higher development activity concentrated in the sponsoring firm, faster releases and market gains, but also more concentrated control. Monetary incentives thus have an ambiguous role: they can stabilize and accelerate development by lowering participation costs, but they also reshape motivations and may crowd out community norms.

2.2 Generative AI, Developer Productivity, and Online Communities

In recent years, a growing number of studies have focused on how genAI and code assistants have affected developers' productivity and the organization of work. (Peng *et al.*, 2023) find in a real-world coding trial that AI assistance substantially accelerates routine programming tasks, with particularly large gains for less-experienced developers. (Cui *et al.*, 2025) show in an enterprise setting that access to GitHub Copilot increases the number of tasks completed by high-skilled developers by around a quarter, again with larger effects for those with lower prior productivity. Thus, AI appears to increase overall productivity with heterogeneous effects depending on developer profiles. However, while AI is effectively integrated into workflows, the tasks it helps to perform seem to vary depending on the context. (Gambacorta *et al.*, 2024) document that genAI adoption is associated with higher labour productivity in firms, mainly by automating routine coding and freeing time for higher-skill work. (Hoffmann *et al.*, 2025) show that AI assistance reallocates time away from project management and coordination toward core coding and more autonomous exploration, especially for less central or less experienced team members. (Yeverechyahu *et al.*, 2024) exploit the language-specific rollout of Copilot to measure its impact on open source contributions. They find a substantial increase in the volume of contributions in projects and languages where Copilot becomes available, but the gains are markedly stronger for iterative innovation (maintenance and refinement of existing capabilities) than for capacity innovation that builds new functionalities. Taken together, these works highlight that generative AI both increases productivity and reshapes the composition of work, with heterogeneous effects depending on experience, type of task, and organizational context. Indeed, several other studies emphasized that realized gains depend on workflow design. (Mohamed *et al.*, 2025) study the use of LLM assistants in software engineering teams and show that benefits in productivity come with risks of over-reliance, disrupted flow, uneven quality, and thinner collaboration if teams do not adapt their practices. (Xiao *et al.*, 2024) show that using genAI to draft pull request descriptions leads to faster reviews and higher merge likelihood when maintainers curate outputs. (Cai *et al.*, 2025) analyse issues raised against LLM-generated code in OSS projects and find that integration and alignment problems dominate, and that operational maturity is a key bottleneck to safe adoption.

Another set of contributions examines how AI affects participation and content in online knowledge communities. (Bui *et al.*, 2024) compare the difference in reaction to AI between Western communities using platforms such as Stack Overflow or GitHub, and Chinese communities that are more active on platforms such as CSDN. They show that in Western communities, opposition to AI tools is primarily moral and political: active contributors perceive their unpaid content as being exploited to train proprietary models without consent or compensation, which undermines trust and motivation to contribute. In Chinese communities, criticism is more focused on performance issues (bugs, hallucinations, unstable behavior…) that reduce perceived usefulness. This paper highlights a double threat since AI may affect both the reciprocity inherent in community norms and the quality of the digital commons produced. (Burtch *et al.*, 2024) report declines in daily traffic and question volume on Stack Overflow after the release of ChatGPT-3.5, associated with a deterioration in answer quality. The effects are heterogeneous across topics and more pronounced in areas where ChatGPT is most proficient in. Quinn and Gutt (2025), using StackExchange data, offer a more nuanced view and show that the drop in questions comes mainly from occasional users, whereas highly engaged members reduce activity much less. Moreover, the residual questions become more complex and novel, suggesting that while genAI reshapes human collaboration in online communities, it may also increase the quality of the remaining interactions and content. A related set of papers examines how AI reshapes help-seeking and social learning. (Hou *et al.*, 2025) show that developer teams increasingly route questions to ChatGPT rather than colleagues, reducing mentoring and diversity in problem-solving approaches. (Shan and

Qiu, 2025) analyse chatbot use in Q&A communities and find that AI-generated answers increase answer volume and concision but do not clearly improve community-evaluated quality.

