Beyond Productivity Gains: Analyzing the Effects of AI-assisted Coding Tools on Developer Experience within a Brazilian Fintech Organization

Sabrina C. Winckler
Institute of Informatics (INF) - UFRGS
Porto Alegre, RS, 91501970, Brazil
sabrina.winckler@inf.ufrgs.br

Josiane Kroll
Asper School of Business - University of Manitoba
Winnipeg, MB, R3T 2N2, Canada
Josiane.Kroll@umanitoba.ca

Leticia S. Machado
Institute of Informatics (INF) - UFRGS
Porto Alegre, RS, 91501970, Brazil
leticia.machado@inf.ufrgs.br

Abstract

This paper presents an ongoing case study conducted at a Brazilian financial technology company that introduced GitHub Copilot and Windsurf in November 2024. Grounded in the Developer Experience (DevEx) framework, this study investigates how AI-assisted coding tools affect developer productivity, cognitive load, and team workflows. We compare development practices before and after adopting AI-assisted coding tools in two comparable periods (Q3-2024 vs. Q3-2025), covering 78 repositories and 27 developers active in both observation periods, alongside GitHub Copilot adoption data for the full 33-developer Q3-2025 workforce. Using repository mining, telemetry, and participant observation, we find that the impact of AI-assisted coding tools is highly context-dependent: it accelerates well-structured, routine work but increases friction and coordination costs in reactive, incident-driven tasks. This reveals a productivity paradox in which individual efficiency gains do not readily translate into system-level performance improvements. The study offers practical guidance for organizations, showing that adoption and usage metrics alone cannot capture how these tools shape engineering workflows, developer experience, and delivery performance.

Keywords

Developer Experience, AI-assisted Coding Tools, AI, GitHub Copilot, Windsurf, Software Productivity, Software Engineering

1 Introduction

Since 2023, the adoption of AI-assisted coding tools in software development workflows has expanded rapidly. Solutions like GitHub Copilot and Windsurf have become part of everyday coding practices in organizations across diverse sectors and sizes. However, their overall impact on the software development team is still largely unquantified within the software industry [1, 7].

Most prior research on this topic is based on controlled experiments, short-term empirical studies, or self-reported satisfaction metrics [5]. Far fewer studies analyze long-term, system-level indicators—such as deployment frequency, incident resolution time, or code review throughput—that more directly capture organizational performance. This gap is especially notable in the Brazilian software industry, where evidence of the organizational impact of AI-assisted development is limited. Such evidence is particularly critical for fintechs, whose engineering teams must meet strict regulatory requirements while ensuring high reliability, security, and delivery performance.

This paper presents an ongoing mixed-methods case study undertaken within a Brazilian fintech that initiated the adoption of GitHub Copilot and Windsurf in November 2024. We investigate three research questions:

- **RQ1:** How do AI-assisted coding tools affect feedback loops (e.g., CI/CD pipelines, code review cycles, and access to technical information)?
- **RQ2:** How do AI-assisted coding tools affect cognitive load (e.g., code comprehension, AI-generated code validation, prompt formulation, and incident resolution)?
- **RQ3:** How do AI-assisted coding tools affect flow state (e.g., developer-reported experience, task focus, and learning)?

The study employs a multi-source data collection strategy, combining repository mining, CI/CD telemetry, Copilot telemetry, Jira records, and qualitative observations, interpreted within the DevEx framework [2]. This work extends a previously published study [6], which reported qualitative findings and preliminary telemetry covering a subset of the observation period analyzed here.

2 Industry Context and Motivation

Organizations are rapidly investing in AI coding assistants with expectations of higher productivity and faster software delivery. However, measuring the actual organizational impact of these tools remains challenging.

To address this challenge, we use Developer Experience (DevEx) as an analytical lens. DevEx conceptualizes software development through three dimensions: *feedback loops* (speed and quality of responses to developer actions), *cognitive load* (the mental effort required to complete a task), and *flow state* (the degree to which developers sustain focused, uninterrupted work) [2, 3]. Together, these dimensions provide a structured lens for examining how AI-assisted coding tools reshape developers' daily work, from information access and problem-solving to collaboration and decision-making.

