iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IX **Month of publication:** September 2026

DOI: https://doi.org/10.22214/ijraset.2026.84804

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IX Sep 2026- Available at www.ijraset.com

AI Coding Agents on GitHub: Repository Popularity as a Hidden Confounder in the Acceptance of 2.4 Million Agent-Authored Pull Requests

Perseus Bhavnagri
Department of Information Technology, Wilson College, University of Mumbai

Abstract: ***Autonomous coding agents now open pull requests on GitHub without a person writing the code, and the share of those pull requests that maintainers merge has become the usual shorthand for how well the agents work. This paper argues that the shorthand measures the projects as much as it measures the agents. Using the public AIDev dataset, a snapshot of 2,743,854 pull requests written by six agents across 326,798 repositories between December 2024 and October 2025, we find that the merge rate falls from 91.58 percent in repositories with no stars to 60.05 percent in repositories with a thousand or more. The decline appears inside every agent separately, and in the five agents with enough volume in the top band to measure it the fall is between 20.3 and 27.6 percentage points, so it cannot be an artefact of which agent is used where. The enriched portion of the dataset that previously published analyses rely on contains only repositories with at least 100 stars, which we verify directly; it holds 2.7 percent of the decided pull requests and merges them at 73.67 percent against 90.63 percent elsewhere. Holding the repository fixed with a Mantel-Haenszel estimator changes the picture between agents: of fifteen comparable pairs, two reverse direction and one loses more than half its effect.***

Time to a decision separates the same way, a median of 51 seconds in unstarred repositories against 5.4 hours in the most popular ones, with a medium effect size. Even within the enriched subset, 75.8 percent of agent pull requests carry no recorded human review. We conclude that published acceptance figures describe a narrow and unusually demanding slice of GitHub, and we recommend that studies of agent contributions report the popularity distribution of their repositories and estimate effects within repositories rather than across them.

Keywords: AI coding agents, pull request acceptance, GitHub, mining software repositories, confounding variables, Simpson's paradox, code review, generative AI, empirical software engineering, agentic software engineering.

I. INTRODUCTION

A pull request is the unit in which work is offered to a software project on GitHub. Somebody proposes a change, somebody else decides whether the project takes it, and the record of that decision is public. Until recently the person proposing the change was a human being. That has altered quickly. Coding agents built on large language models are now given a task in words and reply with a finished pull request, and they do so in volume: the dataset examined in this paper contains more than two and a half million such contributions produced in under a year.

The obvious question is whether the work is any good, and the research community has settled on an obvious proxy. If maintainers merge an agent's pull request, the work is counted a success; if they close it unmerged, a failure. The proportion merged is reported as an acceptance rate and read as a rough grade for the agent. Published figures sit broadly between sixty and ninety percent depending on the study, the agent and the kind of task.

This paper questions what that grade is actually measuring. Merging is not an act performed by the agent. It is a decision taken by a project, under that project's review conventions, its tolerance for risk, and its number of maintainers. A hobby repository with one owner and no audience is not making the same decision as a widely used library with a release process and a reputation to protect, even when the code in front of them is identical. If agents are not distributed evenly across those two kinds of project, and there is no reason they should be, then an acceptance rate pooled across repositories is partly a measurement of the repositories.

We test that concern on the AIDev dataset, the resource on which the empirical literature about agent pull requests currently rests. The contributions of this paper are as follows.

- We show that the merge rate of agent pull requests is strongly related to how popular the receiving repository is, falling from 91.58 percent in repositories with no stars to 60.05 percent in those with a thousand or more.
- We show the same decline holds inside each of the six agents separately, which rules out the explanation that different agents are simply used in different kinds of project.
- We verify directly that the enriched portion of AIDev, the portion prior studies analyse, consists entirely of repositories with at least 100 stars, and we quantify what that selection costs: it covers 2.7 percent of the decided agent pull requests and merges them at a rate nearly seventeen percentage points below the remainder.
- We re-estimate every pairwise comparison between agents while holding the repository fixed, and find that two of fifteen comparisons reverse direction and one more loses over half its magnitude.
- We show that the same divide appears in how long a decision takes, and that even in the most closely watched repositories in the dataset, three quarters of agent pull requests carry no recorded human review.

None of this says that agents are better or worse than has been reported. It says that the number being reported is not a property of the agent alone, and that studies which pool across repositories are measuring something other than what they intend.

II. REVIEW OF LITERATURE

A. *The dataset the field is built on*

Li, Zhang and Hassan assembled AIDev, a collection of pull requests opened by identifiable coding agents on public GitHub repositories, and released it openly [1]. It is the common foundation for nearly all of the empirical work discussed below. The collection has two tiers. A broad tier records the basic outcome of every agent pull request the authors could find. A narrower enriched tier adds reviews, comments, commits and linked issues, and is restricted to repositories above a popularity threshold. That structural detail turns out to matter a great deal, and Sections IV and VI return to it.

B. *How widely agents have been taken up*

Robbes, Matricon, Degueule, Hora and Zacchiroli surveyed adoption across 128,018 projects and put the share of projects showing agent involvement between roughly twenty-two and twenty-nine percent, which is a striking figure for tools that were months old at the time [2]. They also observed that commits produced with agent assistance tend to be larger than purely human ones. Ghaleb studied the traces agents leave behind and how they can be told apart from one another, which is the practical basis on which any study of this kind identifies its subjects [3].

C. *Acceptance rates as they have been reported*

Watanabe, Li, Kashiwa, Reid, Iida and Hassan examined 567 pull requests produced with one agent across 157 projects and reported that 83.8 percent were eventually merged, with slightly over half going in untouched [4]. Pinna, Gong, Williams and Sarro compared five agents over 7,156 pull requests and made the important point that the kind of task dominates the outcome: documentation work was accepted far more readily than new features, and the spread between task types exceeded the spread between agents for most categories [5]. They stratified their comparisons by task type for precisely that reason. Their paper also states plainly that repository-level clustering is not modelled in their tests and that repositories which accept or reject systematically could distort the result, and they name mixed-effects models as future work. The present paper takes up that thread, though with a different estimator.

Yoshioka and colleagues fitted regression models to 40,214 pull requests and compared human and agent submissions [6]. Their central result is that attributes of the submitter dominate the merge outcome for both groups. They also report that 77.51 percent of merged agent pull requests were merged by the same account that opened them, against 57.63 percent for human ones. That finding is important context for Section VI of this paper, and it means the question of self-merging is already settled in the literature rather than open.

