AI Code Generation: Quantifying How Much Help They Need

PHAN TRUONG PHUOC NGUYEN, University of British Columbia, Canada
SAGE YANG, University of British Columbia, Canada

AI coding agents are increasingly capable of opening and updating pull requests (PRs) in real GitHub repositories, yet it remains unclear how autonomous these “agentic” workflows truly are. Using the AIDev dataset, we analyze an enriched subset of 33,596 agent-generated PRs with detailed timelines, comments, reviews, and code changes. We find that fewer than 10% of PRs run end-to-end without human involvement, whereas more than 75% require at least one human-authored commit. Fully automated PRs tend to be smaller and more localized, while larger or more complex changes draw human oversight or direct edits. Review outcomes reinforces this pattern, automated PRs have the lowest acceptance rates, whereas PRs with human commits are merged most frequently and with the shortest turnaround times. These results show that current AI coding agents accelerate early development work but still rely heavily on human judgment to produce reliable, high quality code.

1 Introduction

With the rapid rise of AI coding agents, systems capable of writing, reviewing, and modifying code are increasingly integrated into real-world software development. For example, autonomous agents such as OpenAI Codex have produced hundreds of thousands of pull requests across public repositories [2, 3]. In an ideal workflow, a developer would simply request a feature or report a bug, and an autonomous agent would implement the change, review its own work, and merge the pull request (PR) without human oversight.

However, in practice, human intervention remains essential. Current agents often lack full codebase context, leading to incomplete or error-prone changes. Developers frequently step in to provide feedback, request modifications, or fix issues directly before merging.

To study these dynamics, we use **AIDev**, a large-scale dataset of agent-generated pull requests (Agentic-PRs) from real GitHub repositories. AIDev contains **932,791 PRs** produced by five major agents—OpenAI Codex, Devin, GitHub Copilot, Cursor, and Claude Code—across more than 100,000 repositories and 70,000 developers. The dataset also provides an enriched subset of **33,596 PRs** containing review comments, commit diffs, event timelines, and linked issues. We focus our analyses on this enriched subset to better characterize the nature of human–agent collaboration.

Our goal is to examine **how autonomous current coding agents truly are**, and to what extent human developers remain part of the PR lifecycle. We investigate the following research questions:

- **RQ1 — Automation:** To what extent are PRs fully automated (i.e., all commits authored by bots with no human comments or reviews), and how often do they require human involvement?
- **RQ2 — Work Type and Complexity:** What types of tasks and complexity levels are associated with fully automated PRs versus those that involve human feedback or intervention?
- **RQ3 — Differences in Outcomes Between Automated and Human-Interacted PRs:** How do outcomes vary across pull requests that are fully automated compared to those that require human code contributions, comments, or feedback?

2 Dataset

We use the AIDev dataset, which provides a large collection of agent-generated pull requests from real GitHub repositories. The dataset contains several linked tables, including `pull_request`,

Authors’ Contact Information: PHAN TRUONG PHUOC NGUYEN, University of British Columbia, Canada, nguy3492@student.ubc.ca; SAGE YANG, University of British Columbia, Canada, ---.

pr_comments, pr_reviews, pr_commits, and pr_commit_details—each keyed by pr_id. These tables capture metadata, discussion threads, review events, commit authorship, and file-level code changes. A detailed description of all tables is available in the dataset documentation.[1]

3 Methods

We analyze PR-level automation behavior by merging AIDev tables using pr_id as the primary linking key. Our methodology is structured around the three research questions.

3.1 RQ1: Automation Extent of PRs

To quantify human participation and automation within pull requests, we combine information from three AIDev tables: pr_commits, pr_reviews, and pr_comments. Human involvement is detected when a PR contains at least one human-authored commit, one human review, or one human-authored comment.

3.1.1 Identifying bots.

The review and comment tables contain a user_type column that indicates whether an action was performed by a bot. Since pr_commits does not include such metadata, we construct a bot list by aggregating all users labeled as bots in the review and comment tables, and we additionally classify accounts that follow GitHub's standard [bot] suffix convention as bots.

3.1.2 Filtering administrative / non-code commits.

Timeline events often include administrative updates unrelated to code changes. To avoid overcounting such commits, we exclude commit identifiers associated with non-code event types such as closed, merged, reopened, auto_merge_disabled, auto_merge_enabled, labeled, unlabeled, milestoned, demilestoned, assigned, unassigned, subscribed, unsubscribed, mentioned, referenced, user_blocked, locked, and unlocked.

