When AI Joins the Team: Understanding Human-Agent Collaboration in Pull Requests

Miguel Romero-Arjona
SCORE Lab, I3US Institute
Universidad de Sevilla
Seville, Spain
mrarjona@us.es

Saman Barakat
SCORE Lab, I3US Institute
Universidad de Sevilla
Seville, Spain
salias@us.es

Alberto Martin-Lopez
SCORE Lab, I3US Institute
Universidad de Sevilla
Seville, Spain
alberto.martin@us.es

Ana B. Sánchez
SCORE Lab, I3US Institute
Universidad de Sevilla
Seville, Spain
anabsanchez@us.es

Gabriele Bavota
SEART @ Software Institute
Università della Svizzera italiana
Lugano, Switzerland
gabriele.bavota@usi.ch

Sergio Segura
SCORE Lab, I3US Institute
Universidad de Sevilla
Seville, Spain
sergiosegura@us.es

***Abstract*—Autonomous AI agents are increasingly being used to assist software developers in tasks such as code generation and refactoring. However, we still lack a clear understanding of how these agents interact with humans in collaborative workflows such as GitHub pull requests (PRs). A helping hand to study this phenomenon comes from the AIDev dataset featuring 932k PRs authored (i.e., opened) by AI agents across 116k GitHub repositories. Nevertheless, such a dataset provides a limited view of the collaborations between humans and agents, since the latter may also be involved in PRs opened by humans. In this work, we analyse interaction patterns between humans and AI agents in PRs. To support this analysis, we introduce a novel methodology that extends the AIDev dataset to capture all PRs involving agent participation, irrespective of whether the agents initiated the PR. We apply such a methodology to a subset of 383 repositories from AIDev, finding 6,445 PRs, out of which 5,118 are agent-relevant (*vs* the 3,197 of AIDev on the same repositories) for a total of 74,525 interactions from 671 unique users. We classify these PRs by authorship (i.e., agent, bot, or human) and examine which participant types perform key interactions (e.g., reviews). Our findings indicate that human-agent collaboration is the most common interaction pattern, with about nine out of ten changes being merged. Although agents author a substantial proportion of PRs, fully autonomous contributions remain uncommon and are rarely accepted. These results highlight the increasing importance of AI agents in augmenting human capabilities in software development, as well as the central role of human oversight in ensuring successful contributions.**

***Index Terms*—AI agents, pull requests, human-agent collaboration**

I. Introduction

Autonomous AI agents, such as GitHub Copilot and OpenAI Codex, are increasingly being integrated into software development. These tools assist developers across a wide range of tasks, such as code generation and refactoring [1]–[3], leading to gains in productivity [4] and code quality [5], among other benefits. However, despite their advantages, their adoption raises important questions about how they interact with humans in practice. Addressing these questions is critical for understanding their ability to collaborate effectively with human developers in real-world environments.

Recent studies have started to explore how AI agents contribute to open-source software development [5]–[7]. In particular, the AIDev dataset [6], [8] provides the first large-scale view of autonomous agent activity on GitHub, identifying 932k pull requests (PRs) authored by five specific agents across 116k repositories. However, focusing solely on authorship provides an incomplete picture of agent involvement, as agents may also contribute to human-authored PRs, for example, by creating commits, leaving comments, or conducting reviews. As a result, some cases of human-agent collaboration remain undetected.

In this work, we present an analysis of interaction patterns between human and AI agents in PRs. To account for all types of interactions, and not only those in which an agent initiates the PR, we introduce a novel methodology that extends AIDev to capture all PRs in which agents participate. Our methodology complements the detection heuristics of AIDev by (i) uncovering previously undetected agent-related patterns, (ii) extending agent detection beyond the five agents initially targeted in AIDev, and (iii) including traditional bots. We applied this methodology to 383 repositories from AIDev, resulting in the UNITE (hUman-agent iNteractions In Teamwork Environments) dataset. This augmented dataset comprises 6,445 PRs (1,954 agent-relevant PRs not included in AIDev) involving 671 unique users and 74,525 interactions. To guide our investigation, we aim to answer the following research question:

RQ: *What are the main interaction patterns between humans, AI agents, and bots in pull requests?*