3 Study Design

This study extends our previous exploratory work [6].While [6] presented initial qualitative findings and early GitHub Copilot telemetry, this work adds a longitudinal before-and-after study, repository mining, delivery metrics, statistical analyses, and organizational data gathered over the course of one year, shifting the focus from early perceptions of AI adoption to a broader assessment of its organizational impact. During the study, AI coding assistants were available. Windsurf was initially part of the organization's AI tool set but was discontinued in July 2025, after which GitHub Copilot

became the primary supported assistant. We combine a year-long analysis of repository activity, AI usage telemetry, software delivery metrics, and qualitative evidence from the adoption process. The analysis compares Q3-2024 (pre-adoption) and Q3-2025 (post-adoption), offering a natural before-and-after view in production environments. The study uses different populations depending on the analysis. Quantitative repository mining and software delivery analyses were restricted to the 27 developers who were active in both Q3-2024 and Q3-2025, ensuring direct comparability between the two periods. In contrast, GitHub Copilot adoption metrics describe the entire Q3-2025 workforce, comprising 33 active developers.

Repository, delivery, and AI telemetry data: Data were collected from GitLab and Bitbucket repositories, including GitHub Copilot telemetry when available, complemented by information on Windsurf adoption and usage gathered through organizational records and qualitative evidence. As noted above, the resulting dataset comprised 78 repositories and 27 developers active in both quarters. Metrics included commit frequency and type distribution, code churn, pull request and merge request cycle times, CI/CD pipeline frequency, duration, and success rates, issue volume, and resolution times from Jira, and GitHub Copilot telemetry obtained through Copilot Metrics Viewer v2.0.2.

Qualitative evidence: To understand how developers perceive AI-assisted coding tools, we incorporated findings from participant observation from our previous study [6]. To explore the mechanisms behind the quantitative trends, we also conduct ongoing collaborative retrospectives and semi-structured interviews. Retrospectives use a structured board inspired by DevEx in Action [2], and interviews follow a protocol covering six themes from the DevEx framework and the SPACE model (Satisfaction and well-being, Performance, Activity, Communication and collaboration, and Efficiency and flow). Both instruments use indirect questioning to reduce social desirability bias. Ethics approval was granted under protocol CAAE 87497425.7.0000.5347 (UFRGS).

4 Findings

This section first characterizes the adoption of AI-assisted coding tools, then presents findings by research question. For each RQ, we combine repository activity, software delivery metrics, AI telemetry, and qualitative evidence. Quantitative findings reported in this section are newly collected for this study, whereas qualitative quotations reproduced in Figure 3 originate from our previous study [6].

4.0.1 Characterizing AI-assisted Coding Adoption: Among the 33 developers active during the observation period, 29 had access to GitHub Copilot and 28 used it at least once, corresponding to a 96.6% adoption rate among licensed users. During the observation period, developers produced 37,692 code completions and 17,729 chat interactions. Additionally, two developers tested Windsurf as another AI-assisted development tool.

Figure 1 shows stable engagement throughout the quarter, with approximately 17–20 active users per week. Developers accepting code suggestions closely tracked overall activity levels.

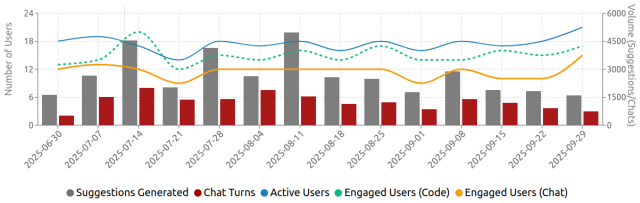

Figure 1: Weekly GitHub Copilot usage in Q3-2025

Chat-based interactions remained lower than code suggestion usage, indicating that GitHub Copilot was primarily used for implementation rather than conversational assistance. Despite usage peaks during periods of higher development activity, engagement remained stable throughout the quarter.

Usage intensity varied throughout the quarter, with a pronounced peak in August that coincided with a phase of elevated development activity. Despite these fluctuations, overall engagement remained stable, suggesting that GitHub Copilot adoption was sustained over time and embedded into day-to-day engineering practices.