3.1.3 Constructing automation levels.

After removing bot-authored and non-code-related commits, we construct indicators of human participation at the PR level and assign each pull request an automation level:

- **Level 0:** No human involvement (bot-only PR).
- **Level 1:** Human comments or reviews are present, but no human-authored commits.
- **Level 2:** At least one human-authored commit.

We report the distribution of these three automation levels overall and by project in the full RQ1 analysis (outside the scope of this milestone).

3.2 RQ2: Work Type and Complexity

3.2.1 Dataset and Scope

For RQ2, a PR is our unit of analysis. We work with the enriched subset of AIDev for which we have interaction-level labels from RQ1. We exclude merge commits when computing commit statistics, since these commits are created by the platform and do not reflect new work by agents or humans.

3.2.2 Work-Type Labels

Each PR is assigned a work-type label derived from the conventional-commit style titles used by the agents. We group these into several high-level categories:

[1] See: https://huggingface.co/datasets/hao-li/AIDev/blob/main/data_table.md.

- feat (feature work),
- fix (bug fixes),
- docs (documentation),
- refactor,
- test,
- and an other bucket for less frequent categories.

We then compute, for each interaction level, the distribution of these work types across PRs.

3.2.3 PR Size and Complexity Measures

To approximate PR size and complexity, we aggregate commit-level metadata to the PR level and compute three metrics:

- **Number of commits** (n_commits): the count of non-merge commits associated with the PR.
- **Number of files touched** (n_files): the number of distinct files that are modified, added, or deleted.
- **Lines changed** (lines_changed): the total number of lines added plus lines deleted, or the provided changes field when available.

For each interaction level we summarize these metrics using medians and interquartile ranges and visualize their distributions with boxplots.

3.3 RQ3:Comparing Outcomes of Automated and Human-Involved PRs

We examine whether pull requests with different levels of human interaction (Levels 0–2 from RQ1) differ in two key outcomes: **acceptance rate** and **turnaround time** (time to close). As before, we analyze PRs from the enriched AIDev subset.

- **Starting point.** We reuse the interaction labels from RQ1 (Levels 0–2), where each PR is classified based on the presence of human comments, reviews, or commits.
- **Join with PR metadata.** We merge these interaction labels with the pull_request table using pr_id (or id) to obtain timestamps (created_at, closed_at, merged_at) and PR state (open, closed).
- **Scope.** We restrict analysis to **Agentic PRs**, i.e., PRs initiated by AI coding agents as identified in the AIDev dataset.
- **Derived variables.**
  - **Accepted** — a binary flag equal to 1 if merged_at is non-null (the PR was merged), and 0 otherwise.
  - **Turnaround time** — computed as the time difference between created_at and closed_at, converted to hours. We compute this only for PRs that were already closed at data collection.
- **Aggregation.** For each interaction level (0–2), we compute:
  - the number of PRs,
  - the acceptance rate (fraction with accepted = 1),
  - the median turnaround time (hours),
  - and the 25th–75th percentile range to summarize the spread of review latency.

These aggregated results form the basis for our RQ3 analysis.

4 Results

4.1 RQ1: Distribution of automation levels across PRs

After assigning automation levels, we obtain the following distribution:

Level 0 (bot-only)	3,269
Level 1 (human comments/reviews)	3,635
Level 2 (human commits)	26,692

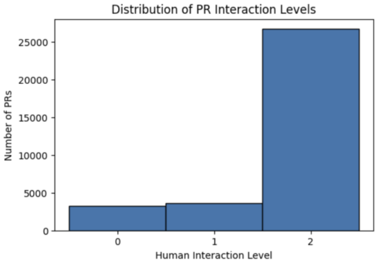

Fig. 1. Automation Levels of Pull Requests

Figure 1 suggests that fully automated workflows remain rare: only about 10% of PRs have no human interaction, another 10% require human feedback without human-authored commits, and over 75% still require at least one human-authored code contributions.