Our analysis reveals that mixed-participant PRs are the dominant form of collaboration. Specifically, 82.8% of PRs involved more than one type of participant, with human-agent collaboration being the most common pattern (44.9%). Although agents authored the majority of PRs in our dataset (62.1%), fully autonomous agent activity was rare, occurring in 3% of cases. We further observed differences in PR completion. Human-only PRs achieved the highest merge rate (92.2%), followed by those involving humans and agents (87.6%). However, agent-only PRs almost never progressed, remaining open in 99% of cases. These findings highlight the increasingly relevant role of AI agents in software development, with human oversight and guidance remaining central for successful contributions.

II. METHODOLOGY

The AIDev dataset [6], [8] features 932k PRs authored by five AI agents—Codex, Copilot, Cursor, Devin, and Claude—collected from 116k public repositories via the GitHub API. To identify agent-authored PRs, the authors used three heuristics: (i) author names associated with specific agents (e.g., devin-ai-integration[bot]); (ii) agent-related branch name conventions (e.g., `codex/<TEXT>`); and (iii) agent-attribution phrases in PR bodies (e.g., "Co-Authored-By: Claude"). Figure 1 shows a partial schema of the AIDev dataset, where non-bold elements indicate the fields covered in the dataset. For every PR, AIDev includes metadata about the host repository and the user who created it. Moreover, for PRs in repositories with more than 100 stars, the dataset provides additional information, including comments, reviews, commits, linked issues, and inferred purposes (e.g., feat, fix).

Although the AIDev dataset reliably identifies PRs authored by five specific AI agents, it does not guarantee that the remaining PRs in the same repositories did not involve AI agents at all. Indeed: (i) other AI agents not matched by the defined heuristics may go unnoticed; and (ii) PRs may be authored by humans or by traditional bots (e.g., those in charge of updating repository dependencies) and still involve AI agents in the process (e.g., to review a human-written patch). To address these limitations, we present a methodology to extend AIDev to include all PRs in which agents have been involved, irrespective of whether the agents initiated the PR. This methodology has been applied to create UNITE (hUman-agent iNteractions In Teamwork Environments), a dataset featuring 6,445 PRs from 383 AIDev's repositories, out of which 5,118 are agent-relevant (*vs* the 3,197 of AIDev on the same repositories), for a total of 74,525 interactions from 671 unique users. We only focus on a subset of the AIDev's repositories due to the high manual effort required by our methodology. Indeed, our goal is to maximise data completeness and quality, also highlighting the amount of data that may be missed via a fully automated process like the one used to build AIDev. Still, the 383 randomly selected repositories ensure a 95%±5% confidence level given the 108k repositories in AIDev that were still accessible in November 2025.

For each selected repository, we used the GitHub API to retrieve all PRs opened between 24 December 2024 and 31 July 2025 (same collection window used in AIDev). For every PR, we collected all participating users along with their interactions (e.g., comments, commits) and identification data, such as user ID, username, name, and email. The user ID was used for preventing duplicate entries in the `user` table shown in Figure 1, and for correctly linking the involved users to the PRs (`interaction` table). However, there were scenarios in which this information was unavailable. First, for commits having multiple authors the API only returns the first author, while the remaining ones appear in the commit body with the format `Co-authored-by: NAME <EMAIL>`. In this case, we used regular expressions to extract co-author details from the commit bodies. Second, for commits from authors without an associated GitHub account, the API returns their name and email as configured in their local git installation. Thus, in both scenarios, we were able to collect the authors' names and emails, but not their GitHub accounts. This process led to a list of 806 users, some featuring the authors' GitHub accounts and others only their names and emails. Thus, duplicates may appear when entries containing only name and email correspond to users whose GitHub accounts we were able to retrieve. To address this issue, we attempted to resolve each email to its corresponding GitHub account, either through an API search query or by matching it against previously collected users. We then removed duplicates by checking whether the resolved GitHub user ID matched an already-identified user. However, some duplicates remained because certain developers used multiple identities with no direct linkage between them (e.g., committing using different emails [9]). To handle these cases, we used gambit [10], a tool that applies similarity metrics to detect and merge records likely belonging to the same individual. After these steps, we reduced the number of users from 806 to 671.