Overall, the literature shows that the production of open source software is shaped by a combination of intrinsic and extrinsic motivations, increasing professional involvement through firms and sponsorship programs, and the diffusion of generative AI tools that affect productivity, task allocation and engagement in online knowledge communities. Building on these insights, our study uses the release of ChatGPT-3.5 and a large GitHub panel to examine how organisation-linked and volunteer developers adjust both the scale of their public activity and the characteristics of their projects in terms of visibility, reuse and collaboration.

3 Towards a Model of Developers Incentives to Open Source

Here we articulate the underlying economic forces that can account for the observed reallocation of effort and visibility within the open-source commons. Interaction between users and developers requires time and attention. As a result, users engage only with repositories whose publicly shared code meets a sufficiently high quality standard. This gives rise to a participation threshold: developers whose idiosyncratic quality falls below a certain cutoff have little incentive to share code publicly, as their projects are unlikely to attract meaningful engagement.

This threshold is plausibly higher for paid developers than for unpaid developers. Paid contributors typically face greater opportunity costs of allocating time to publicly visible projects, since their effort can be deployed in alternative, compensated activities. Even if the underlying distribution of ability were identical across paid and unpaid contributors, these differential opportunity costs would generate selection: among those who remain publicly active, paid developers would, on average, exhibit higher quality than unpaid developers.

The emergence of generative AI provides users with an alternative source of code, effectively raising the minimum quality required to attract interaction with open-source repositories. As this interaction threshold increases, developers whose projects are less likely to clear the higher bar face diminished returns to public participation and are more likely to withdraw. By contrast, higher-quality developers continue to attract attention and remain active. Through this selection process, the average quality of the projects that remain publicly visible may increase following the diffusion of AI tools.

These effects may be amplified for paid developers if firms adjust internal incentives to discourage public code sharing when such sharing enhances competing AI systems. A stronger increase in the effective participation threshold for paid contributors, however, does not mechanically imply greater exit among them; the magnitude of exit depends as well on the distribution of quality within each group.

4 Data

4.1 Data Collection and Sample Characteristics

Table 1: Summary Statistics of Selected Variables: Institutional vs Independent Developers

	Independent			Institutional		
Variable	Mean	SD	Nobs	Mean	SD	Nobs
No. Active Public Repos	2.02	8.50	866,976	2.87	6.11	421,224
Contributions (owner)	4.91	47.43	866,976	6.58	49.50	421,224
Contributions (owner + ext.)	309.90	9,945.30	866,976	498.39	6,074.38	421,224
No. ext. contributors	29.96	343.47	866,976	71.29	338.50	421,224
No. outgoing forks	0.41	7.17	866,976	0.79	4.21	421,224
No. incoming forks	9.02	186.85	866,976	15.49	148.84	421,224
Is active in one public repo	0.68	0.47	866,976	0.75	0.43	421,224
No. stars received	0.57	9.81	866,976	1.32	19.17	421,224

Notes: Summary statistics computed at the developer-month level.

Our empirical analysis relies on a panel dataset constructed from the GitHub REST API. We start from a random sample of one million developer accounts and collect information about all their public repositories. For each repository, we collect the creation date, the last update, the owner type, the main programming languages, license indicators, the number of stars and forks, and the list of collaborators with their respective contribution volumes. We also have data relating to the developer's social activity on the platform, such as their subscribers, subscriptions, or the stars they have given and received. We then filter out clearly organisational accounts and extremely small users with almost no meaningful activity. The resulting cross-section contains about 47 000 developers, of whom roughly 28% are classified as institutional according to whether they list an employer on their profile or have contributed to at least one organisation-owned repository. The remaining developers are treated as independent.

To study dynamics, we construct a monthly panel at the developer level. For each developer and month, we compute the number of public repositories that are active in that month, the number of contributions made by the owner on these repositories, the total number of contributions including external collaborators, the average number of stars and the number of incoming and outgoing forks, and the number of distinct external contributors. This yields roughly 1.3 million developer-month observations, with about 870 000 for independent developers and 420 000 for institutional ones. Table 1 shows that institutional developers have, on average, more active public repositories (2.9 vs 2.0), more stars (1.3 vs 0.6) and more collaborators (about 71 vs 30 external contributors per developer-month), which is consistent with their involvement in larger and more visible projects.