4.2 RQ2: Work Type and Complexity

4.2.1 Work Types Across Human-Interaction Levels

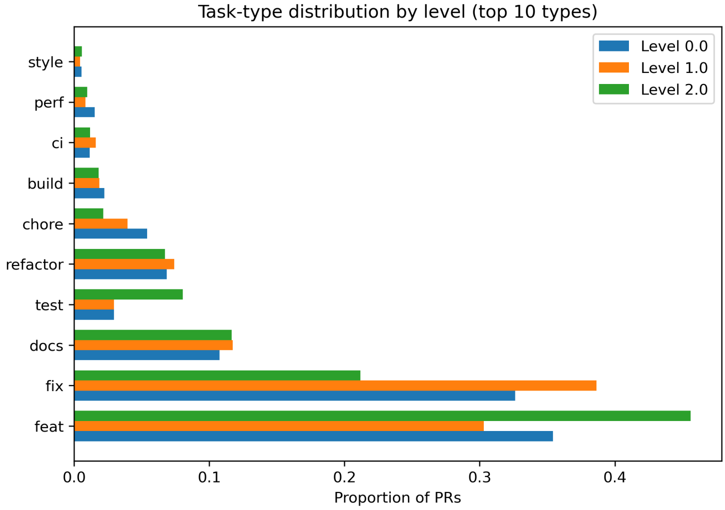

Fig. 2. Task-type distribution by human-interaction level for the most frequent work types.

Figure 2 compares the task-type distributions for the three interaction levels. Across all levels, feature and bug-fix work dominate, but their relative proportions change with the level of human involvement.

Fully automated PRs (level 0) are mainly `feat` and `fix`, which together make up roughly two thirds of PRs. Documentation and refactor work appear but are less frequent. For PRs with human review only (level 1), `fix` becomes more common, while `feat` drops slightly; documentation

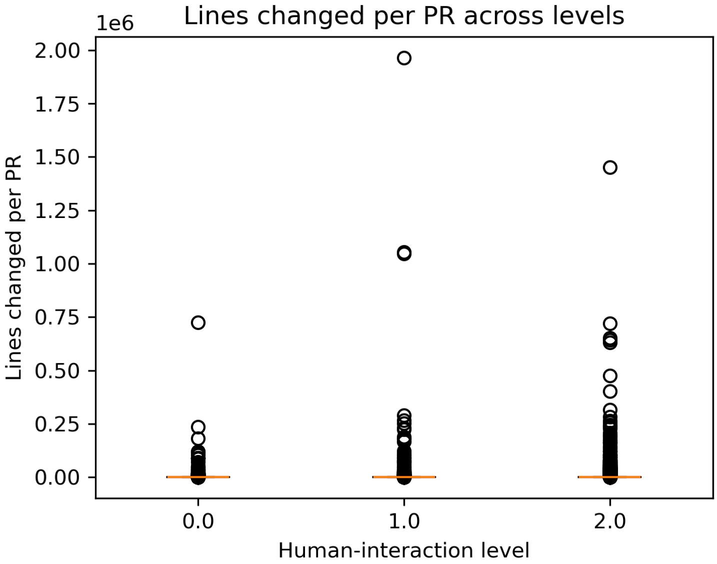

Fig. 3. Lines changed per PR across human-interaction levels.

and refactor work remain important. When humans add commits (level 2), `feat` becomes clearly dominant, whereas `fix`, `docs`, `test`, and `refactor` are present but less frequent. Overall, deeper human involvement is more strongly associated with feature development, while "review-only" involvement is relatively more skewed toward fixes and documentation.

4.2.2 Lines Changed per PR

Figure 3 shows the distribution of total lines changed (`lines_changed`) for each interaction level. Fully automated PRs are consistently smaller: they change tens of lines of code on median, with most PRs falling within a relatively narrow range. In contrast, PRs with human involvement tend to be larger and have a wider spread.

PRs with human review only (level 1) show the largest median number of changed lines, suggesting that reviewers are called in when the diff becomes more substantial. Level-2 PRs, where humans add commits, are also larger than fully automated PRs, but their median size is slightly lower than level 1, possibly because some level-2 PRs bundle a targeted human fix into an otherwise agent-generated PR.

4.2.3 Number of Files Touched

Figure 4 compares the number of files touched per PR. Fully automated PRs typically modify a small number of files (around two at the median), indicating that agents are mainly used for localized changes. Both level-1 and level-2 PRs tend to touch more files, which is consistent with the idea that multi-file or cross-cutting changes are more likely to require human oversight or direct edits.