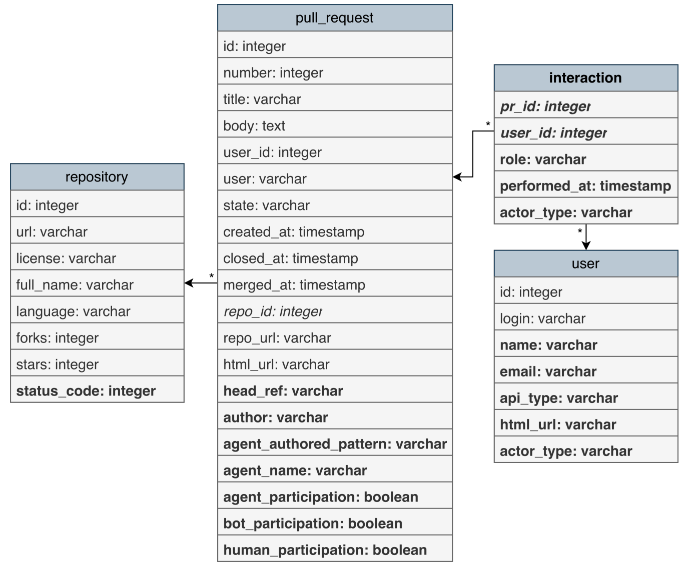


Fig. 1: UNITE dataset. New attributes in bold.

Each user was classified into one of three categories: agent, bot, or human. While the GitHub API provides a `user_type` property indicating bot-related accounts, this property is neither fully reliable for bot identification nor sufficient for distinguishing agents from traditional bots. To identify agents, we compiled a list of 184 potential agent usernames based on AI-assisted applications published in the GitHub Marketplace [11]. Users not appearing in this list were classified as bots if their API type was "Bot" or if their username contained "[bot]"; otherwise, they were classified as human.

The next step of our methodology targets the classification of the PRs as authored by agents, bots, or humans, with the last two not being part of AIDev. To identify agent-authored PRs, we adopt the same heuristics used in the building of AIDev, but extended to foster completeness of the collected data. First, we check whether the user opening the PR was classified as an "agent" in our previous step. If yes, the PR is classified as agent-authored, otherwise we look at the branch name of the PR's head reference. In particular, we manually inspected all branch names of the subject systems containing the name of an agent and, based on this analysis, we defined a list of regexes to match agent-created branches (scripts with regexes available in our replication package [12]). Such a list is more comprehensive as compared to the one used for AIDev (i.e., six branch names matched in our methodology *vs* three in AIDev).

If the PR can be linked to an agent-created branch, we classify it as agent-authored, otherwise we move to the third step. We inspected PR titles and bodies containing agent names and defined a list of lexical patterns likely to indicate the agent as the PR author. These patterns, available in our scripts [12], feature 71 trigger words (e.g., "generated", "created") followed by one of six possible attribution prepositions (e.g., "by", "with") and by the agent name. PRs matching any of these patterns were classified as agent-authored. Also in this case, our methodology substantially extends what was done for AIDev, in which a similar pattern matching has only been done for Claude Code. All unmatched PRs were finally classified according to their author type (bot or human).

In total, we confirmed 3,154 agent-authored PRs previously identified in AIDev and found 847 additional ones that have been missed in the building of AIDev. Among these newly identified PRs, 279 actually met the heuristics of AIDev but were missing from the dataset. Specifically, 181 used the branch name `codex/<TEXT>`, 70 used `copilot/<TEXT>`, and 28 used `cursor/<TEXT>`. Beyond these, we uncovered new patterns, most notably 553 PRs using the branch name `<TEXT>-codex/<TEXT>`, which is used by Codex [13]. We also found eight PRs authored by agents not covered in AIDev (i.e., amazon-q-developer [14] and coderabbitai [15]) and seven PRs using `devin/<TEXT>` branch name [16]. Conversely, 42 PRs originally identified as agent-authored by AIDev could not be included because they were no longer available at collection time, and one PR, although matching the `cursor/<TEXT>` branch name, showed no evidence of agent involvement and was therefore excluded [17].