D. *What review of agent work looks like*

Duma, Wróblewski, Bobińska, Winiarska and Przymus compared review activity on agent and human pull requests drawn from the same repositories and found that a large majority of agent submissions attract no human reviewer at all, and that where review does occur it is frequently conducted by other automated tools rather than people [7]. Their warning is that counting reviews tells a mining study very little about whether a person actually looked at the change.

Nachuma and Zibran modelled integration outcomes with repository-clustered standard errors and found reviewer engagement to be the strongest correlate of a pull request being merged, while large changes and disruptive actions such as force pushes worked against it [8]. Three further studies examine particular slices of agent behaviour. Siddiq, Zhao, Lopes, Casey and Santos isolated security-related agent pull requests and found them merged less often and reviewed more slowly than other agent work [9]. Abujadallah and colleagues looked specifically at why agent-generated fixes get rejected [10]. Sawada and colleagues asked what happens after the merge, and how much maintenance agent-written code subsequently demands [11].

E. Automation in pull requests before language models

Bots were opening pull requests long before agents existed, and that older literature is directly relevant because it separated the effects of automation from the effects of the code. Wessel and colleagues studied what happens to a project's pull request activity after a code review bot is introduced [12]. Wyrich and colleagues compared how contributors respond to automatically created pull requests against manually created ones [13]. Golzadeh and colleagues built and validated a method for telling bot accounts from human ones, together with a labelled dataset, which remains the standard reference for that classification problem [14]. A comparison between language-model agents and rule-based automation of this older kind would be a natural control, and Section IX explains why this study does not provide one.

F. What decides a pull request, independent of who wrote it

There is a long-standing body of work on pull request outcomes in general. Zhang and colleagues surveyed the factors that explain acceptance decisions [15]. Legay and colleagues studied the consequences of those decisions for whether a contributor returns [16]. Hasan and colleagues examined how long a pull request waits before anyone responds [17]. This literature has consistently found project-level characteristics to be among the strongest predictors of what happens to a contribution, which is the expectation this paper carries into the agent setting.

G. Whether the tools help

Evidence on productivity is genuinely mixed and worth stating as such. Peng and colleagues ran a controlled experiment and measured a substantial speed gain for developers working with an assistant [18]. Working against that optimistic reading, Xu and colleagues report that assisted programming slowed experienced developers down while accumulating technical debt [19]. Jimenez and colleagues built SWE-bench, the benchmark that made autonomous issue resolution measurable in the first place, and much of the present discussion of agent capability is anchored to it [20].

III. GAP ANALYSIS

Reading these studies together, three things stand out.

First, the empirical picture of agent pull requests rests almost entirely on one dataset, and specifically on its enriched tier. That tier exists because reviews and comments were only collected for repositories above a popularity threshold. The threshold was a sensible data-collection decision, but it has quietly become the sampling frame of a research area.

Second, the acceptance rate is treated as a measure of the agent, and reported as a single pooled number, while the general pull request literature has known for years that project characteristics drive acceptance strongly. Pinna and colleagues explicitly flag repository clustering as unaddressed in their own analysis [5]. Nachuma and Zibran adjust their standard errors for it [8], which correctly widens their confidence intervals but does not ask whether the point estimate itself is a repository effect in disguise.

Third, no published study we found quantifies how much the headline acceptance figure depends on the popularity of the repositories in the sample. That is the gap this paper addresses. It is a methodological question rather than a question about any particular agent, and answering it changes how the existing numbers should be read.

IV. OBJECTIVES, VARIABLES AND HYPOTHESES

A. Objectives

- To measure how the acceptance of agent-authored pull requests varies with the popularity of the receiving repository.
- To establish whether any such variation is a property of repositories or an artefact of different agents being used in different places.
- To determine how much the enriched subset used by prior work differs from the wider population of agent pull requests.
- To re-estimate comparisons between agents while holding the repository fixed, and report where conclusions change.
- To check whether the same pattern appears in a second, independent outcome, namely the time taken to reach a decision.

B. Variables

The dependent variables are whether a pull request was merged, which is binary, and the number of hours between opening and closing, which is continuous. The independent variable of interest is the star count of the repository, treated in ordered bands. The agent that authored the pull request is treated both as a variable of interest and as a control. Repository fork status and primary programming language are examined as additional candidate confounders.

C. Hypotheses

The hypotheses were written down before any test was run and are kept in Analysis/Hypotheses (pre-registered).md. Two of the questions had already been answered by others; they are stated here as replications rather than as discoveries, which is how the paper reports them. The list below is what was actually tested. It is not identical to the pre-registered list, and Section IV-D sets out every place the two differ.

- R1 (replication). Acceptance rates differ between agents. Previously reported by Pinna, Gong, Williams and Sarro on 7,156 pull requests [5].
- R2 (replication). Agent pull requests are more often merged by the account that submitted them than human ones are. Previously reported by Yoshioka and colleagues [6]. Discussed in Section VI-F but not re-tested here, for the reason given there.
- H2. The pooled difference between agents changes materially, or reverses, once the comparison is made within repositories rather than across them. Null hypothesis: the within-repository estimate equals the pooled estimate.
- H3. Acceptance measured across the whole population of repositories differs materially from acceptance measured in the enriched, high-popularity subset. Null hypothesis: the two are the same.
- H5 (added after the pre-registration, and therefore exploratory). The time taken to decide a pull request differs by repository popularity. Null hypothesis: it does not.

D. Departures from the pre-registration

Four things changed between writing the hypotheses down and running the tests. All four are stated here rather than quietly absorbed, and all four are recorded in the pre-registration file itself, which carries an amendment log and a table mapping its labels to the ones used in this paper.

- H1 proposed comparing agent pull requests against those of rule-based bots such as Dependabot in the same repositories, to separate an effect of artificial intelligence from an effect of automation in general. AIDev contains no such bots, so the test needed a separate collection from the GitHub API, and the study was scoped to the existing dataset instead. H1 was withdrawn without being tested, and appears in Section X as the first item of future work.
- H2 was pre-registered as a comparison between pull requests written by agents and pull requests written by people. The broad tier of AIDev holds none of the latter, so no human baseline exists in this data. The hypothesis was narrowed to the same question asked between one agent and another. The mechanism under test, confounding by repository, is unchanged; the contrast is not, and the narrowing happened before the estimator was run rather than after seeing its output.
- A pre-registered hypothesis on review effort asked whether agent pull requests attract a different number of review events than comparable human ones in the same repository. It fails on the same absence and was not tested. What can be seen of review is reported descriptively in Section VI-G, and nothing is inferred from it.
- The hypothesis on time to decision, H5 here, was not pre-registered at all. It was added afterwards as a second and independent outcome against which the popularity effect could be checked. It is exploratory, it is labelled so, and it is given a new number rather than reusing one from the pre-registration.