4.2.4 Number of Commits per PR

Figure 5 shows the number of commits per PR. Fully automated PRs have a median of about two commits, reflecting the way agents often split their work into a small sequence of updates. Level-1

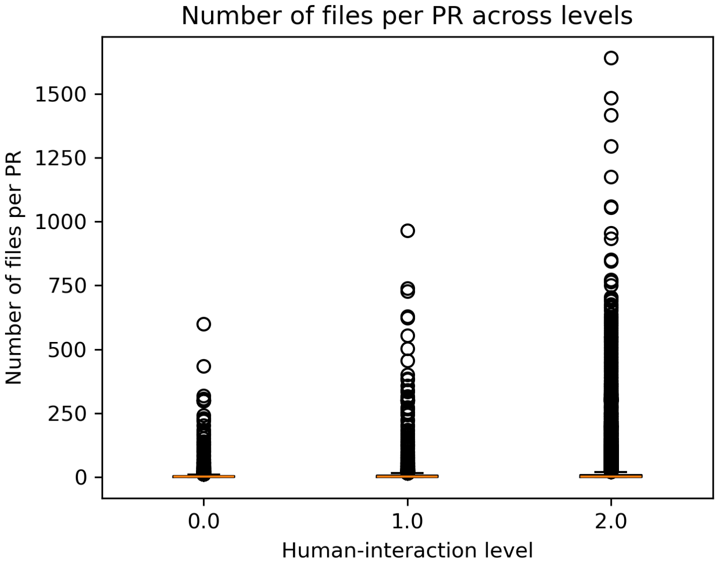

Fig. 4. Number of files touched per PR across human-interaction levels.

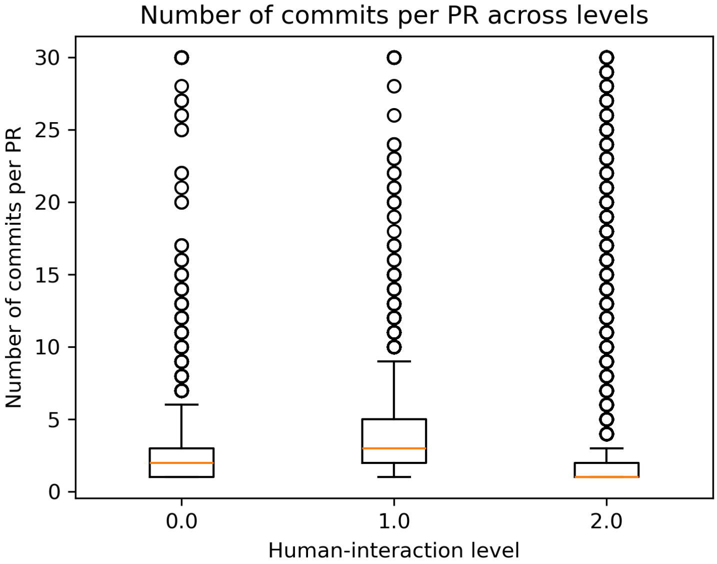

Fig. 5. Number of commits per PR across human-interaction levels.

PRs (with human review only) also tend to have multiple commits, which may represent iterative agent updates in response to review feedback.

Interestingly, level-2 PRs (with human commits) often have fewer commits overall, with a median around one. This pattern suggests that some human-involved PRs are used to apply a focused fix or adjustment on top of an agent-generated change, rather than going through many small iterations.

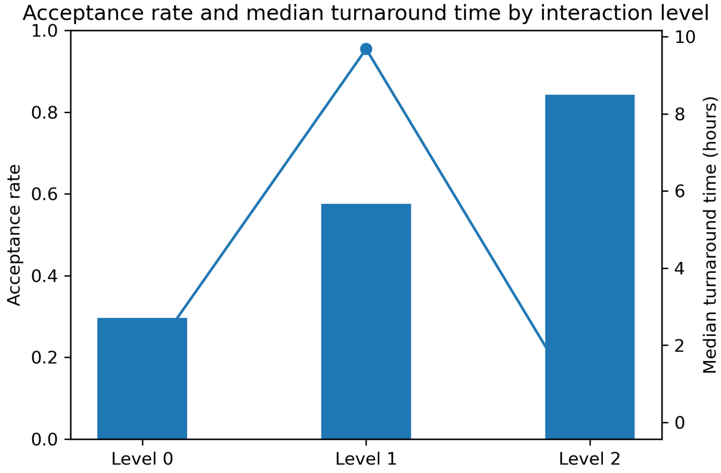

Fig. 6. Acceptance rate and median turnaround time across human-interaction levels. Fully automated PRs (level 0) have the lowest merge rate and a wide spread in turnaround time. PRs with human comments or reviews (level 1) are more likely to be merged but take the longest to process. PRs with human commits (level 2) combine the highest acceptance rate with the shortest median turnaround time, suggesting that deeper human involvement at the code level helps Agentic-PRs get integrated both faster and more reliably.

