Who Writes the Docs in SE 3.0?

Agent vs. Human Documentation Pull Requests

Kazuma Yamasaki
Software Engineering Laboratory
Nara Institute of Science and Technology
Ikoma, Nara, Japan
yamasaki.kazuma.yj9@naist.ac.jp

Joseph Ayobami Joshua
Software Engineering Laboratory
Nara Institute of Science and Technology
Ikoma, Nara, Japan
joseph.ayobami_joshua.je3@naist.ac.jp

Tasha Settewong
Software Engineering Laboratory
Nara Institute of Science and Technology
Ikoma, Nara, Japan
tasha.settewong.ts1@naist.ac.jp

Mahmoud Alfadel
Department of Computer Science (Faculty of Science)
University of Calgary
Calgary, Alberta, Canada
mahmoud.alfadel@ucalgary.ca

Kazumasa Shimari
Software Engineering Laboratory
Nara Institute of Science and Technology
Ikoma, Nara, Japan
k.shimari@is.naist.jp

Kenichi Matsumoto
Software Engineering Laboratory
Nara Institute of Science and Technology
Ikoma, Nara, Japan
matumoto@is.naist.jp

Abstract

As software engineering moves toward SE 3.0, AI agents are increasingly used to carry out development tasks and contribute changes to software projects. It is therefore important to understand the extent of these contributions and how human developers review and intervene, since these factors shape the risks of delegating work to AI agents. While recent studies have examined how AI agents support software development tasks (e.g., code generation, issue resolution, and PR automation), their role in documentation tasks remains underexplored–even though documentation is widely consumed and shapes how developers understand and use software.

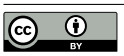
Using the AIDev, we analyze 1,997 documentation-related pull requests (PRs) authored by AI agents and human developers, where documentation PRs are those that create or modify project documentation artifacts. We find that AI agents submit substantially more documentation-related PRs than humans in the studied repositories. We further observe that agent-authored documentation edits are typically integrated with little follow-up modification from humans, raising concerns about review practices and the reliability of agent-generated documentation. Overall, while AI agents already contribute substantially to documentation workflows, our results suggest concerns for emerging challenges for documentation quality assurance and human–AI collaboration in SE 3.0.

CCS Concepts

• **Software and its engineering** → **Documentation**; *Software libraries and repositories*; • **Computing methodologies** → *Intelligent agents.*

Keywords

AI Agents, Software Documentation, Mining Software Repositories



This work is licensed under a Creative Commons Attribution 4.0 International License.
MSR '26, Rio de Janeiro, Brazil
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2474-9/26/04
https://doi.org/10.1145/3793302.3793612

ACM Reference Format:
Kazuma Yamasaki, Joseph Ayobami Joshua, Tasha Settewong, Mahmoud Alfadel, Kazumasa Shimari, and Kenichi Matsumoto. 2026. Who Writes the Docs in SE 3.0?: Agent vs. Human Documentation Pull Requests. In *23rd International Conference on Mining Software Repositories (MSR '26), April 13–14, 2026, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3793302.3793612

1 Introduction

SE 3.0 [8] marks a shift toward software development environments in which autonomous AI agents act as teammates that can propose and implement changes, rather than passive assistants that only respond to prompts. Recent work highlights this transition and the growing capabilities of agentic tools to generate and submit project updates through typical collaboration mechanisms such as pull requests (PRs) [8, 16]. Industry reports similarly suggest rapid adoption of AI assistance in practice [1].

To quantify agents' participation in development workflows and assess their impact, prior work has largely focused on coding-centric activities [5, 16, 18]. However, we still know little about agent involvement in documentation—a critical artifact for understanding, onboarding, maintenance, and reuse—and how human developers respond to such contributions in real PR workflows. MSR research has extensively studied documentation artifacts and quality issues [6, 9, 15], but evidence on agent-authored documentation in practice remains limited.

Prior work on documentation generation by large language models (LLMs) has mostly evaluated output quality in controlled settings (e.g., correctness, usefulness, and readability) [12]. In contrast, agent-authored documentation changes occur within real collaboration workflows, where integration, review, and subsequent revision determine whether documentation is trusted and maintained. This setting raises two practical questions: (i) how prevalent agent-authored documentation changes are relative to human-authored ones, and (ii) whether documentation changes proposed by agents receive meaningful human follow-up after being integrated.