For each PR, we also recorded whether agents, bots, or humans participated in supporting roles. For this, we checked whether users previously categorised as agent, bot, or human appeared in any of the supporting roles (e.g., committer). Additionally, our previous PR-body analysis revealed eight sentences indicative of agent involvement, which we used to detect cases where agents supported a PR even without directly participating through an agent account (e.g., a PR mentioning that a flaky test was fixed "with Claude" [18]). Finally, Codex-authored PRs required special handling because users prompt Codex through an interface with the agent carrying out actions using the user's GitHub account. In this case, we attributed all PR actions of an agent-authored PR to the agent, except those not supported by the agent (i.e., merge and close) or, obviously, performed by another user.

Table I summarises AIDev and UNITE, including the distribution of PRs, users, and interactions for the 383 sampled repositories. UNITE is publicly available on Hugging Face [19] and as part of our replication package [12].

TABLE I: Overview of the UNITE dataset. Values in parentheses indicate AIDev counts.

Table	Agents	Bots	Humans	Total
pull_request	4,001 (3,197)	593	1,851	6,445
user	19	18	634	671 (331)
interaction	24,352	8,799	41,374	74,525

III. Analysis and Findings

Figure 2 shows the distribution of PRs by author and participant types. Agents were responsible for more than half of PRs (62.1%), while humans created 28.7% and bots 9.2%. It is worth recalling that our sample was constructed from repositories containing at least one agent-created PR, and this may not generalise to PRs in general. Among agents, Codex was the most popular (87.4% of agent-authored PRs), followed by Copilot (7.6%) and Cursor (3.3%). The remaining agents (Devin, Amazon Q Developer, Claude, and CodeRabbit) collectively represented $\sim$2% of PRs. Bot authorship was largely driven by dependabot (79.2% of bot-authored PRs), followed by renovate (19.1%). The rest of the bots (architectbot, github-actions, and taylorbot) together accounted for $\sim$2% of PRs.

The majority of PRs (82.8%) involved multiple participant types, showing that mixed interactions predominated over isolated contributions. Human-agent collaboration was the most common pattern (44.9%), followed by interactions involving humans, agents, and bots (30.1%). Overall, patterns that included both humans and agents, regardless of bot participation, accounted for 75% of all PRs. In contrast, PRs with only human participation represented 11.1% of the total, suggesting that purely human-driven PRs are becoming less common in repositories where agents are actively used. Although agents authored the majority of PRs, fully autonomous activities are uncommon: 7.4% of PRs did not involve humans, and just 3% were agent-only. This highlights the crucial role of developer oversight and guidance for completing PRs.

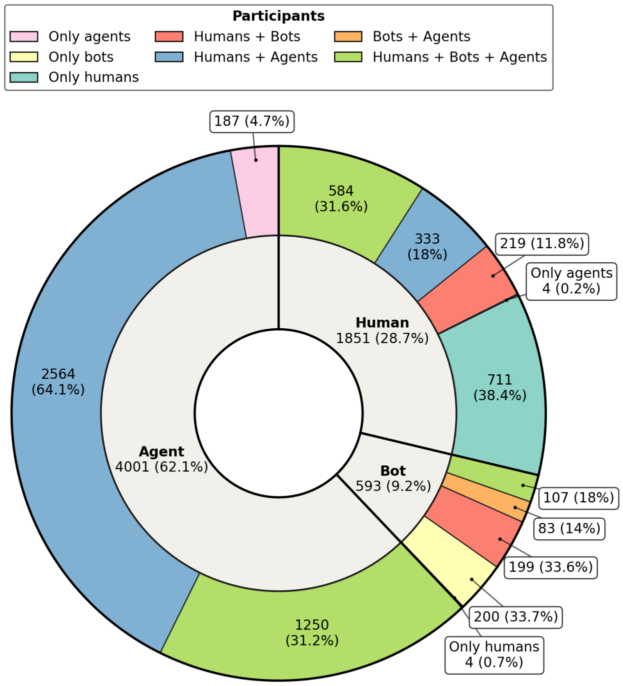
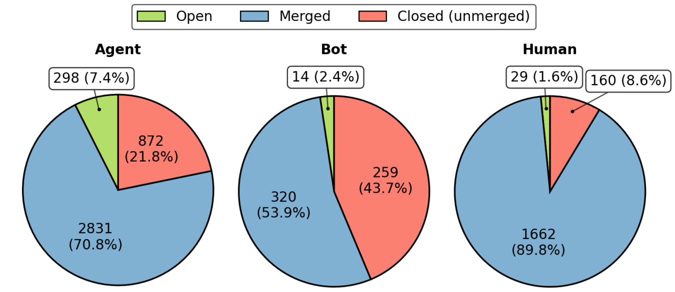