These findings indicate that AI adoption extended beyond tool access. Developers integrated AI assistance into their software engineering workflows, primarily using task-specific, fine-grained code generation rather than general-purpose, conversational interactions. This distinction is particularly relevant when interpreting the broader DevEx findings presented in the following sections, as it suggests that the primary impact of AI adoption occurred within development activities rather than through conversational knowledge support.

4.1 RQ1 — Feedback Loops

To answer Q1, we analyzed CI/CD pipeline performance, code review cycles, and developers' access to technical information. Table 1 summarizes the key delivery-related metrics observed before and after AI adoption.

Table 1: Key feedback loop metrics — Q3-2024 vs. Q3-2025

Metric	GitLab	Bitbucket
Pipeline frequency	↓33.7%	↑185.8%
Pipeline duration	↑79.0%	↑2.7%
Pipeline success rate	−7.9 pp	−16.9 pp
PR creation	↓22.6%	↑16.0%
Median review time	↑123%	↓53.1%
Median merge time	↑133.3%	↓40.3%

The results reveal a workflow-dependent pattern. In Bitbucket, which primarily appears to support planned feature development, pull requests moved through review and merge stages faster than in the baseline period, suggesting reduced friction in structured development activities. Pipeline execution frequency also increased substantially, indicating a higher rate of delivery activity.

In contrast, GitLab workflows appear to be associated with more incident-driven work. This conclusion draws on how the organization tracked issues: Jira incident tickets were routinely associated with GitLab projects via their corresponding repository references. Consequently, the GitLab repositories primarily represented day-to-day maintenance tasks and incident resolution work. They also showed extended review cycles, reduced pipeline frequency, and longer execution times. Pipeline success rates dropped on both platforms, suggesting that gains in development throughput did not necessarily translate into more stable delivery.

These findings are consistent with qualitative observations reported in our previous exploratory study [6]. Developers described AI-assisted coding as valuable tools for quickly obtaining technical information, understanding unfamiliar code, and accelerating implementation tasks. However, participants also reported that AI-generated suggestions were less effective when dealing with complex operational scenarios that required deeper contextual knowledge.

4.2 RQ2 — Cognitive Load

To answer Q2, we analyzed repository activity, incident-management metrics, GitHub Copilot telemetry, and qualitative evidence on code comprehension, AI code validation, prompt formulation, and incident resolution. Figure 2 provides an overview of the evolution of development activity following AI adoption. Relative to the 2024 baseline, the 2025 profile exhibits an increased proportion of Feature-oriented commits, a higher commit frequency, and a reduced normalized code churn per contribution. Collectively, these patterns suggest a shift toward smaller, incremental, and task-focused modifications, with greater emphasis on introducing new functionality than on extensive, cross-cutting codebase revisions.

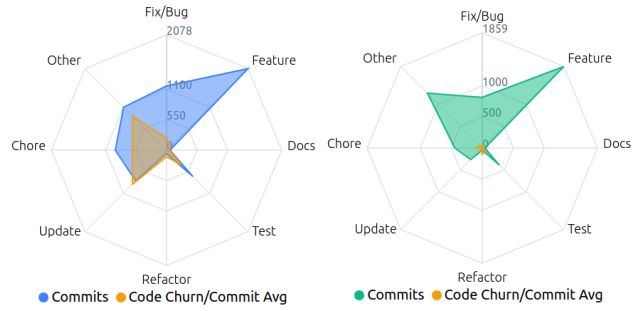

Figure 2: Radar charts showing the distribution of commit message types, commit frequency, and normalized average code churn across 2024 (left) and 2025 (right) quarters.

Commit frequency rose on both platforms, while code churn per commit fell for planned work. The distribution of commit types also shifted, with more Feature commits and fewer Update commits, suggesting developers devoted more effort to implementation-focused tasks while making more incremental changes.

