

Why Are Agentic Pull Requests Merged or Rejected? An Empirical Study

Sien Reeve Ordonez Peralta O.
Waseda University
Tokyo, Japan
sienreeve46@gmail.com

Naoyasu Ubayashi
Waseda University
Tokyo, Japan
ubayashi@aoni.waseda.jp

Hiroki Mukai
Ritsumeikan University
Kyoto, Japan
is0692vs@ed.ritsumei.ac.jp

Hidetake Tanaka
Nara Institute of Science and
Technology
Nara, Japan
tanaka.hidetake.te0@naist.ac.jp

Fumika Hoshi
Waseda University
Tokyo, Japan
fumikah@fuji.waseda.jp

Inase Kondo
Osaka University
Osaka, Japan
i-kondou@ist.osaka-u.ac.jp

Norihiro Yoshida
Ritsumeikan University
Kyoto, Japan
norihiro@fc.ritsumei.ac.jp

Youmei Fan
Nara Institute of Science and
Technology
Nara, Japan
fan.youmei.fs2@is.naist.jp

Hironori Washizaki
Waseda University
Tokyo, Japan
washizaki@waseda.jp

Yoshiki Higo
Osaka University
Osaka, Japan
higo@ist.osaka-u.ac.jp

Kazuki Kusama
Kyushu University
Fukuoka, Japan
kusama@posl.ait.kyushu-u.ac.jp

Abstract

AI coding agents increasingly submit pull requests (Agentic-PRs) to open-source repositories, yet their performance is commonly assessed using merge and rejection outcomes alone. We hypothesized that these outcome labels do not reliably reflect agent capability without considering review interactions. To test this, we conducted a decision-oriented analysis of 11,048 closed Agentic Pull Requests, refined to 9,799 human-reviewed PRs, and manually inspected 717 representative cases to recover decision rationale from interaction artifacts. We found that rejection outcomes substantially overstate agent error: only 35.7% of rejected PRs reflected clear agentic failures, while 31.2% were driven by workflow constraints and 33.1% lacked observable decision rationale. Among merged PRs, 15.4% required explicit reviewer involvement through feedback or direct commits, and 5.5% showed no visible interaction trace. We further observed systematic differences across agents, with Copilot and Devin more often embedded in reviewer-mediated workflows, while Codex and Cursor PRs were typically merged with minimal interaction. These results reject the assumption that PR outcomes alone capture agent performance and demonstrate the need for interaction-aware evaluation grounded in review behavior.

CCS Concepts

• **Software and its engineering** → **Collaboration in software development; Software configuration management and version control systems; Software development techniques.**

Keywords

AI-assisted development, Agentic pull requests, Human-AI collaboration, Software engineering, Code review

ACM Reference Format:

Sien Reeve Ordonez Peralta O., Fumika Hoshi, Hironori Washizaki, Naoyasu Ubayashi, Inase Kondo, Yoshiki Higo, Hiroki Mukai, Norihiro Yoshida, Kazuki Kusama, Hidetake Tanaka, and Youmei Fan. 2026. Why Are Agentic Pull Requests Merged or Rejected? An Empirical Study. In *23rd International Conference on Mining Software Repositories (MSR '26)*, April 13–14, 2026, Rio de Janeiro, Brazil. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3793302.3793575>

1 Introduction

AI coding agents increasingly submit *Agentic Pull Requests* (Agentic-PRs) that participate directly in real-world software development. Evidence from the AIDev dataset shows that agent-authored PRs span thousands of repositories [11], reflecting a broader shift toward AI-native software engineering (SE 3.0) [8]. Prior work reports productivity gains from AI-assisted coding [5, 16], changes in review practices [2], and growing concerns related to maintainability [13] and security risks [15].

Despite widespread adoption, it remains unclear why Agentic-PRs are merged or rejected. Existing studies predominantly rely on outcome-based measures such as merge rates or approval frequencies, which provide limited insight into how review decisions are



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

MSR '26, Rio de Janeiro, Brazil

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2474-9/26/04

<https://doi.org/10.1145/3793302.3793575>

formed. In practice, pull request decisions are shaped by interaction-level factors, including reviewer comments, CI outcomes, commit history, and workflow actions, rather than PR states alone.

This gap complicates the interpretation of both rejection and merge outcomes. Rejections may reflect workflow constraints or undocumented reviewer decisions rather than agentic failures, while merged PRs may depend on varying degrees of human involvement. Without examining interaction evidence, outcome-based evaluations risk misrepresenting agent capability.