Across all three interaction levels, feature and bug-fix work dominate, but feature work is especially common in PRs where humans add commits. Fully automated PRs tend to be smaller and more localized: they change fewer lines, touch fewer files, and have modest numbers of commits. Larger or more complex PRs—those that change more lines or span more files—are increasingly likely to involve human review or direct human edits. These patterns suggest that current agentic systems are primarily used for well-scoped changes, while humans remain heavily involved in broader or riskier modifications.

4.3 RQ3: Review Outcomes Across Human-Interaction Levels

We now examine how review outcomes and efficiency vary across the three human-interaction levels. We focus on two metrics: (i) the acceptance rate (fraction of PRs that are merged) and (ii) turnaround time, measured as the time between PR creation and closure.

4.3.1 Acceptance rate

We observe clear differences in acceptance rates across human-interaction levels. Level-0 PRs (fully automated, $n = 2{,}736$) have an acceptance rate of only 29.6%. Level-1 PRs (with human comments or reviews but no human commits, $n = 3{,}136$) show a substantially higher acceptance rate of 57.5%. Level-2 PRs (with at least one human-authored commit, $n = 25{,}412$) reach the highest acceptance rate of 84.2%. Overall, the more human involvement a PR receives—especially at the commit level—the more likely it is to be merged.

4.3.2 Turnaround time

Turnaround times also differ substantially between interaction levels. We summarize them using the median and the interquartile range (IQR, from the 25th to the 75th percentile). Level-0 PRs have a median turnaround time of 1.41 hours (IQR 0.18–52.87 hours), indicating that many fully automated PRs are decided quickly, but some remain open or under review for much longer. Level-1

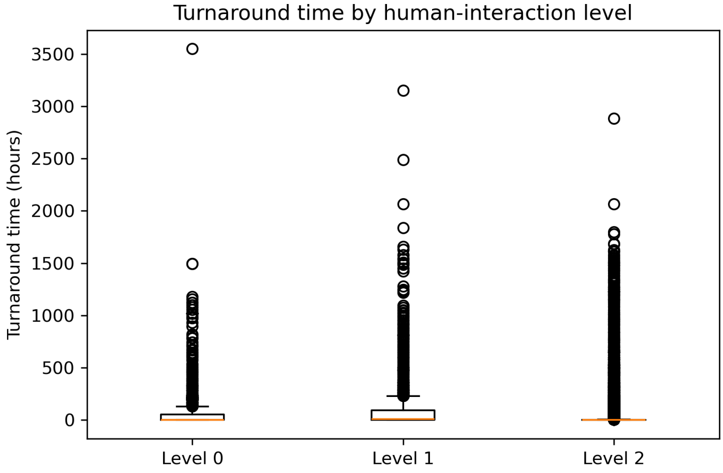

Fig. 7. Turnaround time (in hours) per pull request across human-interaction levels. Fully automated PRs (level 0) show a wide dispersion in turnaround time, with some PRs decided quickly and others lingering for days. PRs with human comments or reviews only (level 1) have the slowest typical processing, reflecting extended discussion cycles. PRs with human commits (level 2) cluster near very short turnaround times, indicating that once humans edit the code directly, decisions are usually made within minutes to a few hours.

PRs exhibit the slowest processing: the median turnaround time rises to 9.68 hours (IQR 0.97–93.12 hours), suggesting that PRs which attract human comments and reviews—but no human commits—tend to stay in review for almost a full working day or longer. Level-2 PRs are processed the fastest, with a median turnaround time of only 0.05 hours (approximately three minutes; IQR 0.01–1.80 hours). This indicates that PRs where humans actively contribute commits are typically decided very quickly, often within minutes to a couple of hours.

5 Discussion

5.1 RQ1: Extent of Human Participation

We find that fewer than 10% of PRs are fully automated, whereas over 75% still require at least one human-authored code contribution.This pattern is similar to findings from prior small-scale studies examining how developer collaborate with AI coding assistants [1, 4]. While tools such as Copilot and LLM-based generators often provide useful starting points and substantially reduce initial coding or search efforts, developers still spend significant time interpreting, validating, and debugging the generated code. Our results reinforce the view that while AI systems accelerate early-stage development, it does not remove the need for human judgement and oversight.