To answer these questions, we analyze documentation-related PRs in the AIDev [16] and compare agent-authored and human-authored contributions. Specifically, we structure our study around the following research questions:

(1) **RQ_1 (Prevalence): How many documentation-related PRs are created by AI agents compared to humans?** Of the 1,997 documentation-related PRs, 1,478 are agent-authored and 519 are human-authored. At the file level, 66.1% of changed files are edited only by agents, 30.2% only by humans, and 3.7% are edited by both.

(2) **RQ_2 (Integration): How often are agent-authored documentation PRs accepted?** We find that an average of 86.8% of line edits made by agents in source code are accepted by human developers. Among accepted agent edits, human follow-up is often limited: in 85.7% of cases, agent additions exceed subsequent human deletions on the same files, and 34.5% of cases show zero human deletions after agent additions.

These results demonstrate that documentation has become a primary entry point for AI agent contributions, yet integrating agent-authored documentation introduces reliability risks that are not automatically mitigated by existing review practices. This study lays the foundation for future work on strengthening documentation quality assurance in SE 3.0 and on designing human–AI workflows and tools that better support accountable review and maintenance of agent-generated documentation.

We release a replication package to support reproducibility [1]. Our replication package includes (i) the derived analysis subset, (ii) scripts to reproduce the subset from the original dataset, and (iii) scripts to reproduce all analyses and figures.

2 Existing Research

2.1 Software Documentation in Modern Repositories

Software documentation is a critical artifact for understanding, onboarding, maintenance, and reuse, yet it is fragile and costly to maintain. Aghajani et al. mined 878 documentation-related artifacts from mailing lists, issue trackers, Stack Overflow, and PRs, producing a taxonomy of 162 documentation issue types. They report common problems such as outdated, incomplete, and inconsistent information, and conclude that developers and users prefer documentation that is correct, complete, up-to-date, usable, maintainable, readable, and useful [2]. Prana et al. analyzed 4,226 README sections from 393 GitHub repositories and proposed a content taxonomy, showing that many projects lack basic information about purpose and status despite rich usage instructions [14]. Work on code–comment co-evolution further shows that comments often lag behind code changes, causing inconsistency and confusion and motivating continuous documentation maintenance [17].

Together, this literature underscores that documentation is critical yet fragile, and that understanding documentation work in repositories remains an important MSR problem. In this work, we study documentation-related PRs, with an emphasis on documentation edits made by AI agents alongside humans.

[1] https://github.com/NAIST-SE/msr2026-docs-prs-replication

2.2 LLM-Based Documentation Editing in Pull Requests

LLM-based tools are increasingly used for documentation work in practice. Observational evidence from software engineering practice shows that developers use ChatGPT for tasks beyond code generation, including improving documentation wording and related documentation work [10]. Repository-scale evidence also suggests that documentation is a common task type for AI teammates: AIDev reports that documentation-related PRs account for a non-trivial share of agentic PRs and often exhibit comparatively high acceptance rates [11]. Moreover, MSR studies of ChatGPT usage in PR workflows indicate that developers use ChatGPT for purposes beyond code generation, including documentation refinement [13].

Despite these signs of widespread use, we still lack a PR-level understanding of how LLM-driven documentation edits are carried out in real repositories. Prior work does not isolate documentation-focused PRs nor quantify how documentation changes are packaged and reviewed. Our study addresses this gap by focusing specifically on documentation-related PRs and by quantifying, at scale, the structure and outcomes of documentation edits authored by AI agents and humans.

3 Approach

3.1 Data Collection

We construct an RQ-specific subset by extracting Agentic-PRs and Human-PRs that involve documentation changes from the AIDev [11], and then retrieving detailed commit-level information for each PR via GitHub API. Figure 1 shows the data extraction process used in our analysis.

We first extract the PRs related to documentation changes from AIDev's *"pull_request"* (33,596 PRs from 2,807 repos) and *"human_pull_request"* (6,618 PRs from 818 repos) tables, which store PR-level metadata. To ensure a fair comparison between Agentic-PRs and Human-PRs under the same repository conditions, we restrict Agentic-PRs to PRs from repositories with more than 500 stars (1,478 repos); the Human-PRs subset in AIDev is already filtered to the same threshold. As a result, we obtained 1,478 documentation-related Agentic-PRs and 519 documentation-related Human-PRs.