To address this limitation, we analyze *closed* Agentic-PRs with evidence of human evaluation. Our dataset includes 11,048 closed PRs from repositories with at least 500 stars, refined to 9,799 human-reviewed PRs. We manually inspect a stratified sample of 717 PRs to examine how merge and rejection decisions are formed in practice.

This study is guided by two research questions:

RQ1: What drives the rejection of Agentic-PRs?

Motivation: Rejections may stem from agentic failures, workflow constraints, or undocumented reviewer decisions. Distinguishing among these factors is necessary to avoid misattributing rejection outcomes to agent capability alone.

RQ2: How does human involvement shape successful Agentic-PRs?

Motivation: We examine whether merged PRs reflect autonomous agent performance or depend on collaborative refinement through reviewer feedback or direct human intervention.

Key Takeaway. Analysis of 717 manually inspected Agentic-PRs demonstrates that pull request outcomes alone are insufficient for assessing agent performance. Only 35.7% of rejections correspond to clear agentic failures, while 31.2% are driven by workflow factors and 33.1% lack observable decision rationale. Moreover, 15.4% of merged PRs require explicit reviewer involvement, indicating that both success and failure outcomes depend on interaction-level processes that are invisible in merge or rejection labels alone.

Contributions. This paper makes three contributions: (1) a decision-oriented empirical analysis that disentangles agentic failures from workflow-driven and undocumented rejection outcomes in Agentic-PRs; (2) a quantitative characterization of human involvement in successful Agentic-PRs, including feedback loops and reviewer-applied commits; and (3) empirical evidence that outcome-based metrics systematically conflate agent capability with repository workflows, motivating interaction-aware evaluation of AI coding agents.

2 Motivation

Prior empirical studies show that AI coding agents can improve developer productivity and accelerate task completion [5, 16], while also introducing challenges related to review effort, maintainability, and security [13, 15]. Large-scale analyses of Agentic-PRs, including those enabled by the AIDev dataset [11], predominantly assess agent performance using outcome-based measures such as approval rates or merge frequency. However, these measures provide limited insight into how merge and rejection decisions are formed.

Software engineering research has shown that pull request decisions rely heavily on interaction-level signals, including discussion threads, CI outcomes, and iterative revisions, rather than final PR

Table 1: Per-agent distribution of merged and rejected Agentic Pull Requests in the refined dataset after removing bot-only reviews (Figure 1B).

Agent	Total PRs	(%) of Total	Merged	Rejected
Claude Code	213	2.2	130	83
Copilot	1,429	14.6	839	590
Cursor	788	8.0	563	225
Devin	3,380	34.5	1,813	1,567
OpenAI Codex	3,989	40.7	2,834	1,155
Total	9,799	100.0	6,179	3,620

states alone [7]. Studies of AI-assisted code review similarly indicate that reviewer judgments are often conveyed through unstructured comments or small corrective actions that are not captured by metadata or formal review labels [4, 14]. As a result, outcome-based evaluations may conflate agent behavior with workflow conventions, reviewer intervention, or project-specific practices.

A decision-oriented analysis grounded in interaction evidence is therefore necessary to distinguish agentic failures from workflow-driven and human-mediated outcomes. Such separation enables more accurate interpretation of agent performance and supports a clearer understanding of Human-AI collaboration in software development [9, 12, 19].

3 Data Collection

Step A in Figure 1 summarizes the initial construction of the dataset. We build on the AIDev dataset [11] and restrict the scope to closed Agentic-PRs, defined as pull requests with explicit merge or rejection decisions. To limit the analysis to repositories with established review activity, we apply a threshold of at least 500 stars. After this filtering step, the dataset contains 11,048 Agentic-PRs, including 6,774 merged and 4,274 rejected submissions.

The refined dataset definition is shown in step B in Figure 1. We exclude bot-only reviewed submissions, defined as agent-authored PRs without any human comments, to reduce cases where no human decision context is available. This step removes 1,249 PRs and yields a final dataset of 9,799 human-reviewed Agentic-PRs across multiple agents and repositories.

Table 1 summarizes the composition of the refined dataset by agent, reporting total, merged, and rejected PR counts. The dataset is unevenly distributed across agents: OpenAI Codex and Devin together account for the majority of Agentic-PRs, while Copilot and Cursor contribute smaller but substantial portions, and Claude Code appears least frequently. Across all agents, merged PRs are more common than rejected PRs, reflecting the overall composition of the refined dataset rather than agent-specific merge behavior.