One replication fell away for the same want of a human baseline: the pre-registration listed a comparison of acceptance rates between agent and human pull requests, already reported by Yoshioka and colleagues [6], and this study does not attempt it. Nothing was added, dropped or renamed after a result was seen.

V. RESEARCH METHODOLOGY

A. Research design

This is secondary, observational research carried out in September 2026 on data that was already public. No experiment was run and no person was contacted, so nothing here supports a causal claim. Where the paper says that one thing is associated with another, that is the whole of what is meant.

B. Data source

The AIDev dataset was downloaded from Hugging Face on 1 September 2026 under a Creative Commons Attribution 4.0 licence [1]. The snapshot analysed contains 2,743,854 pull requests authored by six coding agents across 326,798 repositories, created between 24 December 2024 and 24 October 2025. A separate enriched table holds 71,677 of those pull requests, drawn from 6,673 repositories, together with review records.

The dataset is maintained and grows over time, so the snapshot date is part of the result rather than a detail. The figures reported in the source publication describing AIDev are smaller than those above, because that paper describes an earlier state of the same resource. Anyone repeating this analysis should expect their counts to differ and should record their own download date. The raw files as downloaded are kept in the Dataset folder accompanying this paper.

C. Measures

A pull request that is still open has not yet been decided, so it cannot contribute to a merge rate. All merge-rate calculations therefore use closed pull requests only, and the merge rate is the number merged divided by the number merged plus the number closed without merging. Of the 2,743,854 pull requests in the snapshot, 292,402 were still open and are excluded on this basis, leaving 2,451,452 decided cases.

Repository popularity is measured by star count and grouped into five ordered bands: none, one to nine, ten to ninety-nine, one hundred to nine hundred and ninety-nine, and one thousand or more. Time to decision is the interval in hours between the opening timestamp and the closing timestamp.

D. Statistical methods

Proportions are compared with Pearson's chi-square test of independence without continuity correction. Because a large enough sample makes almost any difference statistically significant, every test is reported with an effect size as well as a p value, and the effect size is what the discussion relies on. Cramér's V is used for contingency tables larger than two by two, and the phi coefficient and the odds ratio for two-by-two comparisons. Odds ratio confidence intervals use the standard logarithmic approximation, with the Haldane-Anscombe correction applied when a cell is empty.

The central method of this paper is the Mantel-Haenszel common odds ratio, which combines evidence across strata while holding the stratifying variable fixed. Each repository is a stratum, so the estimator asks how two agents compare within the same repository and then pools those within-repository comparisons. Comparing it against the crude odds ratio, which ignores repositories entirely, shows how much of an apparent difference between agents is really a difference between the projects that use them. Confidence intervals use the Robins, Breslow and Greenland variance estimator. Only repositories containing pull requests from both agents in a comparison can contribute, and the number of such repositories is reported alongside every result. Where a comparison is described as losing half its magnitude, magnitude means the logarithm of the odds ratio, and the criterion is that the adjusted log odds ratio is less than half the crude one in absolute value.

Distributions of time to decision are compared with the Mann-Whitney U test. The effect size reported beside it is Cliff's delta, read on the conventional thresholds of 0.147, 0.33 and 0.474 for small, medium and large. A rate is reported for a subgroup only where at least 200 pull requests fall in it; smaller cells are suppressed rather than printed, because a proportion estimated on fewer cases carries an interval too wide to read. The implementation of every test used here is in Analysis/statlib.py, and the Mantel-Haenszel routine was checked against the published kidney-stone data of Charig and colleagues, a standard textbook example of Simpson's paradox, before being applied to the study data. The analysis ran in Python, with pandas handling the data and NumPy and SciPy the statistics.

VI. DATA ANALYSIS AND FINDINGS

A. Description of the data

Table I gives the composition of the snapshot by agent, together with the merge rate of each. The six agents differ greatly in volume: one of them accounts for roughly three quarters of all agent pull requests in the dataset, which is itself worth remembering when reading any pooled figure.

TABLE I
AGENT VOLUME AND MERGE RATE IN THE FULL SNAPSHOT

Agent	Decided PRs	Merged	Merge rate
OpenAI Codex	1,923,154	1,792,465	93.20%
GitHub Copilot	286,212	222,196	77.63%
Cursor	143,177	115,104	80.39%
Google Jules	43,600	38,691	88.74%
Devin	38,577	26,902	69.74%
Claude Code	16,732	14,698	87.84%
All agents	2,451,452	2,210,056	90.15%

Excludes 292,402 pull requests still open at the snapshot date.

B. R1: agents do differ from one another

A chi-square test of independence on agent against merge outcome is significant, with a chi-square statistic of 104,382.8 on five degrees of freedom and p below 0.0001 over 2,451,452 cases. Cramér's V is 0.2063, a small to moderate effect. This reproduces on a much larger sample the finding of Pinna and colleagues that agents are not interchangeable [5]. It is reported as a replication, not as a new result. The remainder of this paper asks what that difference actually consists of.

C. H3: acceptance depends strongly on repository popularity

Table II gives the merge rate by star band. The relationship is strong, ordered, and in the direction one would expect if review standards rise with visibility.

TABLE II
MERGE RATE BY REPOSITORY POPULARITY

Stars	Repositories	Decided PRs	Merge rate
0	266,130	1,830,803	91.58%
1 to 9	44,678	476,605	88.35%
10 to 99	9,138	69,509	81.00%
100 to 999	4,232	44,092	80.52%
1,000 or more	2,620	22,198	60.05%

Grouping at the threshold that defines the enriched subset makes the size of the problem plain. Repositories with at least 100 stars merge 73.67 percent of the agent pull requests they receive, against 90.63 percent everywhere else, a gap of 16.96 percentage points. The chi-square statistic is 20,922.2 with p below 0.0001, the odds ratio is 0.289 with a 95 percent confidence interval from 0.284 to 0.295, and phi is 0.0925. That popular slice holds 2.7 percent of the decided pull requests in the dataset.