4.2 Variables of Interest

We consider several dependent variables that capture different dimensions of public open-source activity. The number of public repositories per developer measures the overall scale of a developer's public portfolio. The average number of stars per repository is interpreted as a proxy for visibility and perceived quality. The share of repositories that are outgoing forks reflects the extent to which developers build on pre-existing projects rather than starting new ones from scratch, and therefore how much they rely on reuse and integration. Finally, we use the number of contributions and the number of distinct contributors per repository to summarise activity and collaboration within projects. Taken together, these variables allow us to track both how developers adjust the size of their public footprint and how the remaining projects evolve in terms of visibility, reuse and collaborative intensity after the arrival of generative AI.

In Figure 1, we plot monthly averages for institutional (paid) and independent (unpaid) developers to get an overview of how our variables of interest have evolved over time for

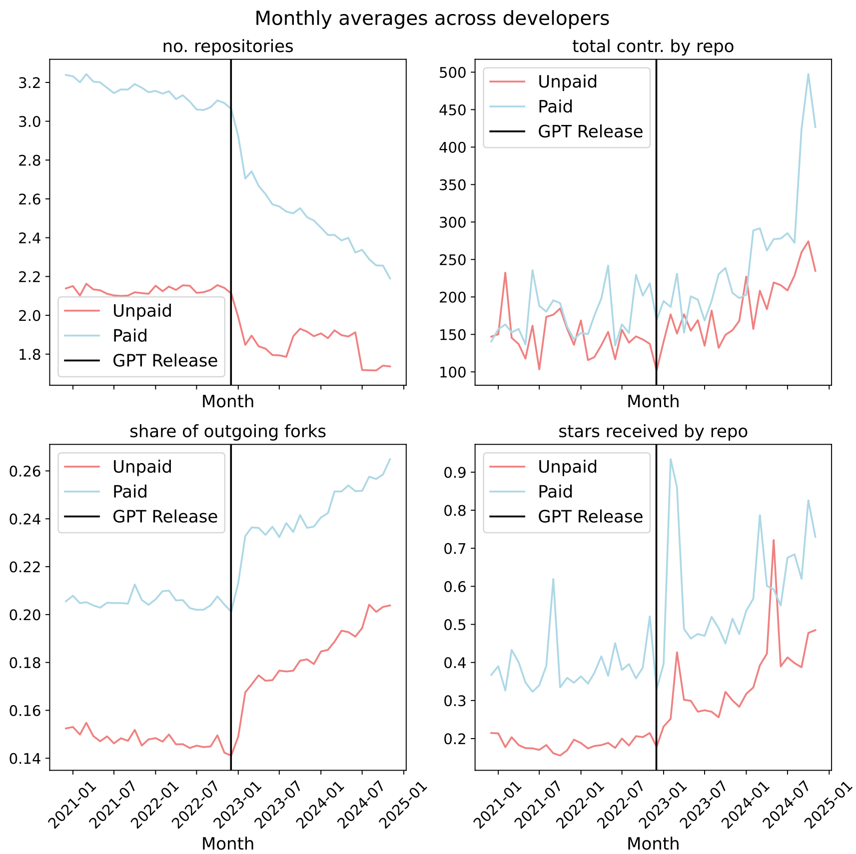

Figure 1: Monthly Averages Across Developers: Institutional vs. Independent

each type of contributor. We observe clear and systematic changes in public open-source activity around the release of ChatGPT-3.5. Before November 2022, institutional developers maintained more public repositories on average than independent ones, but both groups experienced an important drop in the number of active public repositories after the genAI shock, with a visibly larger decline for paid developers. Over the same period, total contributions per repository follow an upward trend, which becomes steeper after 2022 and seems more pronounced for institutional developers. The share of repositories that are outgoing forks is relatively stable before the release and then increases for both groups, again with a particular rise for institutional contributors. Finally, the average number of stars received per repository also increases over time, with a visible upward shift after the release and higher levels for institutional developers. Overall, the series indicate a simultaneous reduction in the number of public repositories and an increase in activity, reuse, and visibility at the level of the remaining repositories, with larger adjustments for institutional than for independent developers.