4 Methodology

Step C in Figure 1 depicts the stratified sampling strategy used for manual inspection. From the refined dataset, we draw a sample of 717 PRs, including 353 rejected and 364 merged submissions, balanced across agents and repositories. This stratification ensures

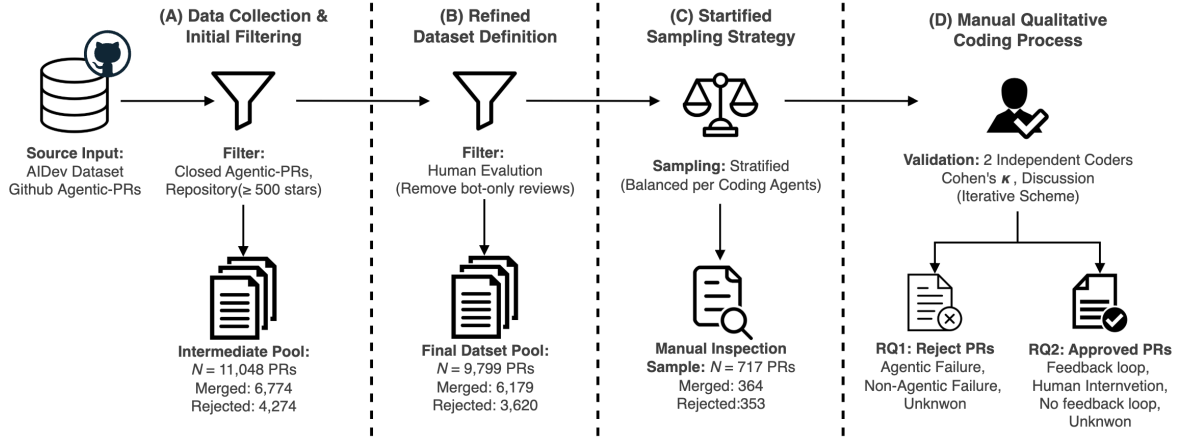


Figure 1: Overview of the study pipeline.

representative coverage of decision outcomes and review interactions and follows prior recommendations for pull request-level empirical studies [18].

The manual qualitative coding process is summarized in step D on Figure 1. We conduct an exploratory empirical analysis of decision making in Agentic-PRs using manual qualitative coding, following established guidelines for empirical software engineering studies involving large language models [3]. The coding scheme is developed iteratively through pilot inspection and calibration. For rejected PRs (RQ1), cases are categorized as agentic failures, non-agentic failures, or unknown when no reliable decision signal is observable in comments, CI results, or closure context. For merged PRs (RQ2), interaction patterns are classified as feedback loops, human intervention, or no feedback loop, with an unknown label used when interaction evidence is missing or ambiguous.

Four annotators participate in the study, with two annotators independently coding each PR. Inter-rater reliability is assessed using Cohen's κ , following best practices for agreement analysis in software engineering research [6, 10]. Disagreements are resolved through discussion, and ambiguous cases are conservatively assigned to the unknown category. All sampling scripts, coding guidelines, and agreement statistics are included in the replication package [1].

5 RQ1: What drives the rejection of Agentic-PRs?

Table 2 reports the distribution of rejection categories in the manually inspected sample of 353 rejected Agentic-PRs. Of these, 126 (35.7%) were categorized as *agentic failures* (AF), 110 (31.2%) as *non-agentic failures* (Non-AF), and 117 (33.1%) as *unknown*. AF cases were associated with observable technical failure signals such as failing checks, failing tests, or reviewer comments indicating that the submitted change did not work. Non-AF cases were associated with observable workflow- or process-related closure signals such as duplicates, superseded changes, inactivity, test PRs, or incorrect submission context. Unknown cases lacked sufficient observable evidence for classification and were often caused by silent repository

Table 2: Distribution of rejection reasons ($N = 353$).

Agent	AF	Non-AF	Unknown	Total
Claude Code	1	4	3	8
Copilot	19	17	21	57
Cursor	15	1	6	22
Devin	37	69	47	153
OpenAI Codex	54	19	40	113
Total	126	110	117	353
Percentage	35.7%	31.2%	33.1%	

Inter-rater reliability: Cohen's $\kappa \approx 0.90$; AF = agentic failure; Non-AF = non-agentic failure.

behaviors (e.g., PRs closed without comments). These cases do not necessarily indicate agentic failure and should be treated separately or excluded from precision-based evaluations to avoid noise. Representative examples are provided in the replication package [1].