Next, we retrieve the commit-level information for each collected PR. Commit-level details for Human-PRs are not included in AIDev, so we retrieved them via the GitHub API. To ensure a consistent data collection procedure and a fair comparison, we also fetched commit-level details for Agentic-PRs via the GitHub API, even though they are available in AIDev.

For each PR, we collect commit metadata, the total number of added and deleted lines across the PR, and per-commit file-level change information, including file names, change types, and the numbers of added and deleted lines for each modified file. From the Agentic-PRs, we retrieved 3,653 commits and 35,428 file-change records; from the Human-PRs, we retrieved 1,889 commits and 17,013 file-change records. We additionally constructed a subset of documentation-related PRs that actually modify documentation files. Following the extension- and path-based heuristic used in prior work [4], we define documentation files as those with extensions such as *".md"* and *".txt"*, or whose paths contain tokens like *"/docs/"*

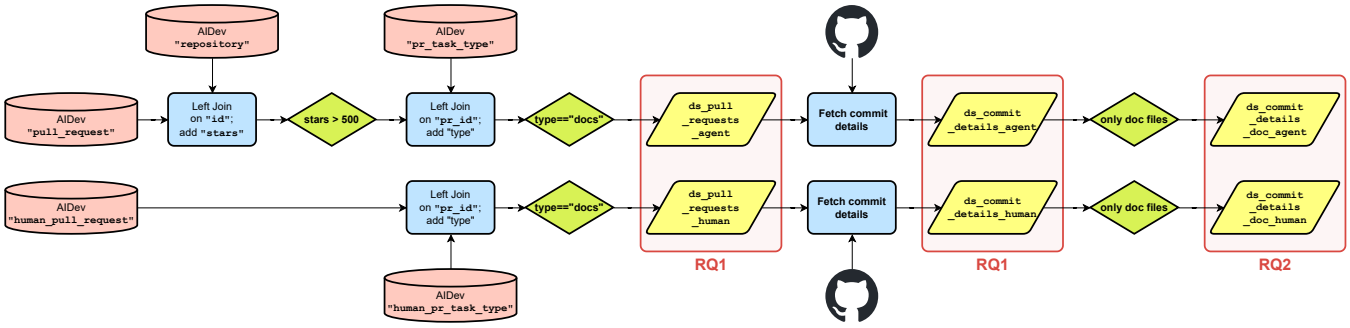


Figure 1: Flowchart of Data Collection

or *"README"*. From the Agentic-PRs, we retrieved 2,363 commits that include documentation-file changes and 14,410 documentation-file change records; from the Human-PRs, we retrieved 993 such commits and 4,236 documentation-file change records.

3.2 Analysis for RQ1 (Prevalence)

To compare documentation contributions between agentic and human developers, we first examine the number of PRs created by each group. Using the documentation-related PRs filtered from the AIDev, we separate Agentic-PRs and Human-PRs; for Agentic-PRs, we further split them by agent type and compare PR counts across agents and against human developers.

We also analyze how agentic and human developers collaborate at the file level using all file-change records in our constructed dataset. We group commits by file path and count, for each file, whether all commits are made exclusively by one group or whether the file is co-edited by both groups through commits from agentic and human developers.

Moreover, to assess the extent to which PRs actually modify documentation, we analyze the types of files included in the commits of each PR. In this analysis, we categorize files into "documentation" ("docs") and "non-documentation" ("non-docs"). Following existing research [4], we treat files with extensions such as *".md"* and *".txt"*, or whose paths contain tokens like *"/docs/"* or *"README"*, as "docs", and define all other files as "non-docs". For each PR, we inspect the modified files and determine whether the PR changes docs only, non-docs only, or both.

3.3 Analysis for RQ2 (Integration)

To examine how often AI's documentation-related changes are accepted, we inspect each file's commit history and extract cases where an agent commit is followed by the file's next human commit. After applying this filtering, we identify 119 such human-after-agent cases. For each such commit, we measure the number of lines added by the agent and the number of lines deleted by the human. If the human deletes a large number of lines relative to the agent's additions, we can infer that the agent's changes were largely removed.