GitHub Copilot telemetry provides additional context for these behavioral changes. Suggestion acceptance reached 30.2% by count but only 21.9% by lines of code, indicating a preference for smaller and more targeted suggestions rather than large-scale code generation. This pattern aligns with the observed reduction in code churn, suggesting that developers frequently incorporated AI-generated recommendations as focused contributions within existing development workflows rather than relying on extensive code generation.

However, incident-related metrics revealed a more complex picture. Although the number of incident tickets decreased by 35.2%, median resolution time increased by 377%. While fewer incidents were reported, the remaining issues required substantially more time to resolve. This finding suggests that AI-assisted coding did not eliminate cognitively demanding work, particularly in contexts characterized by uncertainty, operational pressure, and the need for deep system understanding.

Qualitative evidence from our previous exploratory study [6] helps explain these observations. Developers reported spending less time on routine implementation tasks but more time validating AI-generated outputs, refining prompts, understanding generated solutions, and deciding whether suggestions should be accepted or rejected. Engineering leaders also highlighted the need to actively "steer the agent" and review its outputs before integration into production systems.

Taken together, these findings indicate that AI-assisted coding reshaped rather than reduced cognitive load. Routine implementation activities appear to have become more efficient, enabling developers to work through smaller and more focused changes. At the same time, cognitive effort shifted toward supervision, validation, prompt formulation, and decision-making activities, particularly in incident-driven scenarios where contextual reasoning remained essential.

4.3 RQ3 — Flow State

To answer Q3, we draw on qualitative evidence from our prior exploratory study [6]. Figure 3 summarizes practitioner perspectives across the three DevEx dimensions. The study included practitioners using GitHub Copilot (C) and Windsurf (W), reflecting the tools available in June 2025, before Windsurf was discontinued.

Participants associated AI-assisted coding with positive flow experiences, particularly improved task organization, accelerated learning, and implementation support. However, the figure also highlights cognitive-load concerns: practitioners emphasized the need to guide, review, and validate AI-generated code. These findings suggest that AI assistants can support flow during planned work, but the benefits require continued human oversight.

5 Discussion

Prior studies report a 27% acceptance rate for GitHub Copilot suggestions and up to 55% faster task completion in controlled settings [7], along with improvements in focus and learning [1]. At the same time, concerns about overconfidence, code quality, and human oversight remain [4, 5].

Our results reveal a productivity paradox: AI-assisted tools improved task-level outcomes, such as merge times and commit focus, but were associated with worse system-level outcomes in reactive workflows. This supports the amplification hypothesis, suggesting that AI tools reinforce existing workflow characteristics rather than correct them.

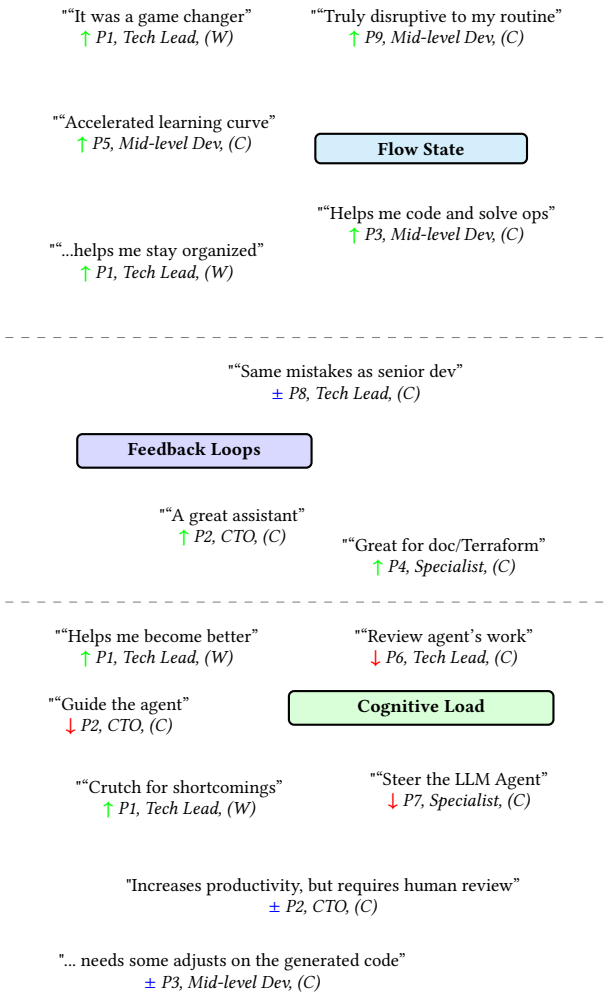

Figure 3: Practitioners' perspectives grouped by DevEx recurring themes (n=9). Source: [6]