The category composition varied across agents. For Devin, 37 of 153 rejected PRs (24.2%) were categorized as AF, 69 (45.1%) as Non-AF, and 47 (30.7%) as Unknown. For OpenAI Codex, 54 of 113 rejected PRs (47.8%) were categorized as AF, 19 (16.8%) as Non-AF, and 40 (35.4%) as Unknown. Copilot and Cursor showed distributions across all three categories, and Claude Code accounted for 8 rejected PRs in the sample, as summarized in Table 2.

RQ1 Answer. Rejection of Agentic-PRs was not primarily driven by agentic failures. Among 353 rejected Agentic-PRs, only 126 (35.7%) were rejected due to observable agentic failures. The remaining rejections were split between non-agentic workflow-related causes (110 PRs, 31.2%) and cases with no observable decision rationale (117 PRs, 33.1%). Thus, a majority of rejected Agentic-PRs were not attributable to identifiable agentic failures.

Table 3: Interaction patterns in merged PRs ($N = 364$).

Agent	FL	HI	No FL	Unknown	Total
Claude Code	0	0	8	0	8
Copilot	13	16	11	9	49
Cursor	1	0	30	2	33
Devin	14	11	74	8	107
OpenAI Codex	0	1	165	1	167
Total	28	28	288	20	364
Percentage	7.7%	7.7%	79.1%	5.5%	100.0%

Inter-rater reliability: Cohen's $\kappa = 1.0$; FL = reviewer feedback followed by agent-only revisions; HI = reviewer commits applied before merge.

6 RQ2: How does human involvement shape successful Agentic-PRs?

Table 3 reports interaction patterns observed in 364 manually inspected merged Agentic-PRs. Most merged PRs, 288 (79.1%), were categorized as *No FL*, indicating merge without an observable feedback loop or reviewer-applied commits. In addition, 28 merged PRs (7.7%) were categorized as *FL*, and 28 (7.7%) as *HI*. The remaining 20 merged PRs (5.5%) were categorized as *Unknown* due to insufficient or ambiguous observable interaction traces.

The distribution of interaction patterns varied across agents. Copilot accounted for 29 of 49 merged PRs (59.2%) with either *FL* or *HI* (13 *FL*; 16 *HI*), and Devin accounted for 25 of 107 (23.4%) with either *FL* or *HI* (14 *FL*; 11 *HI*). OpenAI Codex showed 1 of 167 merged PRs (0.6%) categorized as *HI* and 0 categorized as *FL*. Cursor showed 1 of 33 merged PRs (3.0%) categorized as *FL* and 0 categorized as *HI*. Claude Code showed 0 cases categorized as *FL* or *HI* in the inspected sample. Representative labeled examples (including PR URLs and supporting interaction evidence) were provided in the replication package. [1]

RQ2 Answer. Human involvement in successful Agentic-PRs was observed in 56 of 364 merged PRs (15.4%), split evenly between feedback loops (28 PRs, 7.7%) and reviewer-applied commits (28 PRs, 7.7%). Most merged PRs (288 PRs, 79.1%) were merged with no observed feedback loop, and 20 PRs (5.5%) had ambiguous or insufficient interaction evidence. Human involvement was concentrated in Copilot and Devin PRs, which accounted for 54 of the 56 cases labeled as feedback loop or human intervention in the inspected sample.

7 Discussion

Across the study, we find that **Agentic-PR outcomes cannot be reliably interpreted from merge or rejection labels alone**. Decisions were frequently communicated through interaction-level signals—such as CI behavior, review comments, commit history, and workflow actions—that were not consistently reflected in final PR states. In RQ1, 33.1% of rejected PRs exhibited no observable comments or CI failures, and in RQ2 reviewer involvement was sometimes visible only through follow-up commits. Manual inspection shows that PRs may be closed after reviewers redirect effort, enforce repository conventions, or deprioritize changes without recording an explicit rejection rationale. For example, Copilot PR

#9653 in dotnet/aspire was closed without merge despite extensive discussion, multiple commits, and CI runs, illustrating how outcome-only analyses omit substantial decision context [4, 7].