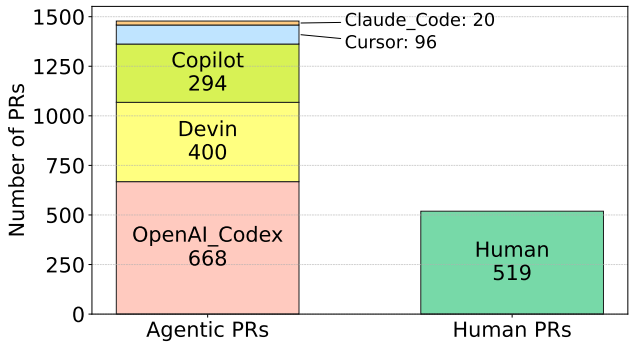


Figure 2: Number of docs-related PRs

4 Results

4.1 Results for RQ1 (Prevalence)

We show the number of documentation-related PRs in the Figure 2. Agentic PRs (1,478) substantially outnumber Human PRs (519), indicating that AI agents are widely used even for documentation edits. Within the Agentic PRs, *"OpenAI_Codex"* accounts for 45%, a distribution that closely mirrors the breakdown observed across all PRs (not limited to documentation) created by AI agents [11].

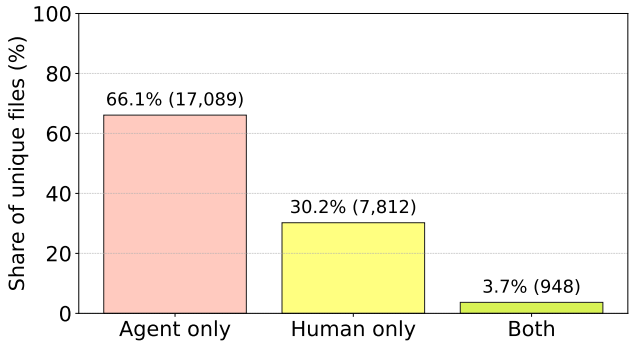


Figure 3: Breakdown of contributions by file

We also show the breakdown of contributions by file in Figure 3. Only 3.7% of files (948) are co-edited by both agents and humans,

indicating that most (96.3%) files are edited exclusively by either agents or humans rather than collaboratively.

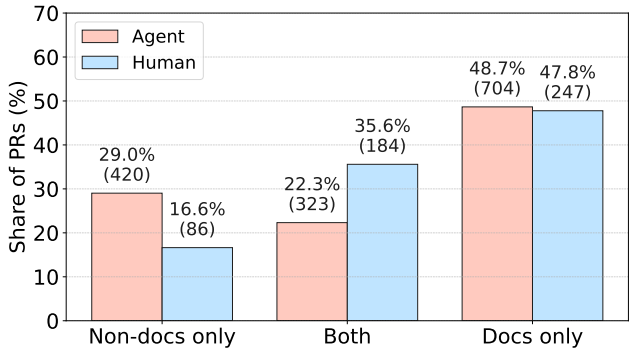


Figure 4: Distribution of PRs categorized by the types of files

Figure 4 shows the distribution of PRs categorized by the types of files they include. Among documentation-related PRs, the share that modifies only documentation is comparable between agents and humans (48.7% and 47.8%, respectively). In addition, a substantial fraction of documentation-related PRs modifies only "non-docs"—i.e., does not edit documentation at all—accounting for 29.0% and 16.6% for agent and human PRs, respectively. In conventional commit conventions, the *"docs"* type is intended for *documentation-only* changes [3, 7]; therefore, PRs labeled as *"docs"* that also modify non-doc files violate this expectation; such mixed-scope PRs are associated with higher reviewer effort and a greater risk that changes are overlooked, which can reduce development productivity.

RQ1 Summary

Answering **RQ1**, we find that AI agents are widely used for documentation edits and that 96.3% of edited files are handled exclusively by either agents or humans; moreover, 29.0% of agent-authored documentation-related PRs do not edit any documentation files, suggesting non-compliance with commit conventions.