5 Empirical Strategy

To quantify whether the post-period changes differ systematically between institutional and independent developers, we estimate a two-way fixed effects difference-in-differences model on developer–month outcomes. Let Post_t be an indicator equal to one for months in the post period (from November 2022 onward), and let Inst_i denote the institutional proxy. Our baseline specification is:

$$Y_{it} = \beta \left(\text{Post}_t \times \text{Inst}_i\right) + \alpha_i + \gamma_t + \varepsilon_{it}, \qquad (1)$$

where α_i are developer fixed effects capturing time-invariant heterogeneity (e.g., baseline productivity or persistent specialization) and γ_t are month fixed effects absorbing common shocks and seasonality. Standard errors are clustered at the developer level. The identifying assumption is that, absent the shock, outcomes for institutional and independent developers would have followed parallel trends. In addition to this baseline specification, we estimate event-study models that replace Post_t with relative-time indicators, which allows us to directly examine pre-trends and trace the dynamic adjustment of both groups around November 2022.

6 Results

Table 2: Difference-in-Differences Estimates Results

	no. repositories	tot. contributions by repo	share outgoing forks	stars by repo
inst. × post	-0.059***	0.000	0.007**	0.012**
	(0.009)	(0.020)	(0.003)	(0.004)
Num. Obs.	818,693	767,702	767,702	767,702
R^2	0.669	0.601	0.779	0.680
Std. Errors	by: login	by: login	by: login	by: login
Developer FE	Yes	Yes	Yes	Yes
Month FE	Yes	Yes	Yes	Yes

Significance level: + $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

The results of Table 2 indicate that the introduction of ChatGPT-3.5 is associated with a reorganisation of public open-source activity. In our baseline regressions, the interaction term between institutional status and the post-ChatGPT-3.5 period is negative and statistically significant for the number of repositories: the estimated coefficient of about –0.06 implies that the number of public repositories owned by institutional developers decreases 6% more compared to the number of public repositories of non-institutional developers. By contrast, we do not find a significant differential effect on contributions per repository, suggesting that the activity-intensity differential between institutional and non-intuitional developers per repository is unaffected by ChatGPT-3.5. Hence, our results suggest that institutional developers specialize more aggressively compared to independent developers in response to ChatGPT-3.5, but this specialization on fewer repositories is not accompanied by a higher activity intensity on the remaining active repositories. At the same time, our results are consistent with institutional developers repositories experiencing a sharper increase in reuse and quality following the introduction of ChatGPT-3.5 The interaction term is positive for the share of outgoing forks and for the number of stars per repository: Institutional developers experience an additional increase of around 0.7 percentage points in the share of forked repositories and about 1.2% in stars per repository relative to independent developers.

Hence, the main adjustments seem to concern how many projects developers maintain and which ones they choose to keep visible. One interpretation, consistent with our qualitative discussion of incentives, is that firms and institutional developers rationalise their public open-source strategy in the presence of generative AI. Prior to ChatGPT-3.5, employing developers to maintain numerous public repositories could serve, at least in part, as a form of brand-building or symbolic participation: being visibly active in open source signalled technical excellence and an open-source friendly image. After the diffusion of LLMs, some of this promotional role can be achieved at lower cost through AI-generated examples, documentation and marketing material. The implicit value of maintaining many small public repositories as a branding device therefore declines. Firms then cut back this type of implicit advertising expenditure and reallocate developer time toward a smaller number of projects that are strategically important or have higher technical and reputational returns. Our empirical results are consistent with this mechanism: institutional developers withdraw from a fraction of their public repositories while concentrating effort on fewer, more visible and more reused projects. Independent developers also adjust, but to a lesser extent, which suggests that generative AI reshapes the balance between paid and volunteer contributions by changing both the opportunity cost of developers' time and the signaling value of public open-source work.