We also observe that **merging Agentic-PRs does not uniformly correspond to autonomous agent completion**. In RQ2, 15.4% of merged PRs involved explicit reviewer participation, either through feedback followed by agent revisions or through reviewer-applied commits. These interventions commonly addressed missing tests, CI configuration issues, or repository-specific conventions rather than core functionality. A representative case is Copilot PR #3512 in primer/view_components, where a reviewer applied corrective commits before merging without further agent revision. Such cases show that merged outcomes may reflect collaborative completion rather than agent performance alone [14, 17].

Interaction patterns further varied systematically across agents and repository contexts. Devin and Copilot accounted for most workflow-driven rejections and reviewer-involved merges, while OpenAI Codex and Cursor contributed a larger share of merges without revision. Qualitative inspection suggests these differences align with repository practices such as stricter CI gating, iterative review norms, and enforcement of contribution policies. Consequently, observed Agentic-PRs outcomes reflect both agent behavior and the review environments in which agents are deployed, complicating direct cross-agent comparisons.

Limits of attribution motivate interaction-aware evaluation. Manual qualitative inspection reliably identified decision rationale when interaction evidence such as reviewer comments, CI outcomes, or corrective commits was present. However, silent closures and workflow-triggered actions lacked sufficient observable signals for attribution, revealing irreducible uncertainty in reconstructing decisions from PR artifacts alone. This suggests that both manual and automated evaluations should incorporate interaction evidence when available and explicitly represent uncertainty when it is absent, rather than inferring agent performance from outcome labels alone.

8 Threats to Validity

Construct Validity. Our results depend on operational definitions of *agentic failure*, *non-agentic outcome*, and *feedback loop*. Although these constructs were refined through pilot coding, ambiguity remains when reviewer intent is implicit or PR artifacts are sparse. Some decision rationale may be unobservable (e.g., silent closures), leading to an *unknown* category that may hide unrecorded factors. We also lack explicit effort metrics (e.g., review duration, comment volume). Although 15.4% of merges involved human intervention, the associated cost remains unquantified.

Internal Validity. Manual coding may introduce subjectivity. We mitigated this through independent dual annotation and high inter-rater agreement (Cohen's κ), though subtle or multi-causal cases may still be misclassified. Stratified sampling reduces bias, but unobserved contextual factors cannot be fully controlled.

External Validity. Our study focuses on high-star open-source repositories and five widely used coding agents. While this improves data quality, findings may not generalize to smaller projects, proprietary settings, or future agent systems, where workflows and review practices may differ.

9 Conclusion and Future Work

This study examined decision formation in Agentic Pull Requests by analyzing interaction artifacts from 11,048 closed PRs, including 717 manually inspected cases drawn from 9,799 human-reviewed submissions. We show that PR outcomes alone are unreliable indicators of agent performance: 33.1% of rejected PRs lacked observable decision rationale, while only 35.7% reflected clear agentic failures. Among merged PRs, 15.4% required explicit reviewer involvement through feedback loops or direct human commits. We further observed systematic differences across agents, with Copilot and Devin more often embedded in workflow-heavy repositories and Codex and Cursor more frequently merged without interaction, indicating that observed outcomes reflect deployment context as much as agent-generated code.

These findings motivate evaluation approaches that move beyond outcome-based metrics and incorporate interaction-level evidence such as review comments, CI signals, commit histories, and authorship changes, while accounting for uncertainty when signals are absent. Future work should leverage manually curated ground truth to automate detection of workflow-driven closures, reviewer intervention, and feedback loops at scale, and develop interaction-aware benchmarks that better reflect agent effectiveness in real-world software development.