We confirmed the composition of the enriched subset directly rather than relying on its documentation: all 6,673 of its repositories have at least 100 stars, the smallest has 101 and the median has 558.

The obvious alternative explanation is that different agents are popular in different kinds of project, so that the gradient reflects the agent mix rather than the repositories. Fig. 1 tests this by computing the same gradient inside each agent separately. It survives in every case, with a drop between the unstarred and the most popular band of 21.8 points for OpenAI Codex, 23.5 for GitHub Copilot, 27.6 for Cursor, 20.3 for Devin and 20.5 for Claude Code. Google Jules has too few pull requests in the highest band to report, 163 against a threshold of 200, and falls 16.6 points across the four bands where it does have the volume. The null hypothesis for H3 is rejected.

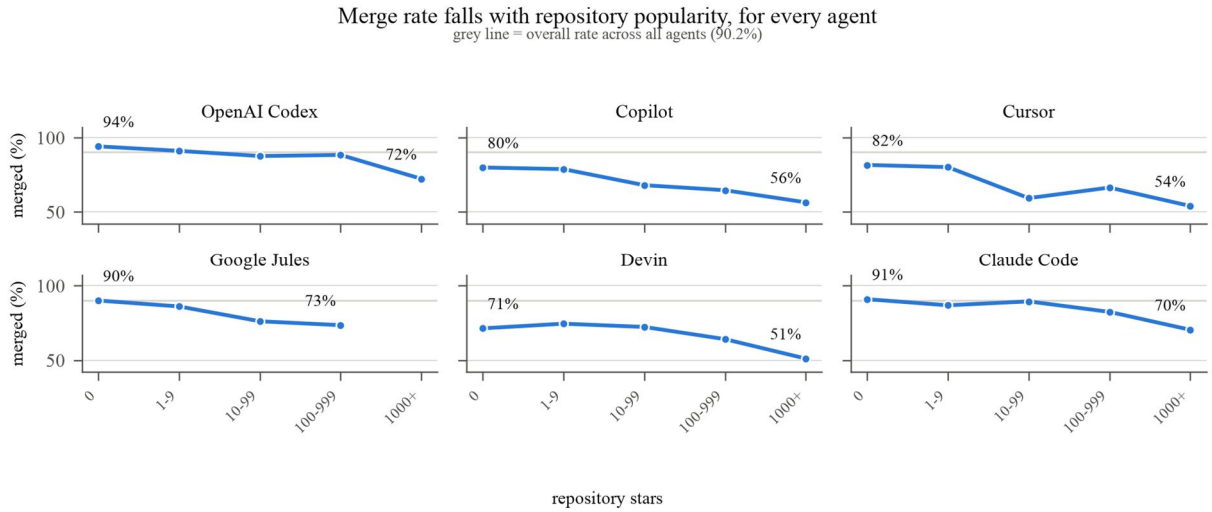


Fig. 1 Merge rate against repository popularity, shown separately for each agent. The decline is present in every panel, so it is a property of the repositories rather than of the agent mix.

D. H2: holding the repository fixed changes conclusions about agents

Confounding by repository is only possible if agents are actually used in different kinds of repository, so that premise is worth establishing before anything is built on it. Table III gives, for each agent, where its own pull requests go. They go to very different places. Devin sends 15.60 percent of its work to repositories with at least 100 stars; Google Jules sends 0.95 percent, a difference of more than sixteen times. A chi-square test on agent against star band gives 118,390.6 on twenty degrees of freedom with p below 0.0001 and Cramér's V of 0.1101 over 2,443,207 pull requests.

TABLE III
WHERE EACH AGENT'S PULL REQUESTS GO, AS A SHARE OF THAT AGENT'S OWN DECIDED TOTAL

Agent	0	1 to 9	10 to 99	100 to 999	1,000+	100+ stars
OpenAI Codex	77.0%	18.9%	2.3%	1.5%	0.3%	1.8%
GitHub Copilot	61.9%	25.8%	5.2%	3.5%	3.5%	7.0%
Cursor	81.5%	13.9%	2.1%	1.1%	1.4%	2.5%
Google Jules	76.4%	20.5%	2.1%	0.6%	0.4%	0.9%
Devin	55.1%	16.9%	12.4%	6.7%	8.9%	15.6%
Claude Code	47.8%	32.4%	8.7%	5.2%	5.9%	11.2%

Rows sum across the five star bands. The last column is the two most popular bands together, which is the population the published literature measures.

That is the mechanism in plain view. An agent whose work goes mostly to unstarred repositories is being graded by lenient markers, and an agent sending one pull request in six to a repository with a hundred stars or more is being graded by strict ones. A pooled comparison between the two is not a like-for-like comparison. Each of the fifteen pairs of agents was therefore estimated twice: once pooled across all repositories, and once with the repository held fixed by the Mantel-Haenszel estimator. Table IV reports the pairs where the conclusion changed.

TABLE IV
AGENT COMPARISONS WHOSE DIRECTION OR MAGNITUDE CHANGES WHEN THE REPOSITORY IS HELD FIXED

Comparison	Pooled OR	Within-repository OR	95% CI	Repos	Change
Copilot vs Devin	3.134	0.758	0.554 to 1.036	96	Reverses
Cursor vs Devin	0.933	1.080	0.865 to 1.349	131	Reverses
Cursor vs Google Jules	2.063	1.370	1.033 to 1.817	101	Halves

The remaining twelve pairs keep both their direction and their approximate magnitude. Full results for all fifteen are in the accompanying results file.

The Copilot and Devin comparison is the clearest case, and Table III has already shown why: Devin's work is concentrated in exactly the repositories that merge least. Pooled across all repositories, a Copilot pull request appears to have more than three times the odds of being merged that a Devin pull request has. Restricted to the 96 repositories where both agents actually operate, the estimate falls to 0.758 with a confidence interval running from 0.554 to 1.036, which includes one. The apparent advantage does not merely shrink; it ceases to be a difference at all, and the point estimate points the other way. What the pooled figure was largely recording is that the two agents are used in different kinds of project.

The second reversal deserves less weight than the first, and the paper does not give it more. The Cursor and Devin comparison moves from 0.933 to 1.080, so the direction of the point estimate flips, but both estimates sit close to no difference and the adjusted interval from 0.865 to 1.349 contains one. What changes there is the sign of a small effect, not the size of a large one.