Fig. 2: Distribution of PRs by author (inner ring) and participation types (outer ring). Dataset size: 6445.

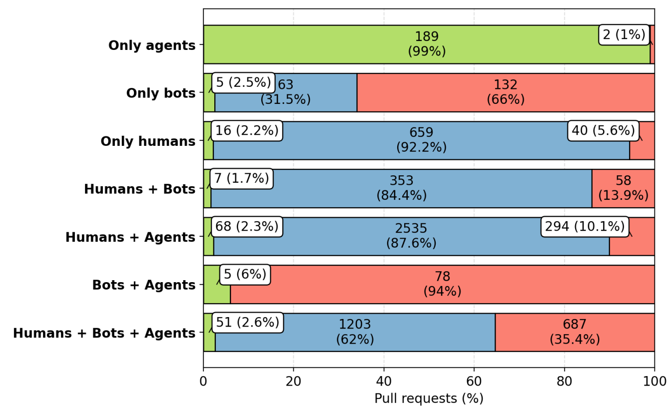
> 💡 Human-agent interactions (occasionally involving bots) emerge as the dominant collaboration pattern, accounting for 75% of the PRs analysed. In contrast, agent-only PRs represent a small minority (3%), underscoring the role of agents in complementing, rather than replacing, human developers.

Figure 3 shows the distribution of opened, merged and closed PRs across author and participant types. Completion rates (Figure 3a) varied depending on who created the PR. Human-authored PRs achieved the highest merge rate (89.8%), followed by those created by agents (70.8%) and bots (53.9%). Conversely, bot-authored PRs had the highest closure (unmerged) rate (43.7%), compared to 21.8% for agent-authored PRs and 8.6% for those initiated by humans.

Participant type patterns (Figure 3b) exhibited similar trends. PRs involving only humans achieved the highest merge rate (92.2%), suggesting that fully human-driven interactions remain the most effective for completing PRs. Similarly, PRs involving both humans and agents also showed high completion, with an 87.6% merge rate. In contrast, PRs handled exclusively by agents rarely progressed and remained open in 99% of cases, likely because they were created primarily for testing purposes or awaited intervention from non-agent contributors. Overall, patterns involving human participation are associated with the highest PR completion rates: 79.6% of PRs with at least one human participant were merged, while 18.1% were closed without merging.



(a) Per author.



(b) Per participant.

Fig. 3: Distribution of PRs by state. Dataset size: 6,445.

> 💡 PRs involving human-agent interaction achieve high merge rates (87.6%), whereas agent-only PRs are rare and largely unsuccessful (99% remain open), confirming the critical role of human developers in supervising agent activity to ensure successful contributions.

IV. Related Work

Several studies have explored how AI systems support humans in software development. Tufano et al. [20] analysed 1,501 commits, issues, and PRs where developers mentioned ChatGPT usage, resulting in a taxonomy of 45 automated tasks. Xiao et al. [21] introduced DevGPT, a dataset of 29,778 developer-ChatGPT conversations linked to GitHub and Hacker News artefacts (e.g., commits), enabling research on refactoring [22], [23] and code review [24].

Li et al. [6] introduced AIDev [8], the first large-scale dataset capturing agent behaviour on GitHub. Horikawa et al. [5] used AIDev to analyse 12,256 refactoring PRs, finding that agents focus on simple, localised improvements. Watanabe et al. [7] examined 567 PRs driven by Claude Code, reporting comparable revision effort to human-authored PRs. Coutinho et al. [25], through surveys and communication analysis at a software company, reported developer concerns about AI output reliability. Similarly, Kumar et al. [26] observed developers using Cursor, finding that iterative, interactive human-agent collaboration often outperforms full task delegation.