5.2 RQ2: Work Type and Complexity

Our RQ2 analysis indicates that teams are comfortable letting agents handle small, localized changes end-to-end, but they still rely on human oversight for larger or more complex work. Even when agents generate PRs for important work types such as features and fixes, humans frequently review or directly modify the proposed changes.

Several limitations apply to these findings. First, we rely on interaction-level labels and size metrics derived from metadata; these are proxies and may not fully capture semantic complexity. Second, work-type labels are based on commit titles and may misclassify some PRs. Third, we only study PRs that are part of the AIDev dataset; projects that opt into this dataset may use agents differently from other repositories. Finally, our analysis is cross-sectional and does not model how usage patterns evolve over time.

Despite these limitations, the consistent differences in work types and size distributions across interaction levels provide initial evidence that automation is concentrated on smaller changes, while humans remain central for larger and more complex PRs.

5.3 RQ3: Review Outcomes and Efficiency

The RQ3 results further highlight a trade-off between automation, review effort, and decision speed. Fully automated PRs (level 0) are not only less likely to be merged, but also show highly variable turnaround times, suggesting that some purely agentic changes linger in review or are closed without integration. PRs with human comments and reviews (level 1) improve acceptance rates but take the longest to process, reflecting extended discussion and negotiation.

In contrast, PRs with human commits (level 2) achieve both the highest acceptance rate and the shortest median turnaround time. One plausible interpretation is that when humans take ownership of AI-generated changes, by editing or extending them directly, the review process becomes more decisive and efficient. Instead of merely adding delay, human involvement at the code level appears to help converge quickly on an acceptable solution. From a practical perspective, our findings suggest that teams may benefit most from workflows where agents propose initial changes and humans refine them, rather than leaving agentic PRs to stand alone.

6 Conclusion

Our large-scale analysis of 33,596 agent-generated pull requests shows that current AI coding agents are far from fully autonomous. Fewer than 10% of PRs run end-to-end without human involvement, and over 75% require at least one human-authored commit. Agents are most effective for small, localized changes, while larger or more complex work consistently draws human oversight or direct edits. Review outcomes mirror this pattern: fully automated PRs have the lowest acceptance rates, whereas PRs with human commits are merged more often and much faster. Overall, AI agents accelerate early-stage work but still depend heavily on human judgment to produce reliable, high-quality contributions. Future work should examine how these collaboration patterns evolve as agent capabilities and development practices mature.

Team Roles and Contributions

Phan Truong Phuoc Nguyen. Developed the overall study design and formulated the research questions. Designed and implemented the RQ1 pipeline for labeling PRs by level of human interaction, and produced the interaction-level dataset used as input for RQ2 and RQ3. Drafted the introduction, methodology, and discussion sections for RQ1.

Sage Yang. Led the RQ2 and RQ3 analyses on work types, PR size, and review outcomes. Implemented the aggregation of task types, size metrics, and review metrics, produced the RQ2 visualizations, and drafted the RQ2 and RQ3 methodology, results, and discussion sections. Organized the structure of this milestone report.

GenAI Disclosure

We used ChatGPT (OpenAI) to assist with editing the text of this report and to suggest Python code patterns for data aggregation and plotting. All analyses were run by us, and all interpretations and final code were reviewed and validated by the authors.

Acknowledgments

We thank the AIDev dataset authors for collecting and releasing the data used in this study.

References

[1] H. Barke, J. Alber, and S. Baltes. 2023. Grounded Copilot: How Programmers Interact with Code-Generating Models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM.

[2] Hao Li. 2023. AIDev: Agentic Pull Requests Dataset. https://huggingface.co/datasets/hao-li/AIDev. Accessed 2025-01-01.

[3] Hao Li. 2023. The Rise of AI Teammates in Software Engineering (SE 3.0): How Autonomous Coding Agents Are Reshaping Software Engineering. arXiv preprint arXiv:2307.XXXX (2023).

[4] D. Vaithilingam and R. Miller. 2023. Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM.

[5] Hao Li, Haoxiang Zhang, and Ahmed E. Hassan. 2025. AIDev: Studying AI Coding Agents on GitHub. Preprint and dataset.