Alternative interpretations must be considered. The comparison covers different quarters and organizational contexts, so team composition, repository maturity, project mix, process changes, and operational pressures may have influenced the observed variations. Although repository mining focused on developers active in both periods, causal claims should be treated with caution.

The findings also highlight the rapid evolution of AI tooling ecosystems. Windsurf was launched and phased out within a few months, underscoring the importance of longitudinal research that tracks both technological effects and changing adoption patterns. As a single-fintech case study, the findings should not be assumed to generalize broadly; Organizational culture, development practices, regulatory context, and AI adoption strategies may differ across companies. Rather than providing definitive answers, the results offer analytical insights and hypotheses for future research involving a broader set of organizations.

6 Implications for the Software Industry

The results suggest three practical implications. First, organizations should evaluate AI adoption alongside delivery and reliability metrics rather than usage metrics alone. Second, expected productivity gains should account for workflow characteristics, as benefits were stronger in structured than reactive work. Third, organizations should anticipate a shift in cognitive load from routine implementation toward validation, supervision, and decision-making, adapting review practices and developer training accordingly.

7 Conclusions

This study shows that the impact of AI-assisted software development extends beyond individual productivity gains. While AI tools accelerated structured development tasks, their effects varied across workflows and were not consistently reflected in system-level outcomes.

The results indicate that AI adoption should be understood as a socio-technical transformation that influences developer experience, software delivery workflows, and organizational outcomes. The apparent paradox may stem from how AI adoption interacts with existing workflow properties, rather than from AI use in isolation. Focusing solely on adoption or on usage is therefore insufficient to accurately assess the value created by AI-assisted development.

Future work includes completing the qualitative phase through retrospectives and interviews, extending the longitudinal analysis, and refining methods for assessing the organizational impact of AI-assisted software development.

Artifact Availability

Scripts and replication instructions are available at https://doi.org/10.5281/zenodo.21938780.

Acknowledgements

The authors thank the fintech team for their participation and openness in sharing operational data. This work was supported by FAPERGS, Grant No. 25/2551-0000883-2 and research was approved under ethics protocol CAAE 87497425.7.0000.5347 (UFRGS).

References

[1] Thomas Dohmke, Marco Iansiti, and Greg Richards. 2023. *Sea Change in Software Development: Economic and Productivity Analysis of the AI-Powered Developer Lifecycle.* Technical Report 2306.15033. arXiv.
[2] Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler. 2023. DevEx: What Actually Drives Productivity. *ACM Queue* 21, 2 (2023). doi:10.1145/3595878
[3] Michaela Greiler, M. Storey, and Abi Noda. 2022. An Actionable Framework for Understanding and Improving Developer Experience. *IEEE Transactions on Software Engineering* 49 (2022), 1411–1425. doi:10.1109/tse.2022.3175660
[4] Hussein Mozannar et al. 2024. Reading Between the Lines: Modeling User Behavior and Costs in AI-Assisted Programming. In *Proc. ACM CHI.*
[5] Daniel Russo. 2024. Navigating the Complexity of Generative AI Adoption in Software Engineering. *ACM Trans. Softw. Eng. Methodol.* 33, 5 (2024).
[6] Sabrina C. Winckler, Jéssica F. Ribeiro, Leticia S. Machado, and Josiane Kroll. 2025. AI-assisted Collaboration: Exploring Developer Experience with GitHub Copilot and Windsurf. In *Proceedings of the IEEE Computer Society.* doi:10.1109/IEEECS.2025.11429684
[7] Albert Ziegler et al. 2024. Measuring GitHub Copilot's Impact on Productivity. *Commun. ACM* 67, 3 (2024), 54–63.