References

- [1] 2026. Agentic-PRs-Analysis: Replication Package for Decision-Oriented Analysis of Agentic Pull Requests. <https://github.com/cn-eveer/Agentic-PRs-Analysis>. Accessed: 2026-01-25.
- [2] Adam Alami and Neil A. Ernst. 2025. Human and Machine: How Software Engineers Perceive and Engage with AI-Assisted Code Reviews Compared to Their Peers. arXiv:2501.02092 [cs.SE] <https://arxiv.org/abs/2501.02092>
- [3] Sebastian Baltes, Florian Angermeier, Chetan Arora, Marvin Muñoz Barón, Chunyang Chen, Lukas Böhme, Fabio Calefato, Neil Ernst, Davide Falessi, Brian Fitzgerald, Davide Fucci, Marcos Kalinowski, Stefano Lambiasi, Daniel Russo, Mircea Lungu, Lutz Prechelt, Paul Ralph, Rijnard van Tonder, Christoph Treude, and Stefan Wagner. 2025. Guidelines for Empirical Studies in Software Engineering involving Large Language Models. arXiv:2508.15503 [cs.SE] <https://arxiv.org/abs/2508.15503>
- [4] Umut Cihan, Vahid Haratian, Arda İçöz, Mert Kaan Gül, Ömercan Devran, Emircan Furkan Bayendur, Baykal Mehmet Uçar, and Eray Tüzün. 2024. Automated Code Review In Practice. arXiv:2412.18531 [cs.SE] <https://arxiv.org/abs/2412.18531>
- [5] Zheyuan Kevin Cui, Mert Demirel, Sonia Jaffe, Leon Musolff, Sida Peng, and Tobias Salz. 2025. The effects of generative AI on high-skilled work: Evidence from three field experiments with software developers. Available at SSRN 4945566 (2025).
- [6] Jessica Díaz, Jorge Pérez, Carolina Gallardo, and Ángel González-Prieto. 2021. Applying Inter-rater Reliability and Agreement in Grounded Theory Studies in Software Engineering. arXiv:2107.11449 [cs.SE] <https://arxiv.org/abs/2107.11449>
- [7] Mehdi Golzadeh, Alexandre Decan, and Tom Mens. 2019. On the Effect of Discussions on Pull Request Decisions. In *BENEVOLE*.
- [8] Ahmed E. Hassan, Gustavo A. Oliva, Dayi Lin, Boyuan Chen, Zhen Ming, and Jiang. 2024. Towards AI-Native Software Engineering (SE 3.0): A Vision and a Challenge Roadmap. arXiv:2410.06107 [cs.SE] <https://arxiv.org/abs/2410.06107>
- [9] Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, and Haoyu Wang. 2024. Large Language Models for Software Engineering: A Systematic Literature Review. arXiv:2308.10620 [cs.SE] <https://arxiv.org/abs/2308.10620>
- [10] Klaus Krippendorff. 2011. Computing Krippendorff's alpha-reliability. (2011).
- [11] Hao Li, Haoxiang Zhang, and Ahmed E. Hassan. 2025. The Rise of AI Teammates in Software Engineering (SE) 3.0: How Autonomous Coding Agents Are Reshaping Software Engineering. arXiv:2507.15003 [cs.SE] <https://arxiv.org/abs/2507.15003>
- [12] Yifei Ming, Zixuan Ke, Xuan-Phi Nguyen, Jiayu Wang, and Shafiq Joty. 2025. Helpful Agent Meets Deceptive Judge: Understanding Vulnerabilities in Agentic Workflows. arXiv:2506.03332 [cs.AI] <https://arxiv.org/abs/2506.03332>
- [13] Sunday Oladele and Faruk Lawal. 2025. The Impact of AI-Assisted Code Generation on Software Vulnerabilities and the Role of AI in Automated Security Testing. Available at SSRN 5253508 (2025).
- [14] Nivishree Palvannan and Chris Brown. 2023. Suggestion Bot: Analyzing the Impact of Automated Suggested Changes on Code Reviews. arXiv:2305.06328 [cs.SE] <https://arxiv.org/abs/2305.06328>
- [15] Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. 2021. Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions. arXiv:2108.09293 [cs.CR] <https://arxiv.org/abs/2108.09293>
- [16] Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirel. 2023. The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. arXiv:2302.06590 [cs.SE] <https://arxiv.org/abs/2302.06590>
- [17] Kexin Sun, Hongyu Kuang, Sebastian Baltes, Xin Zhou, He Zhang, Xiaoxing Ma, Guoping Rong, Dong Shao, and Christoph Treude. 2025. Does AI Code Review Lead to Code Changes? A Case Study of GitHub Actions. arXiv:2508.18771 [cs.SE] <https://arxiv.org/abs/2508.18771>
- [18] Dong Wang, Yuki Ueda, Raula Gaikovina Kula, Takashi Ishio, and Kenichi Matsumoto. 2021. Can We Benchmark Code Review Studies? A Systematic Mapping Study of Methodology, Dataset, and Metric. arXiv:1911.08816 [cs.SE] <https://arxiv.org/abs/1911.08816>
- [19] Miku Watanabe, Hao Li, Yutaro Kashiwa, Brittany Reid, Hajimu Iida, and Ahmed E. Hassan. 2025. On the Use of Agentic Coding: An Empirical Study of Pull Requests on GitHub. arXiv:2509.14745 [cs.SE] <https://arxiv.org/abs/2509.14745>