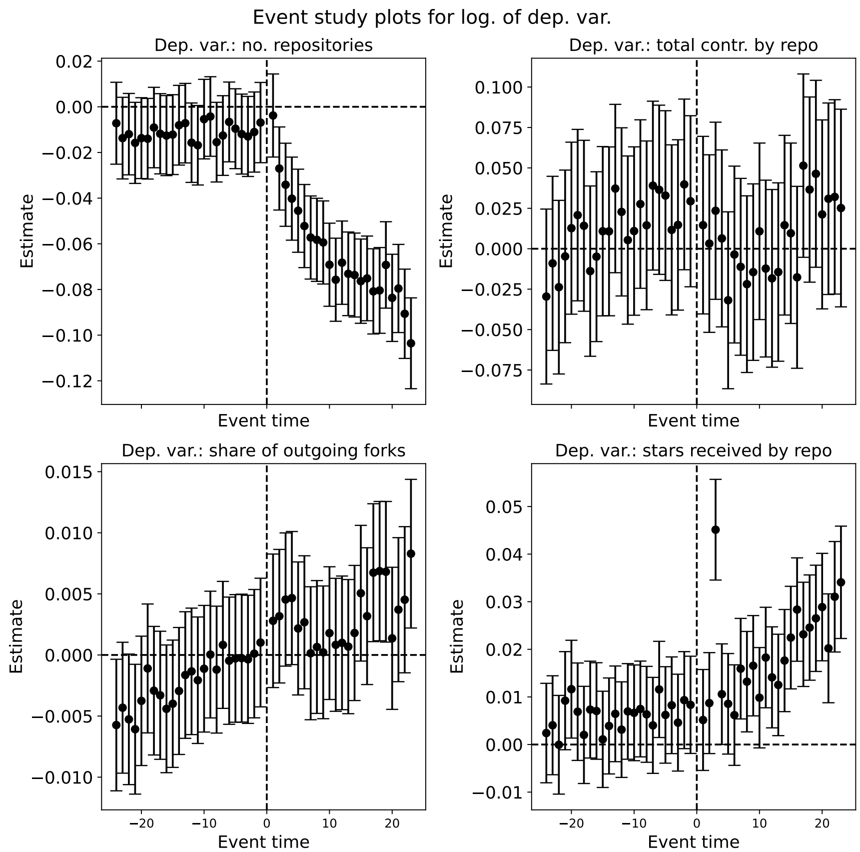

Figure 2: Event-Study Estimates: Institutional vs. Independent Developers

The event-study plots in Figure 2 confirm a clear differential adjustment between institutional and independent developers after the release of ChatGPT-3.5. For the number of repositories, coefficients are close to zero in the pre-period, which is consistent with parallel trends. Following the event date, the estimates become increasingly negative and reach about -0.10 twenty months later, indicating that institutional developers progressively maintain fewer public repositories than independent ones. By contrast, the coefficients for total contributions per repository fluctuate around zero throughout the window with wide confidence intervals, suggesting that there is no systematic difference between the two groups in the level of activity carried out on each remaining repository. For the share of outgoing forks, coefficients are near zero before the event and then rise steadily, implying that institutional developers become more likely than independent developers to base their public projects on existing repositories. A similar pattern emerges for stars received per repository, where post-event coefficients are positive and tend to increase over time, pointing to a relative gain in visibility for the projects maintained by institutional developers. Taken together, these profiles are consistent with a reallocation in which paid developers reduce the number of public repositories they maintain while concentrating on projects that are more connected to the existing codebase and that attract more attention, whereas independent developers adjust less along these dimensions.