Most studies examining agent involvement in PRs focus on a limited set of agents and primarily consider only their role as authors. In contrast, our work analyses interaction patterns among all PR participants (humans, AI agents, and bots) and introduces detection patterns to identify agent participation beyond predefined subsets.

V. Threats to Validity

Although we compiled a list of 184 potential agent usernames and used heuristics to classify GitHub users as agents, bots, or humans, some misclassifications may have occurred. To minimise this threat, two authors independently reviewed all 671 classifications, achieving a Cohen's kappa of 0.89, which indicates almost perfect agreement [27]. Discrepancies were resolved through discussion to establish a consensus ground truth. Our data collection window (i.e., 24 December 2024-31 July 2025) also introduces limitations. Given the rapid evolution of agent capabilities and adoption [6], findings may not generalise to future periods. Moreover, our analysis focused on a random sample of 383 repositories that included at least one agent-authored PR. Therefore, the results may not extend to repositories with no PRs created by agents. These repositories also varied in size, programming language, and domain, which may influence agent adoption and human-agent collaboration dynamics.

VI. Conclusions

In this paper, we analyse interaction patterns between human and AI agents in GitHub PRs. To capture all types of interactions, we introduce a novel methodology that extends AIDev to identify all PRs involving agent participation. Our results show that the majority of PRs feature collaboration between humans and AI agents, with these collaborative efforts frequently resulting in merges. However, although agents contribute significantly to PRs, they rarely succeed when operating in isolation. These results highlight that while AI agents play a major role in software development, their effectiveness remains highly dependent on human oversight and intervention. Potential lines of future work include exploring how authorship and participation patterns shape PR dynamics (e.g., review workload, discussion length) and analysing the factors that influence agent adoption across repositories and development teams.

Acknowledgment

This work is a result of grant TED2021-131023B-C21, funded by MCIN/AEI/10.13039/501100011033 and by European Union "NextGenerationEU/PRTR"; grant DGP_PRED_2024_00262, funded by the Junta de Andalucía/CIIU and the FSE+; and project PID2024-156482NB-I00, funded by MICIU/AEI/10.13039/501100011033 and by ERDF/EU. This work is also supported by the Spanish Ministry of Science and Innovation under the Excellence Network AI4Software (Red2022-134647-T).

References

[1] W. Takerngsaksiri, J. Pasuksmit, P. Thongtanunam, C. Tantithamthavorn, R. Zhang, F. Jiang, J. Li, E. Cook, K. Chen, and M. Wu, "Human-In-The-Loop Software Development Agents," in *2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2025, pp. 342–352.

[2] M. H. Nguyen, T. Phan Chau, P. X. Nguyen, and N. D. Q. Bui, "AgileCoder: Dynamic Collaborative Agents for Software Development based on Agile Methodology," in *2025 IEEE/ACM Second International Conference on AI Foundation Models and Software Engineering (Forge)*, 2025, pp. 156–167.

[3] F. Batole, D. OBrien, T. N. Nguyen, R. Dyer, and H. Rajan, "An LLM-Based Agent-Oriented Approach for Automated Code Design Issue Localization," in *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, 2025, pp. 1320–1332.

[4] J. D. Weisz, S. V. Kumar, M. Muller, K.-E. Browne, A. Goldberg, K. E. Heintze, and S. Bajpai, "Examining the Use and Impact of an AI Code Assistant on Developer Productivity and Experience in the Enterprise," in *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '25, 2025, pp. 1–13.

[5] K. Horikawa, H. Li, Y. Kashiwa, B. Adams, H. Iida, and A. E. Hassan, "Agentic Refactoring: An Empirical Study of AI Coding Agents," *arXiv preprint arXiv:2511.04824*, 2025.

[6] H. Li, H. Zhang, and A. E. Hassan, "The Rise of AI Teammates in Software Engineering (SE) 3.0: How Autonomous Coding Agents Are Reshaping Software Engineering," *arXiv preprint arXiv:2507.15003*, 2025.