The honest summary is a modest one, and Fig. 2 shows it. Twelve of fifteen comparisons are essentially unaffected. Two reverse and one loses more than half its magnitude. Confounding by repository is therefore not universal, and this paper does not claim that published agent rankings are generally wrong. It claims something narrower and checkable: that a pooled comparison can reverse, that whether it does cannot be known without testing, and that the test is inexpensive. H2 is therefore supported for those three pairs and not for the other twelve, judged by whether the within-repository estimate departs materially from the pooled one. No formal test of equality between the two estimators was carried out, and none is claimed.

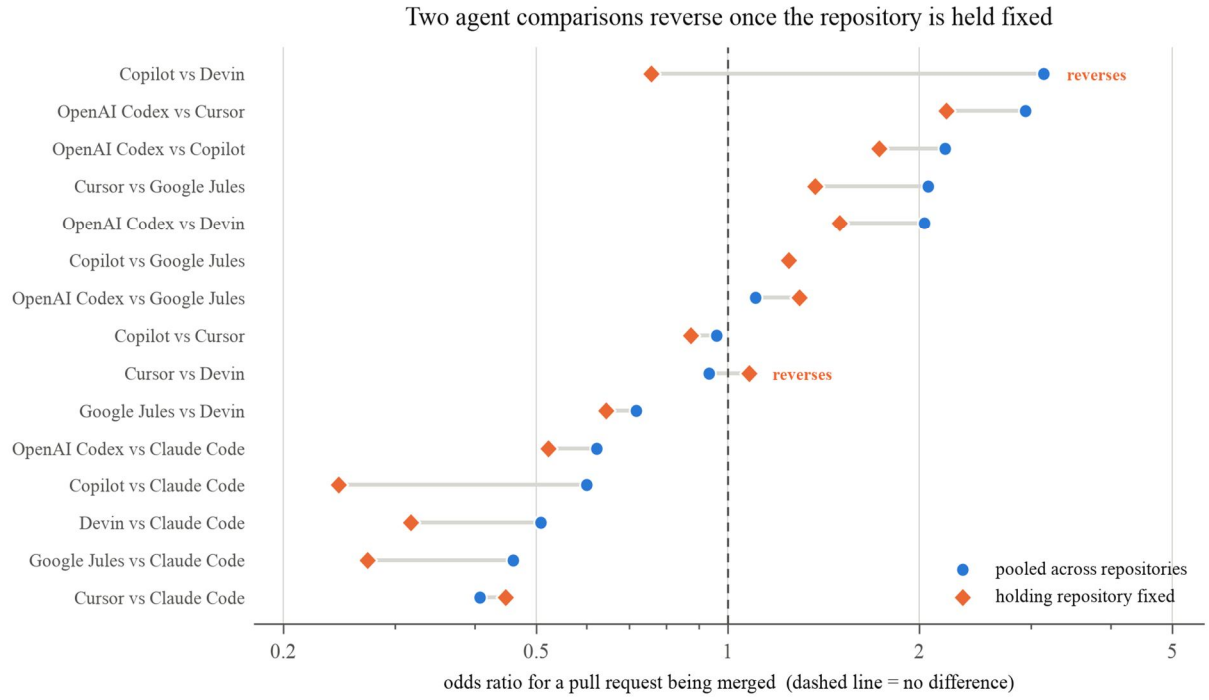


Fig. 2 Every pairwise comparison between agents, estimated pooled across repositories and again with the repository held fixed. Two comparisons cross the line of no difference.

E. H5: the same divide appears in how long decisions take

Merge outcome is one signal and could be idiosyncratic, so the analysis was repeated on an independent one. Fig. 3 gives the median time between a pull request being opened and being decided. In repositories with no stars the median is 51 seconds. In repositories with a thousand or more it is 5.4 hours, a factor of roughly 380.

Comparing the enriched threshold directly, repositories with at least 100 stars take a median of 28 minutes against 63 seconds elsewhere. The Mann-Whitney U statistic is 110,954,113,424 with p below 0.0001, and Cliff's delta is 0.4084, a medium effect. The null hypothesis for H5 is rejected.

One detail should be stated rather than smoothed over, because it does not fit the story neatly. The ordering across the five bands is not monotonic: repositories with 100 to 999 stars decide faster, at a median of 6 minutes, than those with 10 to 99 stars, at 17 minutes. The two ends of the scale are far apart and that separation is robust, but the middle of the scale does not order cleanly, and we have no explanation for the middle that we can support with this data.

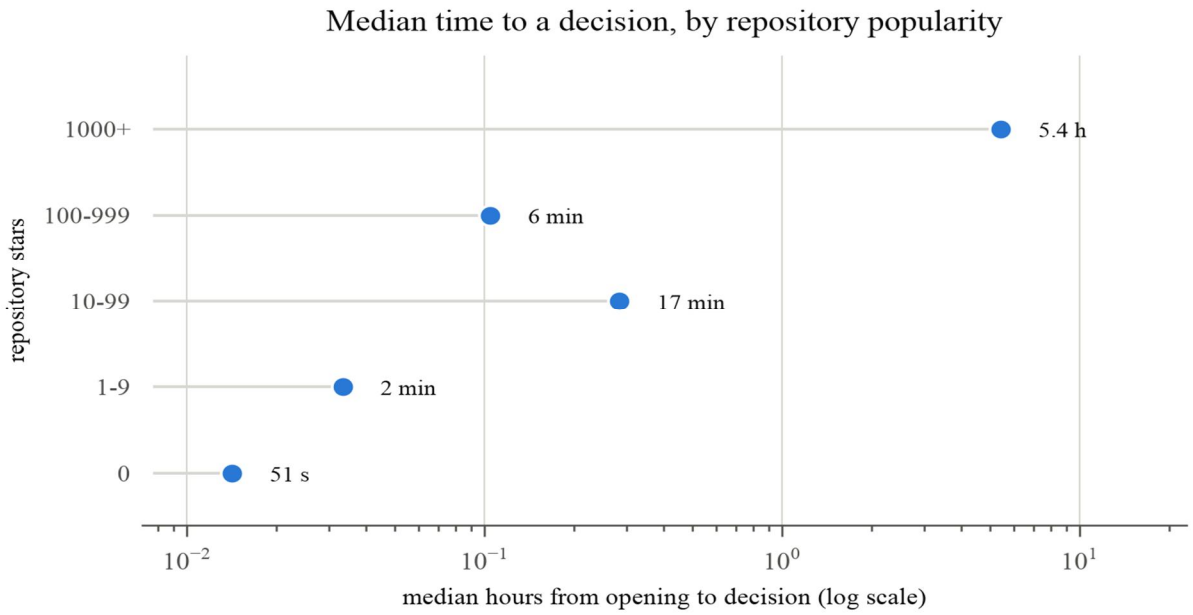


Fig. 3 Median time from opening to decision by repository popularity. Note that the middle bands do not order monotonically.