4.2 Results for RQ2 (Integration)

Figure 5 shows a scatter plot (log scale) of the number of lines added by agents versus the number of lines deleted by humans. Across 119 files, we observe 17 cases (14.3%) where human deletions exceed agent additions, and 102 cases (85.7%) where agent additions exceed or match human deletions. Notably, within the latter group, there are 41 files (34.5% overall) in which humans perform no deletions at all after agent additions. Moreover, when agent additions exceed human deletions, most added lines remain in the integrated version: on average, 86.8% of the added lines persist (median 98.7%). Taken together, these results suggest that agent-authored documentation changes tend to be accepted by human developers, yet a substantial subset appears to be merged with minimal follow-up, raising questions about the level of scrutiny that agent edits receive.

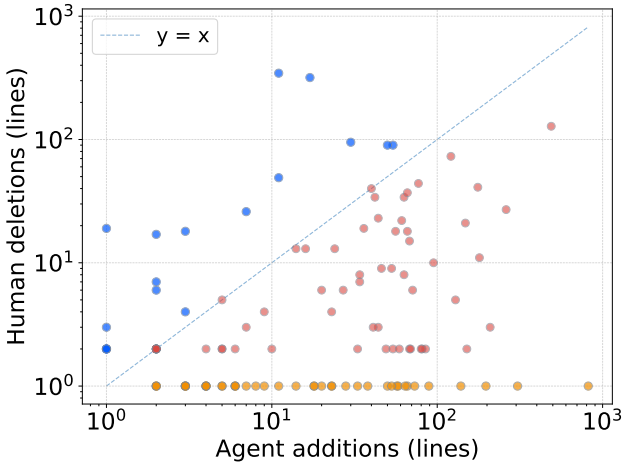


Figure 5: Human Followup Impact

RQ2 Summary

Answering **RQ2**, agent-authored documentation changes are largely retained, with limited human follow-up edits. Agent additions exceed or match human deletions in 85.7% of files, including 34.5% with *no* human deletions. When agent additions dominate, most added lines remain (mean 86.8%, median 98.7%).

5 Discussion and Future Work

This paper makes three contributions toward understanding documentation work in SE 3.0: (1) using AIDev, we conduct a large-scale comparison showing that agents drive the volume of documentation-related PRs and that file co-editing between agents and humans is relatively rare; (2) we analyze how documentation changes are packaged in practice, showing that agent "docs"-related PRs more often include non-docs files and that some agent-authored edits are accepted with little observable human follow-up, raising review and quality-assurance concerns; and (3) we release a replication package with the analysis data and scripts to support reproducibility and follow-up studies.

We discuss three threats to validity.

For external validity, our results are limited by the set of repositories covered by the AIDev [11]. AIDev is constructed only from repositories that contain agent-authored PRs; although we extract Agentic-PRs and Human-PRs under identical repository conditions for a fair comparison, it is important to note that repositories that do not use agents at all are not represented in the dataset.

For internal validity, our identification of docs versus non-docs files has limitations. Following existing research [4], we classify documentation files using a lightweight heuristic based on file extensions and paths; however, some PRs labeled as documentation-related in AIDev do not actually modify documentation files (e.g., 29.0% of agent-authored documentation-related PRs touch only non-docs files). This mismatch introduces uncertainty into our estimates and interpretation of documentation work, motivating

future work to better align task labels, file classification, and actual documentation edits in practice.

Moreover, our interpretation of "limited human follow-up" in RQ2 relies on quantitative signals (e.g., no human deletions and high line-retention) and does not yet consider human additions, modifications, or review discussions. Future work will move beyond aggregate metrics by examining cases with little or no follow-up (e.g., review comments, approval patterns, and subsequent edits) to assess how labels, file-level changes, and review practices align in practice.

Acknowledgments

This work has been supported by JSPS KAKENHI No. JP24K14895.

References

[1] 2024. AI | 2024 Stack Overflow Developer Survey. https://survey.stackoverflow.co/2024/ai. Accessed: 2025-12-26.