7 Robustness Checks

A natural concern is that our institutional proxy may partly capture differences in developer quality rather than organisational status. To assess whether the baseline patterns are simply driven by heterogeneous responses to the AI shock across quality types, we re-estimate the event-study specifications within strata defined by pre-period developer visibility, proxied by the stock of stars. Figures 3 and 4 report the estimates for developers with relatively high and relatively low pre-period stars, respectively. In the high-quality subsample, pre-period coefficients for the number of repositories remain close to zero and display no systematic divergence, while post-period estimates become increasingly negative, indicating a sharper contraction in the public repository portfolio for institutional developers within this quality group. In the low-quality subsample, the repository coefficient is negative throughout the window but does not exhibit an abrupt pre-period divergence around the event date, and it remains negative after November 2022, suggesting that the post-period gap does not mechanically arise from a composition shift toward high-star developers. For outcomes related to project orientation and visibility, both strata display similar qualitative adjustments. The share of outgoing forks and stars received per repository are close to zero in the pre-period and rise in the post-period, with a particularly clear upward trajectory among low-quality developers. By contrast, estimates for total contributions per repository are comparatively imprecise in both strata and fluctuate around zero, indicating no robust differential change in activity intensity conditional on maintaining a repository. Taken together, these robustness checks indicate that our estimates capture systematic differences associated with organizational status beyond pre-existing quality differences.

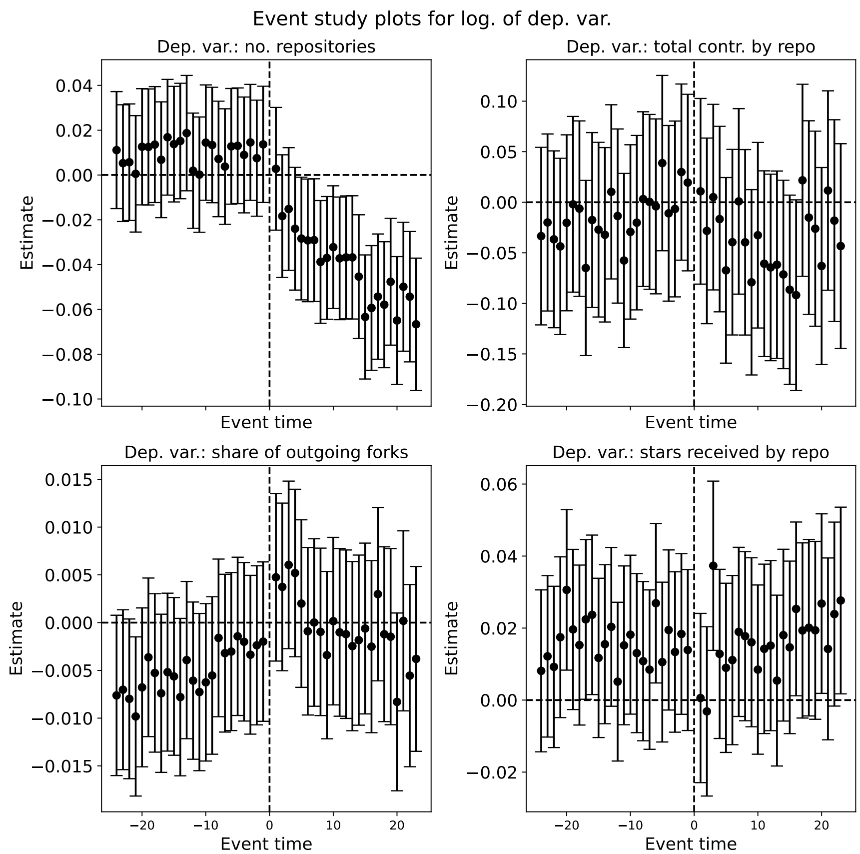

Figure 3: Event-Study Estimates (High-Quality Developers): Institutional vs. Independent Developers