F. On self-merging, and why it is not re-tested here

A decision reached in a median of 51 seconds invites an obvious interpretation, namely that in small repositories the person who asked the agent for the work is also the person who accepts it, with nobody else involved. That interpretation is not ours to claim. Yoshioka and colleagues have already measured it: 77.51 percent of merged agent pull requests were merged by the account that opened them, against 57.63 percent of human ones, a difference they test with a two-proportion z test at z equal to -29.38 and p below 0.001 [6]. The broad tier of AIDev does not record who performed a merge, so this study cannot re-test their result on the full population, and it does not attempt to. The finding is cited as theirs.

G. What can be seen of review, and what cannot

Review records exist in AIDev only for the enriched subset, which is to say only for repositories above 100 stars. Review activity across the other 97.3 percent of decided agent pull requests is not observable in this dataset at all. That is a limitation, and it is also part of the explanation for why the literature has concentrated where it has.

Within what can be seen, 83,816 review records attach to 71,677 pull requests. Of those records, 58.1 percent were written by human accounts and 41.9 percent by bots. Only 31.0 percent of the enriched pull requests carry any review record at all, and only 24.2 percent carry one written by a human, leaving 75.8 percent with no recorded human review. This is consistent in direction with the finding of Duma and colleagues that human review of agent work is scarcer than review counts alone suggest [7]. It is worth restating what it means: even in the most closely watched repositories in the dataset, three out of four agent pull requests are decided without a recorded human review.

H. Two other repository properties

Fork status and programming language were examined as further candidate confounders. Agent pull requests to forked repositories are merged 85.05 percent of the time against 90.46 percent for non-forks, with an odds ratio of 0.600 and a confidence interval from 0.590 to 0.609, though phi is only 0.0412. Across the twenty-four languages with at least 5,000 decided pull requests the merge rate runs from 80.89 percent for Ruby to 92.79 percent for HTML, significant at p below 0.0001 but with Cramér's V of 0.0805. Both are real and both are small. Neither approaches the popularity effect, which is why the paper concentrates on popularity.

VII. RESULTS

Taking the hypotheses in order:

- R1, replication, supported. Agents differ in acceptance rate; chi-square 104,382.8, p below 0.0001, Cramér's V 0.2063 on 2,451,452 cases. This confirms Pinna and colleagues [5] on a sample roughly three hundred times larger.
- H2, supported in part. Of fifteen pairwise comparisons, two reverse direction and one loses more than half its magnitude once the repository is held fixed. Twelve are unchanged. The effect is real but not general.
- H3, supported. Merge rate falls from 91.58 percent at zero stars to 60.05 percent at a thousand or more. The subset used by prior work merges at 73.67 percent against 90.63 percent elsewhere, odds ratio 0.289, and holds 2.7 percent of the decided pull requests. The gradient holds within every agent.
- H5, supported, and exploratory rather than pre-registered. Median time to decision rises from 51 seconds to 5.4 hours across the same range, Cliff's delta 0.4084, a medium effect, though the middle bands are not monotonic.
- R2, the withdrawn automation hypothesis H1, and the pre-registered hypothesis on review effort were not tested here, for the reasons given in Sections VI-F and IV-D.

VIII. DISCUSSION

The practical consequence is that an acceptance rate quoted for a coding agent is only interpretable alongside a description of where it was measured. A figure near ninety percent and a figure near sixty percent can both be correct for the same agent in the same period, because they describe different populations of repository. Neither is wrong; a comparison between them is.

This also offers a plain reading of why published figures disagree. Watanabe and colleagues report 83.8 percent [4], Pinna and colleagues report acceptance for a single agent ranging from about 60 to about 89 percent depending on the type of task [5], and this paper measures 90.15 percent overall. Those are not necessarily contradictory results about agent capability. They are consistent with a single underlying pattern observed through different sampling frames.

There is a further point about what merging means. In a repository where the median decision takes under a minute, a merge is a weak signal of quality, because almost no assessment can have taken place. In a repository where it takes hours and passes a review, the same merge carries more information. Aggregating both into one percentage treats them as equivalent evidence when they are not. Pinna and colleagues make a related argument for complementing acceptance with static analysis and maintenance measures [5], and the maintenance-focused work of Sawada and colleagues [11] points the same way.

For researchers the recommendations are concrete and cheap to apply. Report the popularity distribution of the repositories in the sample, because it is a sampling frame and should be described as one. Estimate within repositories where the design permits it, using a stratified estimator or a mixed-effects model, and report the pooled and adjusted estimates side by side so readers can see whether they diverge. Treat an acceptance rate as jointly describing agent and project, and avoid comparing figures across studies with different frames.

For practitioners choosing between tools, the published rankings should be read with care. Two of fifteen comparisons in this data reverse under adjustment. A team's own repository, with its own review conventions, is the relevant context, and no pooled public figure substitutes for it.

IX. LIMITATIONS AND THREATS TO VALIDITY

A. What merging does and does not mean

The central construct threat is that merging is not quality. A merged pull request may contain defects, and a rejected one may be sound but unwanted. This paper uses merge outcome because the literature it examines uses it, and part of its argument is that the measure is weaker than its use implies. That argument does not exempt this paper: our own merge rates inherit the same weakness.

B. Association, not cause

Nothing here is causal. Popular repositories differ from unpopular ones in many ways that travel together, including number of maintainers, presence of continuous integration, contribution guidelines and review requirements. Star count is a proxy for that bundle and not a mechanism. The Mantel-Haenszel estimator removes confounding by repository identity, which is a strong control, but it cannot address confounding within a repository, such as agents being assigned harder tasks in some projects than others. Pinna and colleagues show task type matters a great deal [5], and task type is not controlled in this analysis.

C. What the dataset covers

All results describe AIDev between December 2024 and October 2025 and do not extend beyond it. Agents are identified by AIDev's own procedure; any agent that contributes under a developer's personal account rather than its own identity will be invisible here, and such contributions would be counted as human elsewhere. The dataset is updated over time, so the counts in this paper are tied to the snapshot of 1 September 2026 and will not reproduce exactly at a later date.