[2] Emad Aghajani, Csaba Nagy, Olga Lucero Vega-Márquez, Mario Linares-Vásquez, Laura Moreno, Gabriele Bavota, and Michele Lanza. 2019. Software documentation issues unveiled. In *Proceedings of the 41st International Conference on Software Engineering* (Montreal, Quebec, Canada) *(ICSE '19)*. IEEE Press, 1199–1210. doi:10.1109/ICSE.2019.00122

[3] Ned Batchelder. 2022. OEP-51: Conventional Commits. Open edX Proposals (Best Practice). https://docs.openedx.org/projects/openedx-proposals/en/latest/best-practices/oep-0051-bp-conventional-commits.html Last modified: 2022-10-19; Created: 2021-02-01; Accessed: 2025-12-26.

[4] Egor Bogomolov, Aleksandra Eliseeva, Timur Galimzyanov, Evgeniy Glukhov, Anton Shapkin, Maria Tigina, Yaroslav Golubev, Alexander Kovrigin, Arie Van Deursen, Maliheh Izadi, et al. 2024. Long code arena: a set of benchmarks for long-context code models. *arXiv preprint arXiv:2406.11612* (2024).

[5] Mark Chen. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).

[6] Zadia Codabux, Melina Vidoni, and Fatemeh H. Fard. 2021. Technical Debt in the Peer-Review Documentation of R Packages: a rOpenSci Case Study. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*. 195–206. doi:10.1109/MSR52588.2021.00032

[7] conventionalcommit contributors. 2025. commitlint: commitlint checks if your commit messages meets the conventional commit format. GitHub repository documentation. https://github.com/conventionalcommit/commitlint Commit Types table includes "docs: Documentation only changes"; Accessed: 2025-12-26.

[8] Ahmed E Hassan, Gustavo A Oliva, Dayi Lin, Boyuan Chen, Zhen Ming, et al. 2024. Towards AI-native software engineering (SE 3.0): A vision and a challenge roadmap. *arXiv preprint arXiv:2410.06107* (2024).

[9] Shohei Ikeda, Akinori Ihara, Raula Gaikovina Kula, and Kenichi Matsumoto. 2019. An empirical study of readme contents for javascript packages. *IEICE TRANSACTIONS on Information and Systems* 102, 2 (2019), 280–288.

[10] Ranim Khojah, Mazen Mohamad, Philipp Leitner, and Francisco Gomes de Oliveira Neto. 2024. Beyond code generation: An observational study of chatgpt usage in software engineering practice. *Proceedings of the ACM on Software Engineering* 1, FSE (2024), 1819–1840.

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

[12] William Macke and Michael Doyle. 2024. Testing the effect of code documentation on large language model code understanding. *arXiv preprint arXiv:2404.03114* (2024).

[13] Daniel Ogenrwot and John Businge. 2025. PatchTrack: A Comprehensive Analysis of ChatGPT's Influence on Pull Request Outcomes. *arXiv preprint arXiv:2505.07700* (2025).

[14] Gede Artha Prana, Christoph Treude, Ferdian Thung, Thushari Atapattu, and David Lo. 2019. Categorizing the Content of GitHub README Files. *Empirical Softw. Engg.* 24, 3 (June 2019), 1296–1327. doi:10.1007/s10664-018-9660-3

[15] Henry Tang and Sarah Nadi. 2023. Evaluating Software Documentation Quality. In *2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR)*. 67–78. doi:10.1109/MSR59073.2023.00023

[16] Miku Watanabe, Hao Li, Yutaro Kashiwa, Brittany Reid, Hajimu Iida, and Ahmed E Hassan. 2025. On the use of agentic coding: An empirical study of pull requests on github. *arXiv preprint arXiv:2509.14745* (2025).

[17] Fengcai Wen, Csaba Nagy, Gabriele Bavota, and Michele Lanza. 2019. A large-scale empirical study on code-comment inconsistencies. In *Proceedings of the 27th International Conference on Program Comprehension* (Montreal, Quebec, Canada) *(ICPC '19)*. IEEE Press, 53–64. doi:10.1109/ICPC.2019.00019

[18] Chunqiu Steven Xia, Yuxiang Wei, and Lingming Zhang. 2023. Automated Program Repair in the Era of Large Pre-Trained Language Models. In *Proceedings of the 45th International Conference on Software Engineering* (Melbourne, Victoria, Australia) *(ICSE '23)*. IEEE Press, 1482–1494. doi:10.1109/ICSE48619.2023.00129