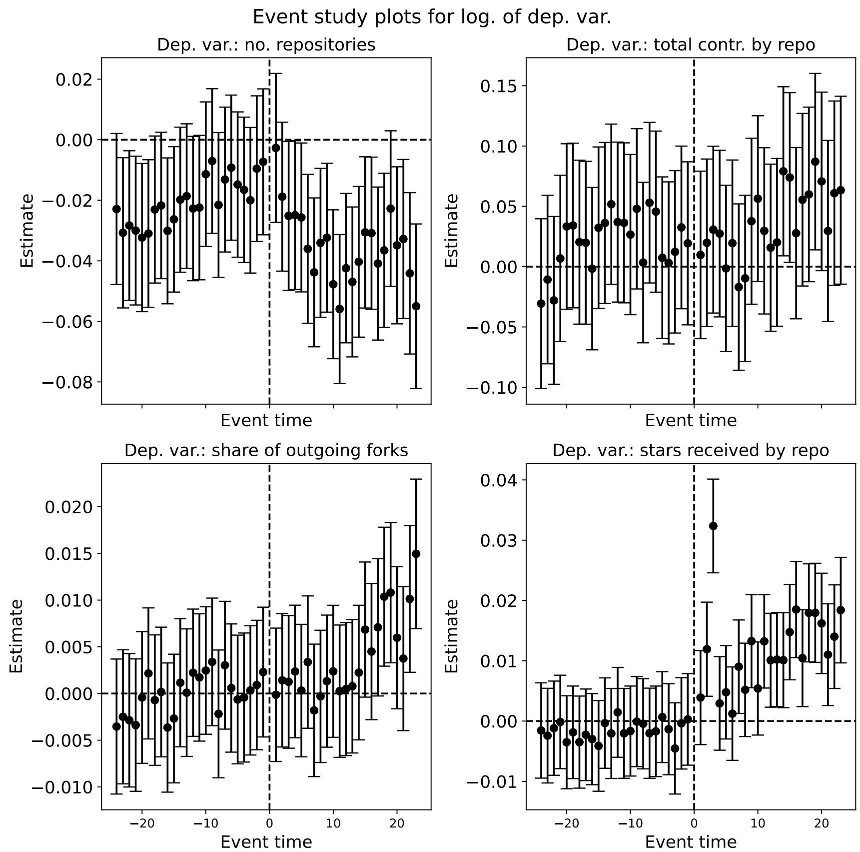

Figure 4: Event-Study Estimates (Low-Quality Developers): Institutional vs. Independent Developers

8 Conclusion

Our preliminary results indicate that the arrival of generative AI led to differential responses between developer types. The main adjustment seems to concern how many projects developers maintain. One interpretation, consistent with our qualitative discussion of incentives, is that firms and institutional developers rationalise their public open-source strategy in the presence of generative AI. Prior to ChatGPT-3.5, employing developers to maintain numerous public repositories could serve, at least in part, as a form of brand-building or symbolic participation: being visibly active in open source signalled technical excellence and an open-source friendly image. After the diffusion of LLMs, some of this promotional role can be achieved at lower cost through AI-generated examples, documentation and marketing material. The implicit value of maintaining many small public repositories as a branding device therefore declines. Firms then cut back this type of implicit advertising expenditure and reallocate developer time toward a smaller number of projects that are strategically important or have higher technical and reputational returns. Our empirical results are consistent with this mechanism: institutional developers withdraw from a fraction of their public repositories while concentrating effort on fewer, more visible and more reused projects. Independent developers also adjust, but to a lesser extent, which suggests that generative AI reshapes the balance between paid and volunteer contributions by changing both the opportunity cost of developers' time and the signaling value of public open-source work. Finally, the robustness analysis mitigates the concern that our institutional proxy merely captures pre-existing differences in developer quality. Stratifying the event-study estimates by pre-period visibility, proxied by the stock of stars, yields qualitatively similar adjustment patterns within both higher- and lower-visibility subsamples. This suggests that the differential post-2022 response we attribute to organisational status is not simply a by-product of comparing inherently higher- versus lower-quality contributors.

References

Bitzer, J., Schrettl, W. and Schröder, P. J. (2007). Intrinsic motivation in open source software development. *Journal of comparative economics*. 35(1), 160–169.

Bui, Q., Tu, Q. J., Liu, H., Aube, J., Chen, H., Song, Y., Hu, J., Lv, Y. and Li, L. (2024). Adoption of Artificial Intelligence in Online Communities: A Socio-Technical Perspective.

Burtch, G., Lee, D. and Chen, Z. (2024). Generative ai degrades online communities. *Communications of the ACM*. 67(3), 40–42.

Cai, Y., Liang, P., Wang, Y., Li, Z. and Shahin, M. (2025). Demystifying issues, causes and solutions in LLM open-source projects. *Journal of Systems and Software*. 227, 112452.

Carillo, K., Huff, S. and Chawner, B. (2014). Understanding contributor behavior within large free/open source software projects: a socialization perspective.

Cui, Z. K., Demirer, M., Jaffe, S., Musolff, L., Peng, S. and Salz, T. (2025). The effects of generative AI on high-skilled work: Evidence from three field experiments with software developers. *Available at SSRN 4945566*.

Gambacorta, L., Qiu, H., Shan, S. and Rees, D. M. (2024). *Generative AI and labour productivity: a field experiment on coding*. vol. 1208. Bank for International Settlements, Monetary and Economic Department.

Ghosh, R. A., Glott, R., Krieger, B. and Robles, G. (2002). Free/libre and open source software: Survey and study.

Haese, J. and Peukert, C. (2025). Open at the core: Moving from proprietary technology to building a product on open source software. *Management Science*.

Hoffmann, M., Boysel, S., Nagle, F., Peng, S. and Xu, K. (2025). Generative AI and the Nature of Work. *Harvard Business School Strategy Unit Working Paper*. (25-021), 25–021.

Hou, I., Man, O., Hamilton, K., Muthusekaran, S., Johnykutty, J., Zadeh, L. and MacNeil, S. (2025). 'All Roads Lead to ChatGPT': How Generative AI is Eroding Social Interactions and Student Learning Communities. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. 79–85.

Jullien, N., Viseur, R. and Zimmermann, J.-B. (2025). A theory of FLOSS projects and Open Source business models dynamics. *Journal of Systems and Software*. 224, 112383.

Lakhani, K. R. and Von Hippel, E. (2003). How open source software works: "free" user-to-user assistance. *Research policy*. 32(6), 923–943.

Lerner, J. and Tirole, J. (2002). Some simple economics of open source. *The journal of industrial economics*. 50(2), 197–234.

Li, X., Zhang, Y., Osborne, C., Zhou, M., Jin, Z. and Liu, H. (2025). Systematic literature review of commercial participation in open source software. *ACM Transactions on Software Engineering and Methodology.* 34(2), 1–31.

Meissonier, R., Bourdon, I., Houze, E., Amabile, S. and Boudrandi, S. (2010). Comprendre les motivations des développeurs de l'open source à partir de leur participation. *Systèmes d'information & management.* 15(2), 71–97.

Mohamed, A., Assi, M. and Guizani, M. (2025). The impact of LLM-assistants on software developer productivity: A systematic literature review. *arXiv preprint arXiv:2507.03156.*

Orman, W. H. (2008). Giving it away for free? The nature of job-market signaling by open-source software developers. *The Nature of Job-Market Signaling by Open-Source Software Developers (January 23, 2008).*

Peng, S., Kalliamvakou, E., Cihon, P. and Demirer, M. (2023). The impact of ai on developer productivity: Evidence from github copilot. *arXiv preprint arXiv:2302.06590.*

Quinn, M. and Gutt, D. (2025). Heterogeneous effects of generative artificial intelligence (GenAI) on knowledge seeking in online communities. *Journal of Management Information Systems.* 42(2), 370–399.

Ryan, R. M. and Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology.* 25(1), 54–67.

Shan, G. and Qiu, L. (2025). Examining the impact of generative ai on users' voluntary knowledge contribution: Evidence from a natural experiment on stack overflow. *Information Systems Research.*

Xiao, T., Hata, H., Treude, C. and Matsumoto, K. (2024). Generative AI for pull request descriptions: Adoption, impact, and developer interventions. *Proceedings of the ACM on Software Engineering.* 1(FSE), 1043–1065.

Yeverechyahu, D., Mayya, R. and Oestreicher-Singer, G. (2024). The impact of large language models on open-source innovation: Evidence from github copilot. *arXiv preprint arXiv:2409.08379.*

Zhang, Y., Qin, M., Stol, K.-J., Zhou, M. and Liu, H. (2024). How are paid and volunteer open source developers different? A study of the rust project. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering.* 1–13.