Volume is very unevenly distributed across agents, with a single agent supplying about three quarters of all pull requests, so pooled figures are dominated by it. That is another reason the within-repository estimates matter more here than the pooled ones. Six agents are represented; the market contains more.

D. Comparisons this study cannot make

Two absences should be stated plainly. First, the broad tier of AIDev contains no human-authored pull requests, so this paper offers no human baseline of its own and relies on published work for that comparison [6], [7]. Second, the dataset contains no rule-based automation such as Dependabot or Renovate. A hypothesis comparing language-model agents against that older automation was pre-registered and then withdrawn, untested, when the study was scoped to this dataset. Without it we cannot separate what is specific to artificial intelligence from what is common to any bot opening pull requests at volume, and readers should not infer that separation from anything reported here.

E. Statistical caution

With samples in the millions, statistical significance is close to guaranteed and carries almost no information. Every finding above is therefore reported with an effect size, and the effect sizes are mixed: the popularity gradient is substantial, the agent-versus-agent effect is small to moderate at 0.2063, and the language effect at 0.0805 is small enough that we would not build an argument on it. Where a confidence interval includes no difference, as in the Copilot and Devin comparison after adjustment, the paper says so rather than presenting a reversal as established.

X. RECOMMENDATIONS AND FUTURE WORK

1) Studies of agent-authored pull requests should publish the star distribution of their repositories, and should report pooled and within-repository estimates together rather than pooled estimates alone.
2) The withdrawn hypothesis H1 is the most direct next step: collect pull requests from rule-based bots such as Dependabot and Renovate in the same repositories as agent pull requests, and test whether the differences attributed to artificial intelligence survive a comparison against ordinary automation. The bot literature already supplies the classification method [14].
3) Acceptance should be paired with an outcome measured after the merge, such as subsequent reverts, defect-fixing commits or maintenance effort, along the lines of Sawada and colleagues [11]. A merge granted in under a minute needs corroboration.
4) Extending the observation window past October 2025 would show whether the popularity gradient narrows as review practices for agent contributions mature.
5) The unexplained non-monotonicity in decision time across the middle popularity bands deserves direct study; we report it without being able to account for it.

XI. CONCLUSION

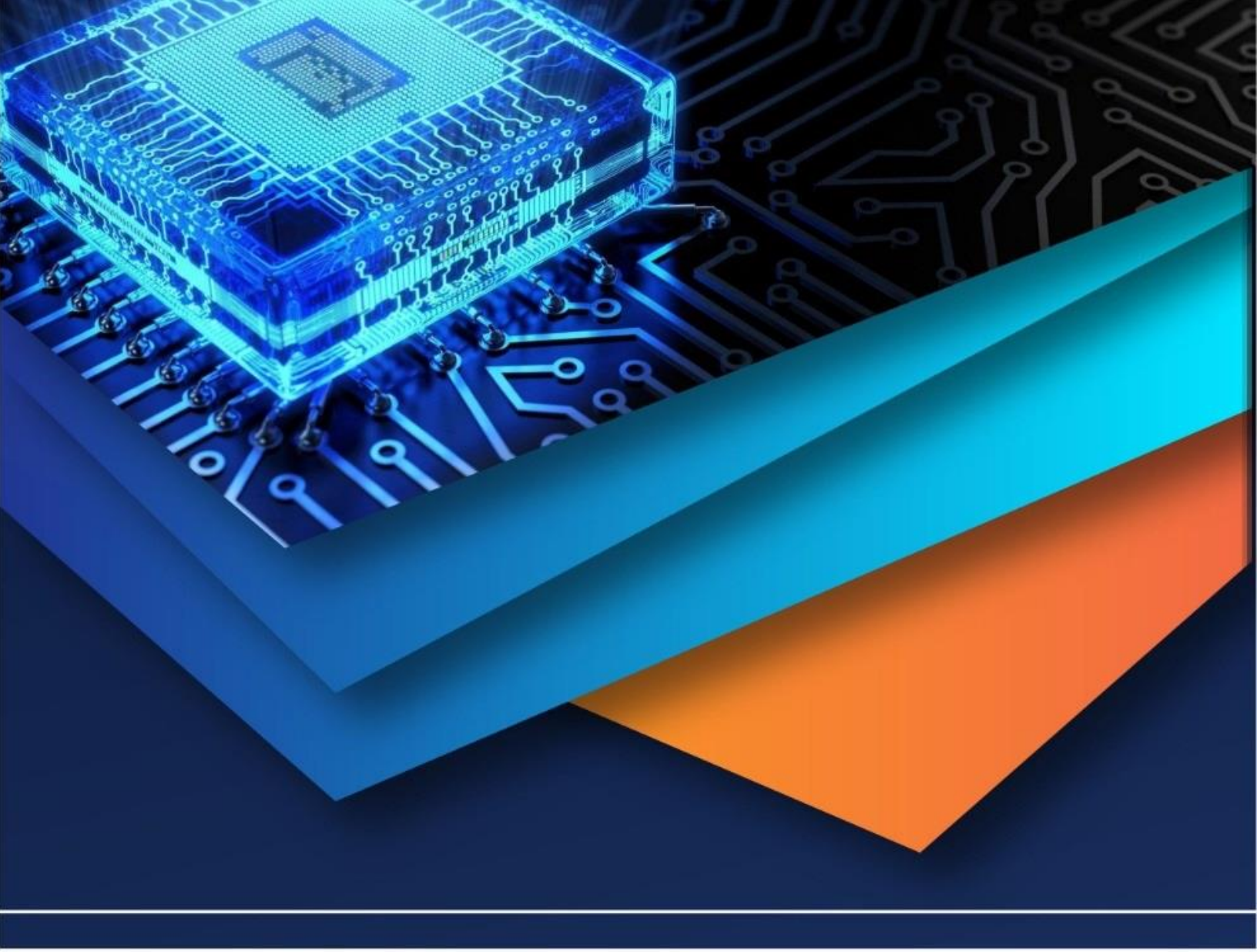
Across 2,451,452 decided pull requests written by six coding agents, the single strongest thing determining whether an agent's work is accepted is not which agent wrote it but where it was sent. Acceptance falls from 91.58 percent in repositories with no stars to 60.05 percent in those with a thousand or more, and that decline is present inside every agent examined. The portion of the data on which the published literature rests consists entirely of repositories above 100 stars, holds 2.7 percent of the decided pull requests, and merges them roughly seventeen percentage points below the rest of the population.