[7] M. Watanabe, H. Li, Y. Kashiwa, B. Reid, H. Iida, and A. E. Hassan, "On the Use of Agentic Coding: An Empirical Study of Pull Requests on GitHub," *ACM Trans. Softw. Eng. Methodol.*, 2026.

[8] "AIDev dataset," https://huggingface.co/datasets/hao-li/AIDev., 2025, accessed February 2026.

[9] 2025. [Online]. Available: https://github.com/The-Mcorp/tca-testing-tasks/pull/3/commits

[10] C. Gote and C. Zingg, "gambit – An Open Source Name Disambiguation Tool for Version Control Systems," in *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*, 2021, pp. 80–84.

[11] "GitHub Marketplace: AI Assisted Apps," https://github.com/marketplace?type=apps&category=ai-assisted., 2025, accessed February 2026.

[12] M. Romero-Arjona, S. Barakat, A. Martin-Lopez, A. B. Sánchez, G. Bavota, and S. Segura, "[Replication package] When AI Joins the Team: Understanding Human-Agent Collaboration in Pull Requests," 2026. [Online]. Available: https://doi.org/10.5281/zenodo.20284115

[13] 2025. [Online]. Available: https://github.com/hyndex/libslac/pull/128

[14] 2025. [Online]. Available: https://github.com/genaforvena/watching_u_watching/pull/45

[15] 2025. [Online]. Available: https://github.com/genaforvena/watching_u_watching/pull/32

[16] 2025. [Online]. Available: https://github.com/ryogrid/web-photo-frame/pull/5

[17] 2025. [Online]. Available: https://github.com/graydelay/spring-cloud-user-service/pull/11

[18] 2025. [Online]. Available: https://github.com/artsy/force/pull/15864

[19] "UNITE dataset," https://huggingface.co/datasets/interaction-patterns/UNITE., 2025, accessed April 2026.

[20] R. Tufano, A. Mastropaolo, F. Pepe, O. Dabic, M. Di Penta, and G. Bavota, "Unveiling ChatGPT's Usage in Open Source Projects: A Mining-based Study," in *2024 IEEE/ACM 21st International Conference on Mining Software Repositories (MSR)*, 2024, p. 571–583.

[21] T. Xiao, C. Treude, H. Hata, and K. Matsumoto, "DevGPT: Studying Developer-ChatGPT Conversations," in *2024 IEEE/ACM 21st International Conference on Mining Software Repositories (MSR)*, 2024, p. 227–230.

[22] E. A. AlOmar, A. Venkatakrishnan, M. W. Mkaouer, C. Newman, and A. Ouni, "How to Refactor this Code? An Exploratory Study on Developer-ChatGPT Refactoring Conversations," in *2024 IEEE/ACM 21st International Conference on Mining Software Repositories (MSR)*, 2024, p. 202–206.

[23] O. S. Chavan, D. D. Hinge, S. S. Deo, Y. O. Wang, and M. W. Mkaouer, "Analyzing Developer-ChatGPT Conversations for Software Refactoring: An Exploratory Study," in *2024 IEEE/ACM 21st International Conference on Mining Software Repositories (MSR)*, 2024, p. 207–211.

[24] M. Watanabe, Y. Kashiwa, B. Lin, T. Hirao, K. Yamaguchi, and H. Iida, "On the Use of ChatGPT for Code Review: Do Developers Like Reviews By ChatGPT?" in *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '24, 2024, p. 375–380.

[25] M. Coutinho, L. Marques, A. Santos, M. Dahia, C. França, and R. de Souza Santos, "The Role of Generative AI in Software Development Productivity: A Pilot Case Study," in *Proceedings of the 1st ACM International Conference on AI-Powered Software*, ser. AIware 2024, 2024, p. 131–138.

[26] A. Kumar, Y. Bajpai, S. Gulwani, G. Soares, and E. Murphy-Hill, "Why AI Agents Still Need You: Findings from Developer-Agent Collaborations in the Wild," in *2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2025, pp. 432–444.

[27] J. Sim and C. C. Wright, "The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements," *Physical Therapy*, vol. 85, no. 3, pp. 257–268, 2005.