Holding the repository fixed changes the comparison between agents in three of fifteen cases and reverses it in two, including one where a threefold apparent advantage becomes no measurable difference. The same split shows in decision time, from a median of 51 seconds in unstarred repositories to 5.4 hours in the most popular. Even inside the most scrutinised repositories available, three quarters of agent pull requests carry no recorded human review.

The conclusion is not that coding agents are worse than reported, nor better. It is that the number by which they are commonly judged belongs partly to the projects that receive their work, and that any study reporting it should say which projects those were.

REFERENCES

[1] H. Li, H. Zhang, and A. E. Hassan, "AIDev: Studying AI Coding Agents on GitHub," arXiv preprint arXiv:2602.09185, 2026. [Online]. Available: https://arxiv.org/abs/2602.09185
[2] R. Robbes, T. Matricon, T. Degueule, A. Hora, and S. Zacchiroli, "Agentic Much? Adoption of Coding Agents on GitHub," arXiv preprint arXiv:2601.18341, 2026. [Online]. Available: https://arxiv.org/abs/2601.18341
[3] T. A. Ghaleb, "Fingerprinting AI Coding Agents on GitHub," arXiv preprint arXiv:2601.17406, 2026. [Online]. Available: https://arxiv.org/abs/2601.17406
[4] [4] 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," arXiv preprint arXiv:2509.14745, 2025. [Online]. Available: https://arxiv.org/abs/2509.14745
[5] G. Pinna, J. Gong, D. Williams, and F. Sarro, "Comparing AI Coding Agents: A Task-Stratified Analysis of Pull Request Acceptance," arXiv preprint arXiv:2602.08915, 2026. [Online]. Available: https://arxiv.org/abs/2602.08915
[6] H. Yoshioka, T. Monno, H. Tokumasu, T. Wakamatsu, Y. Ota, N. Weeraddana, and K. Matsumoto, "Let's Make Every Pull Request Meaningful: An Empirical Analysis of Developer and Agentic Pull Requests," arXiv preprint arXiv:2601.18749, 2026. [Online]. Available: https://arxiv.org/abs/2601.18749
[7] K. Duma, P. Wróblewski, J. Bobińska, J. Winiarska, and P. Przymus, "These Aren't the Reviews You're Looking For How Humans Review AI-Generated Pull Requests," arXiv preprint arXiv:2605.02273, 2026. [Online]. Available: https://arxiv.org/abs/2605.02273
[8] C. Nachuma and M. Zibran, "When AI Teammates Meet Code Review: Collaboration Signals Shaping the Integration of Agent-Authored Pull Requests," arXiv preprint arXiv:2602.19441, 2026. [Online]. Available: https://arxiv.org/abs/2602.19441
[9] M. L. Siddiq, X. Zhao, V. C. Lopes, B. Casey, and J. C. S. Santos, "Security in the Age of AI Teammates: An Empirical Study of Agentic Pull Requests on GitHub," arXiv preprint arXiv:2601.00477, 2026. [Online]. Available: https://arxiv.org/abs/2601.00477
[10] M. Abujadallah, A. Arabat, and M. Sayagh, "Understanding the Rejection of Fixes Generated by Agentic Pull Requests — Insights from the AIDev Dataset," arXiv preprint arXiv:2606.13468, 2026. [Online]. Available: https://arxiv.org/abs/2606.13468
[11] S. Sawada, T. Shirai, Y. Kashiwa, K. Yamaguchi, H. Iwata, and H. Iida, "To What Extent Does Agent-generated Code Require Maintenance? An Empirical Study," arXiv preprint arXiv:2605.06464, 2026. [Online]. Available: https://arxiv.org/abs/2605.06464
[12] M. Wessel, A. Serebrenik, I. Wiese, I. Steinmacher, and M. A. Gerosa, "Quality Gatekeepers: Investigating the Effects of Code Review Bots on Pull Request Activities," arXiv preprint arXiv:2103.13547, 2021. [Online]. Available: https://arxiv.org/abs/2103.13547
[13] M. Wyrich, R. Ghit, T. Haller, and C. Müller, "Bots Don't Mind Waiting, Do They? Comparing the Interaction With Automatically and Manually Created Pull Requests," arXiv preprint arXiv:2103.03591, 2021. [Online]. Available: https://arxiv.org/abs/2103.03591
[14] M. Golzadeh, A. Decan, D. Legay, and T. Mens, "A ground-truth dataset and classification model for detecting bots in GitHub issue and PR comments," arXiv preprint arXiv:2010.03303, 2020. [Online]. Available: https://arxiv.org/abs/2010.03303
[15] X. Zhang, Y. Yu, G. Gousios, and A. Rastogi, "Pull Request Decision Explained: An Empirical Overview," arXiv preprint arXiv:2105.13970, 2021. [Online]. Available: https://arxiv.org/abs/2105.13970
[16] D. Legay, A. Decan, and T. Mens, "On the impact of pull request decisions on future contributions," arXiv preprint arXiv:1812.06269, 2018. [Online]. Available: https://arxiv.org/abs/1812.06269
[17] K. A. Hasan, M. Macedo, Y. Tian, B. Adams, and S. Ding, "Understanding the Time to First Response In GitHub Pull Requests," arXiv preprint arXiv:2304.08426, 2023. [Online]. Available: https://arxiv.org/abs/2304.08426
[18] S. Peng, E. Kalliamvakou, P. Cihon, and M. Demirer, "The Impact of AI on Developer Productivity: Evidence from GitHub Copilot," arXiv preprint arXiv:2302.06590, 2023. [Online]. Available: https://arxiv.org/abs/2302.06590
[19] F. Xu, P. K. Medappa, M. M. Tunc, M. Vroegindeweij, and J. C. Fransoo, "AI-Assisted Programming Decreases the Productivity of Experienced Developers by Increasing the Technical Debt and Maintenance Burden," arXiv preprint arXiv:2510.10165, 2025. [Online]. Available: https://arxiv.org/abs/2510.10165
[20] C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press, and K. Narasimhan, "SWE-bench: Can Language Models Resolve Real-World GitHub Issues?," arXiv preprint arXiv:2310.06770, 2023. [Online]. Available: https://arxiv.org/abs/2310.06770



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY