

AI Policy, Disclosure, and Human in the Loop: How Are Contribution Guidelines Adapting to GenAI?

Andre Hora

Department of Computer Science, UFMG
Belo Horizonte, Brazil
andrehora@dcc.ufmg.br

Romain Robbes

Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI
Bordeaux, France
romain.robbes@labri.fr

Abstract—Generative AI (GenAI) has recently transformed software development. Due to the ease of generating code, open source projects are experiencing a growth in contributions. To address the rise of GenAI, open source projects have begun implementing policies for AI usage in contributions. However, the extent to which open source specifies whether AI-assisted contributions are allowed or prohibited, along with the best practices for contributors, remains unclear. This paper provides an initial empirical study to explore how open source projects are adapting to GenAI contributions. We analyzed 1,000 popular GitHub repositories and identified 118 AI policies for contributors. Our results show that (1) 78% of the AI policies allow AI-assisted contributions, while 22% explicitly discourage AI use. (2) 51% of the AI policies require the disclosure of AI-assisted contributions; and (3) 74% of the AI policies require a human in the loop during contribution. Overall, we find that the majority of the analyzed AI policies are positive regarding the usage of GenAI. However, AI disclosure and human in the loop are fundamental in the contribution process. Finally, we conclude by discussing implications for developers and researchers.

Index Terms—AI policy, Generative AI, Contribution guidelines, Software maintenance, AI slop

I. INTRODUCTION

Generative AI (GenAI) has transformed how software is designed, developed, and engineered [1], [2]. Coding agents such as Claude Code, Cursor, and Codex have seen rapid recent adoption, operating with a high degree of *autonomy*; for example, they can invoke external tools, execute code, and complete end-to-end development tasks [3]. A recent large-scale study estimates that coding agent adoption on GitHub is close to 30% [4] and 76% in new projects [5].

Due to the ease of generating code, open source projects are experiencing a growth in contributions [6]–[9]. The volume contributions partially or entirely AI-generated have outpaced the maintenance capacity of many projects, creating debates within the open source community [8], [10]–[12]. As a result, maintainers often lack sufficient capacity to review submissions effectively [8]. AI-assisted contributions may also lack quality—a phenomenon known as AI slop, which refers to low-quality content produced at scale using AI [9], [13], [14].

To address the rise of generative AI, open source projects have begun implementing policies for AI usage in contributions. Several open source projects have established AI policies, including Linux [15], Fedora [16], and LLVM [17]. Figure 1 presents the AI policy of the Ghostty project, which

AI Usage Policy

The Ghostty project has strict rules for AI usage:

- All AI usage in any form must be disclosed. You must state the tool you used (e.g. Claude Code, Cursor, Amp) along with the extent that the work was AI-assisted.
- The human-in-the-loop must fully understand all code. If you can't explain what your changes do and how they interact with the greater system without the aid of AI tools, do not contribute to this project.
- Issues and discussions can use AI assistance but must have a full human-in-the-loop. This means that any content generated with AI must have been reviewed and edited by a human before submission. AI is very good at being overly verbose and including noise that distracts from the main point. Humans must do their research and trim this down.

Fig. 1: AI policy of project ghostty-org/ghostty.

states that AI-generated contributions are allowed provided they are disclosed and include human involvement in the contribution process [18]. FastAPI contribution guidelines state that low-effort or low-quality pull requests are not accepted, and that AI-generated submissions may be closed and accounts blocked [19]. Some projects are more welcoming toward generative AI. For instance, the AI policy of huggingface/transformers states that “*AI-assisted contributions are welcome*” [20]. In contrast, other projects are more restrictive, for example, the 9001/copyparty policy states: “*Do not use AI / LLM when writing code*” [21].

However, the extent to which open source projects specify whether AI-assisted contributions are allowed or prohibited, along with the best practices contributors are expected to follow, remains unclear.

This short paper provides an initial empirical study to explore how open source projects are adapting to generative AI contributions. We analyzed 1,000 popular GitHub repositories and identified 118 AI policies for contributors. We propose three research questions to address AI policy, AI disclosure, and human involvement in contributions:

- **RQ1. How do open source projects address the use of generative AI in contributions?** Overall, 78% of the AI policies allow AI-assisted contributions, while 22% explicitly discourage AI use.
- **RQ2. How do open source projects address disclosure of AI-assisted contributions?** 51% of the AI policies require the disclosure of AI-assisted contributions.
- **RQ3. How do open source projects address human involvement in AI-assisted contributions?** 74% of the AI policies require a human in the loop during contribution.

Overall, we find that the majority of the analyzed AI policies are positive regarding the usage of generative AI. However, AI disclosure and human in the loop are fundamental in the contribution process. AI disclosure is important to support maintainers during the review process, enabling them to verify AI-generated content more effectively. The human in the loop reinforces the idea that contributors remain responsible for the code they submit. Based on our findings, we discuss implications for practitioners and researchers, including the lack of clear guidance in AI policies, practices for AI disclosure, and the growing concern around AI slop. Dataset availability: [22].

II. STUDY DESIGN

A. Initial Set of Repositories

Our goal is to analyze real-world, actively maintained repositories hosted on GitHub. To this end, we rely on the SEART GitHub Search Engine (seart-ghs), a tool that allows researchers to sample repositories to use for empirical studies by using multiple combinations of selection criteria [23]. This tool maintains metadata for all GitHub repositories with at least ten stars. Based on seart-ghs, we selected the top 1,000 repositories with the most stars that meet the following criteria: at least 100 commits, not being forks, and having at least one commit in 2026 (the star metric is primarily adopted in the software mining literature as a proxy of popularity [24]).

B. Detecting Repositories with AI Policy

We define an AI policy as any set of rules governing AI usage for contributors. From the initial set of repositories, we identified those containing AI policies using two complementary methods: (1) Contribution guideline files containing AI policies and (2) Explicit AI policy files.

1. Contribution guideline files containing AI policies.

We verified the presence of contribution guideline files CONTRIBUTING.md and DEVELOPING.md with AI policies [25]. Specifically, we looked for generative AI-related terms (*LLM, generative AI, GenAI, AI, AI agents*) as signs of AI policy. In addition, when performing this analysis, we observed that some repositories use GitHub guideline files primarily to redirect contributors to external documentation. For instance, in FastAPI, the CONTRIBUTING.md [26] file points to a separate contributing documentation. To find such cases, we also analyzed external contributing documentation whenever the GitHub guideline files included a single link, checking for the presence of generative AI-related terms in those sources as well. In total, we identified 752 repositories with contribution guideline files, of which 158 included generative AI-related terms. We then manually reviewed these 158 cases and found 45 false positives (e.g., AI/LLM-focused projects, benchmarks, and datasets) and 113 true positives, i.e., repositories that actually include AI policies for contributors.

2. Explicit AI policy files. We also checked for the presence of the standard AI policy file AI_POLICY.md. In addition, we included two other AI policy files: LLM_POLICY.md and AI_USAGE_POLICY.md. We found 13 repositories with explicit AI policy files, such as ghostty-org/ghostty.

In total, we detected 118 repositories with AI policies for contributors. On the median, these repositories have 40K stars, 9.5K commits, and 373 contributors. **Our dataset is publicly available [22].**

C. Research Questions

We propose three research questions addressing AI policies (RQ1), AI disclosure (RQ2), and human involvement (RQ3). The rationale is that these requirements are key factors in guiding contributors throughout the contribution process. Ideally, open source projects should clearly state whether they accept AI-assisted contributions. When such contributions are allowed, projects should also provide additional guidance, including whether AI usage must be disclosed and the expected level of human involvement in the contribution process. To our knowledge, there is still a limited understanding of how these factors are addressed in practice.

III. RESULTS

Table I presents an overview of the results. Overall, we identified 118 repositories containing AI policies. Among them, 51% explicitly welcome the use of generative AI in contributions, while 27% permit its use without explicitly encouraging it, totaling 78% of repositories that allow generative AI contributions. In contrast, 22% explicitly discourage the use of generative AI in contributions.

Moreover, 51% of the repositories require disclosure of AI-assisted contributions, and 74% require human oversight in the contribution process (i.e., human in the loop). Next, we discuss each category and provide examples for each of them.

TABLE I: Overview of the results.

Data	#	%
AI Policy	118	100%
• AI is welcome	60	51%
• AI is permitted	32	27%
• AI is discouraged	26	22%
AI Disclosure	60	51%
Human in the Loop	88	74%

A. RQ1: AI Policy

AI is Welcome. We found that 51% (60 out of 118) of the AI policies explicitly welcome the use of generative AI in contributions. For example, the AI policy of github/spec-kit states: “*We welcome and encourage the use of AI tools to help improve Spec Kit! Many valuable contributions have been enhanced with AI assistance for code generation, issue detection, and feature definition*” [27]. The AI policy of project ohmyzsh/ohmyzsh states: “*AI-assisted contributions: We’ll admit it: AI tools can be pretty helpful for coding tasks, and we’re not here to gatekeep how you get your work done*” [28]. The AI policy of project huggingface/transformers mentions: “*AI-assisted contributions are welcome, but they must be coordinated, scoped, and verified to keep review load manageable*” [20]. Similarly, in project microsoft/typescript,

TABLE II: Examples of AI policy (RQ1).

Repository	Message
• AI is Welcome	
storybookjs/storybook	The team welcomes the use of AI as a personal assistant when contributing to Storybook. However, we strongly believe that a real person must be behind every issue and pull request.
astral-sh/uv	We support using AI (i.e., LLMs) as tools for coding. However, you remain responsible for any code you publish and we are responsible for any code we merge and release. We hold a high bar for all contributions to our projects.
pytorch/pytorch	PyTorch encourages the use of AI in its development, however, PyTorch is a large and technically complex project and it is easy for current LLMs, if not properly guided, to produce seemingly correct PRs or issues with major flaws.
fastapi/fastapi	You are encouraged to use all the tools you want to do your work and contribute as efficiently as possible, this includes AI (LLM) tools, etc. Nevertheless, contributions should have meaningful human intervention, judgement, context, etc.
openclaw/openclaw	AI PRs are first-class citizens here. We just want transparency so reviewers know what to look for.
• AI is Permitted	
freqtrade/freqtrade	If you're using AI for your PR, please both mention it in the PR description
owasp/cheatsheetseries	If it is genuinely helpful to use generative AI then it must be declared in any pull request; failure to do so can result in the contribution being closed or deleted.
biomejs/biome	If you are using any kind of AI assistance to contribute to Biome, it must be disclosed in the pull request.
denoland/deno	AI-assisted contributions: If you use AI tools to help write your contribution, you must disclose this in your PR description. There is no penalty for using AI tools [...]
JuliaLang/julia	If your pull request contains substantial contributions from a generative AI tool, please disclose so with details
• AI is Discouraged	
immich-app/immich	Use of generative AI: We ask you not to open PRs generated with an LLM. We find that code generated like this tends to need a large amount of back-and-forth, which is a very inefficient use of our time.
teamnewpipe/newpipe	Using generative AI to develop new features or making larger code changes is generally prohibited.
aseprite/aseprite	We're not going to merge code generated by AI or co-authored by AI.
typst/typst	Implement your change. Do not vibecode the change! Contributions that were implemented by an AI model will not be accepted.
codemirror/codemirror5	Code written by AI language models (either partially or fully) is not welcome.

the AI policy states: “It is acceptable to use AI tools to assist in developing PRs. However, we ask that you disclose this in the PR description” [29].

Note that welcoming the use of generative AI in contributions does not imply that it should be used blindly. On the contrary, most welcoming projects also emphasize the importance of AI disclosure and human review throughout the contribution process. Both aspects (AI disclosure and human in the loop practices) are further explored in RQs 2 and 3.

AI is Permitted. We detected that 27% (32 out of 118) of the AI policies permit the usage of generative AI, but without explicitly encouraging or forbidding. For example, project SillyTavern/SillyTavern is neutral regarding the AI usage: “We do not prohibit nor encourage the use of AI tools for coding assistance to help you write code, documentation, etc” [30]. Similarly, the AI policy of freqtrade/freqtrade simply mentions: “If you’re using AI for your PR, please both mention it in the PR description” [31]. Other projects are concerned with substantial usage of AI. For instance, the AI policy of gohugoio/hugo states that “If a substantial part of your contribution is autogenerated with AI, this must be disclosed in the pull request” [32].

AI is Discouraged. We found that 22% (26 out of 118) of the AI policies explicitly discourage the use of generative AI. In mastodon/mastodon, the AI policy states: “We generally do not encourage AI-assisted contributions; we have adopted this policy to reinforce the importance of genuine human

interaction in everything we work on” [33]. Other projects are more direct regarding the prohibition. Project 9001/copyparty mentions: “Do not use AI / LLM when writing code: copyparty is 100% organic, free-range, human-written software!” [21]. Project PostgREST/postgrest states: “It is expressly forbidden to contribute [...] any content that has been created with the assistance of [...] artificial intelligence tools” [34]. Table II presents complementary examples for all categories.

Finding 1: Overall, 78% of the AI policies allow AI-assisted contributions, while 22% discourage AI use.

B. RQ2: AI Disclosure

We found that 51% (60 out of 118) of the AI policies require disclosure of AI-assisted contributions. The rationale is to support maintainers during the review process, enabling them to verify AI-generated content more effectively. Projects recommend disclosing the AI tool used, the extent of AI assistance, and the adoption of specific commit message trailers, such as Co-Authored-By: and Assisted-By:.

In project electron/electron, the AI policy states: “We encourage disclosure of AI tool assistance in code contributions. This practice helps facilitate productive code reviews, and is not used to police tool usage. When AI tools meaningfully assist in your contribution, note it in the commit message with a trailer: Assisted-By:” [35]. In the project eslint/eslint,

TABLE III: Examples of AI disclosure (RQ2).

Repository	Message
• Disclosure is required	
electron/electron	We encourage disclosure of AI tool assistance in code contributions. This practice helps facilitate productive code reviews, and is not used to police tool usage. [...], note it in the commit message with a trailer: <i>Assisted-By:</i> .
eslint/eslint	Disclosure: If you use AI to generate an issue or a pull request, you must clearly disclose this in your submission. This helps maintainers understand the context of the contribution and perform appropriate reviews.
envoyproxy/envoy	You are transparent about your AI usage. It is often helpful to a reviewer to know that an AI tool was used; please include that information in the PR description.
github/spec-kit	If you are using any kind of AI assistance to contribute to Spec Kit, it must be disclosed in the pull request or issue.
pandas-dev/pandas	You must disclose that you used an automated tool in the contribution.
• Disclosure is not required	
clickhouse/clickhouse	You don't have to disclose your usage of AI. You can tell about it, share your experience, and show the methods, but it is not required. AI is a normal developer's tool, similar to an IDE, an OS, or a keyboard.
cypress-io/cypress	We do not restrict or require disclosure of AI usage.
block/goose	There's no need to tell us you used AI in your work. You are contributing to an agent; it would be odd if you had not.
zeroclaw-labs/zeroclaw	We do not require PRs to declare an AI-vs-human line ratio.

TABLE IV: Examples of Human in the Loop (RQ3).

Repository	Message
ghostty-org/ghostty	The human-in-the-loop must fully understand all code. If you can't explain what your changes do and how they interact with the greater system without the aid of AI tools, do not contribute to this project.
storybookjs/storybook	We strongly believe that a real person must be behind every issue and pull request. All issues and pull requests must be opened by a real person using the official templates.
facebook/docusaurus	Be accountable: You are responsible for the code you submit, regardless of whether it was generated by AI or written by you. You should be able to explain every line of the code, ensure all tests pass, and address our reviews.
keras-team/keras	The ultimate responsibility for any code contributed to Keras rests entirely with the human author .
streamlit/streamlit	AI can help contributors move faster, but it does not replace author ownership . If you open a PR, you are responsible for the correctness, scope, testing, and maintainability of that change.

the AI policy also adopts a similar recommendation: “*Disclosure: If you use AI to generate an issue or a pull request, you must clearly disclose this in your submission. This helps maintainers understand the context of the contribution and perform appropriate reviews*” [36]. Project envoyproxy/envoy mentions: “*You are transparent about your AI usage. It is often helpful to a reviewer to know that an AI tool was used; please include that information in the PR description*” [37].

We also found four cases in which disclosure is not required. For example, in project clickhouse/clickhouse, the AI policy states: “*You don't have to disclose your usage of AI. You can tell about it, share your experience, and show the methods, but it is not required. AI is a normal developer's tool, similar to an IDE, an OS, or a keyboard*” [38]. Table III complements by presenting other examples.

Finding 2: 51% of the AI policies require the disclosure of AI-generated contributions.

C. RQ3: Human in the Loop

We detected that 74% (88 out of 118) require human oversight in the contribution process (i.e., human in the loop). The rationale is that contributors should remain responsible for the code they submit by reviewing, understanding,

and being able to explain their contributions. The seleniumhq/selenium AI policy states: “*Human-in-the-loop is required: You are the author. You must read, review, and understand all AI-assisted output before requesting review*” [39]. Project run-llama/llama_index highlights: “*Accountability: we require human oversight for every contribution, and we hold human developers accountable for their changes*” [40]. Project sipeed/picoclaw mentions the responsibility: “*You Are Responsible for What You Submit: Using AI to generate code does not reduce your responsibility as the contributor*” [41]. The human in the loop requirement also aims to exclude contributions produced entirely by autonomous coding agents. For example, project astral-sh/uv states: “*Due to the foundational nature of our projects, we require a human in the loop who understands the work produced by AI. We do not allow autonomous agents*” [42]. Other examples are presented in Table IV.

Finding 3: 74% of the AI policies require a human in the loop during the contribution process.

IV. DISCUSSION AND IMPLICATIONS

A. AI policies often lack clear guidance

AI policies are becoming a reality in open source projects. We found that 51% of the analyzed AI policies require AI

disclosure, and human involvement is explicitly required in 74%. However, this also means that many AI policies do not clearly specify whether AI usage should be disclosed, and leave unclear whether contributors are expected to review and take responsibility for AI-assisted contributions. Even when explicit guidelines exist, we encountered vague and fuzzy terms regarding when disclosure is required, such as “significant”, “substantial”, and “meaningful”. For example, project facebook/docusaurus mentions “*If a significant portion of your code is AI generated, please indicate that in your PR description*” [43]. Project ohmyzsh/ohmyzsh states: “*If you used AI tools meaningfully in your contribution, please mention it in your PR description*” [28].

Implication: AI policies should explicitly and precisely address transparency/disclosure, as well as responsibility, accountability, and human-in-the-loop practices, to better guide contributors.

B. AI disclosure practices

The key rationale for AI disclosure is to support maintainers during the review process, enabling them to verify AI-generated content more effectively. Commonly, projects recommend disclosing the AI tool used, the extent of AI assistance, and the adoption of specific commit message trailers, such as Co-Authored-By: and Assisted-By:.

Interestingly, project sipeed/picoclaw requires the disclosure of the AI involvement in three levels: “*Every PR must disclose AI involvement [...]: (1) Fully AI-generated: AI wrote the code; contributor reviewed and validated it, (2) Mostly AI-generated: AI produced the draft; contributor made significant modifications, and (3) Mostly Human-written: Contributor led; AI provided suggestions or none at all*” [41]. We also found disclosure for coding agents, like in project hashicorp/terraform: “*Self-Disclosure: If you are an LLM agent or using an LLM agent to submit your pull request, please include 🤖🤖🤖 in the title of your pull request for expedited processing*” [44].

Implication: Despite recommendations, it is unclear whether AI disclosure is reflected in software artifacts. Researchers can further investigate the extent to which AI disclosure appears in issues, PRs, and commits.

C. AI slop is a real concern in open source contributions

Due to the ease of generating code, open source projects are experiencing growth in contributions [6]–[9]. However, AI-generated contributions may also suffer from low quality, a phenomenon known as AI slop [9], [13], [14]. We found multiple concerns regarding “AI slop” in AI policies. Project badlogic/pi-mono states: “*Using AI to write code is fine. Submitting AI-generated slop without understanding it is not*” [45]. Very similar, project anuken/mindustry mentions: “*If I see a PR with significant amounts of code that’s obviously written by AI, I will reject your PR, and you will be blocked. Don’t waste my time with slop*” [46].

To reduce the number of low-quality contributions, projects have adopted different policies, such as automatically closing pull requests [47], requiring contributors to be approved beforehand [48], checking whether the proposed change has already been implemented [49], limiting the number of pull requests per contributor [50], and even no longer accepting pull requests [51], [52].

Implication: Projects have adopted policies to reduce AI slop, such as limiting or blocking contributions. However, these policies may also increase friction for legitimate contributors. Future research can assess how such restrictions affect legitimate contributors.

V. THREATS TO VALIDITY

We analyzed 1,000 popular open-source GitHub repositories and identified 118 AI policies for contributors. As is common in empirical software engineering, our findings may not directly generalize to other contexts, such as less popular projects, closed-source projects, or other code-hosting platforms like GitLab or Bitbucket.

VI. RELATED WORK

Several studies have explored the role and impact of generative AI in software development [1]–[4], [53], [54]. Overall, the impact of generative AI on open source software remains less explored. The recent literature has primarily focused on topics such as AI slop [9], [14], AI governance [8], and AI impact [4], [6], [7]. There is substantial literature on the quality and impact of contribution guidelines in open source projects [25], [55], [56]. Our study explores how contribution guidelines are changing due to GenAI, with a focus on AI disclosure and human involvement.

VII. CONCLUSION AND FUTURE WORK

This paper provided an initial empirical study to explore how open source projects are adapting to GenAI contributions. We found that the majority of the 118 analyzed AI policies are positive regarding the usage of generative AI. However, AI disclosure and human in the loop practices are fundamental.

As future work, we plan to explore other perspectives of open source software in the context of generative AI, including the use of fully autonomous agents for contributions, concerns related to licensing and legal requirements, and issues related to AI-generated communication.

ACKNOWLEDGMENTS

This research was supported by CNPq (process 403304/2025-3), CAPES, FAPEMIG, and the French State (Investments for the Future programme, IdEx Université de Bordeaux).

REFERENCES

- [1] A. Fan, B. Gokkaya, M. Harman, M. Lyubarskiy, S. Sengupta, S. Yoo, and J. M. Zhang, “Large language models for software engineering: Survey and open problems,” in *International Conference on Software Engineering: Future of Software Engineering*. IEEE, 2023, pp. 31–53.
- [2] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, and H. Wang, “Large language models for software engineering: A systematic literature review,” *ACM Transactions on Software Engineering and Methodology*, 2023.
- [3] R. Robbes, T. Matricon, T. Degueule, A. Hora, and S. Zacchiroli, “Promises, perils, and (timely) heuristics for mining coding agent activity,” in *International Conference on Mining Software Repositories*, 2026.
- [4] —, “Agentic Much? Adoption of Coding Agents on GitHub,” *ACM Transactions on Software Engineering and Methodology*, 2026.
- [5] —, “Agentic very much! adoption of coding agent in new github projects,” *arXiv preprint arXiv:2606.07448*, 2026.
- [6] F. Song, A. Agarwal, and W. Wen, “The impact of generative ai on collaborative open-source software development: Evidence from github copilot,” *arXiv preprint arXiv:2410.02091*, 2024.
- [7] S. Nakashima, Y. Ishimoto, M. Kondo, S. McIntosh, and Y. Kamei, “Why agentic-prs get rejected: A comparative study of coding agents,” *arXiv preprint arXiv:2602.04226*, 2026.
- [8] W. Yang, R. He, and M. Zhou, “Beyond Banning AI: A First Look at GenAI Governance in Open Source Software Communities,” *arXiv preprint arXiv:2603.26487*, 2026.
- [9] S. Baltes, M. Cheong, and C. Treude, ““An Endless Stream of AI Slop”: The Growing Burden of AI-Assisted Software Development,” *arXiv preprint arXiv:2603.27249*, 2026.
- [10] LLVM: Our AI policy vs code of conduct and vs reality, <https://discourse.llvm.org/t/our-ai-policy-vs-code-of-conduct-and-vs-reality/88300>.
- [11] LLVM AI tool policy: start small, no slop, <https://discourse.llvm.org/t/rfc-llvm-ai-tool-policy-start-small-no-slop/88476>.
- [12] Policy: Empower reviewers to reject burdensome PRs, <https://github.com/rust-lang/compiler-team/issues/893>.
- [13] Slop is the new name for unwanted AI-generated content , <https://simonwillison.net/2024/May/8/slop>.
- [14] S. Baltes, M. Cheong, and C. Treude, “AI Slop and the Software Commons,” *arXiv preprint arXiv:2604.16754*, 2026.
- [15] Linux: AI Coding Assistants, <https://docs.kernel.org/process/coding-assistants.html>.
- [16] Fedora AI-Assisted Contributions Policy, <https://docs.fedoraproject.org/en-US/council/policy/ai-contribution-policy>.
- [17] LLVM AI Tool Use Policy, <https://llvm.org/docs/AIToolPolicy.html>.
- [18] AI policy of ghostty-org/ghostty, https://github.com/ghostty-org/ghostty/blob/55a3e33ab26a23d75b274b23c7f76d837db00578/AI_POLICY.md.
- [19] AI policy of fastapi-fastapi, <https://fastapi.tiangolo.com/contributing/>.
- [20] AI policy of huggingface/transformers, <https://github.com/huggingface/transformers/blob/c472755e79aac54d675845b1f5e5c821c21260af/CONTRIBUTING.md>.
- [21] AI policy of 9001/copyparty, <https://github.com/9001/copyparty/blob/6e25d648a900f65a4546a1b17a9761c0f1e9e3cb/CONTRIBUTING.md>.
- [22] AI policy dataset, <https://doi.org/10.5281/zenodo.20214334>.
- [23] O. Dabic, E. Aghajani, and G. Bavota, “Sampling Projects in GitHub for MSR Studies,” in *International Conference on Mining Software Repositories*. IEEE, 2021, pp. 560–564.
- [24] H. Borges, A. Hora, and M. T. Valente, “Understanding the factors that impact the popularity of GitHub repositories,” in *International Conference on Software Maintenance and Evolution*, 2016, pp. 334–344.
- [25] B. Falcucci, F. Gomide, and A. Hora, “What do contribution guidelines say about software testing?” in *International Conference on Mining Software Repositories*. IEEE, 2025, pp. 434–438.
- [26] FastAPI CONTRIBUTING.md, <https://github.com/fastapi/fastapi/blob/34a82b1b874f216db5b8bacb1f5c7191055e1fa2/CONTRIBUTING.md>.
- [27] AI policy of github/spec-kit, <https://github.com/github/spec-kit/blob/171b65ac33a3bf51c23b9f7a5287032ed1ae72ba/CONTRIBUTING.md>.
- [28] AI policy of ohmyzsh/ohmyzsh, <https://github.com/ohmyzsh/ohmyzsh/blob/349b9e49ced7682e27927ffb34b6522f011f3e74/CONTRIBUTING.md>.
- [29] AI policy of microsoft/typescript, <https://github.com/microsoft/typescript/blob/55423abe4d029017f19b6e4c32097591994836b4/CONTRIBUTING.md>.
- [30] AI policy of SillyTavern/SillyTavern, <https://github.com/SillyTavern/SillyTavern/blob/004f1336e6e59d476c1043f1dc94c92d028ac5d0/CONTRIBUTING.md>.
- [31] AI policy of freqtrade/freqtrade, <https://github.com/freqtrade/freqtrade/blob/af1de46cd4dd968482dee15a5deebd57005d8691/CONTRIBUTING.md>.
- [32] AI policy of gohugoio/hugo, <https://github.com/gohugoio/hugo/blob/90d8bf34aea897a8a329480bde54ff1c61c0c9b3/CONTRIBUTING.md>.
- [33] AI policy of mastodon/mastodon, https://github.com/mastodon/.github/blob/main/AI_POLICY.md.
- [34] AI policy of PostgREST/postgrest, <https://github.com/PostgREST/postgrest/blob/c9253ed0569b24c925cc4e99b1453f94dc1a80ef/CONTRIBUTING.md>.
- [35] AI policy of electron/electron, <https://github.com/electron/governance/blob/main/policy/ai.md>.
- [36] AI policy of eslint/eslint, <https://github.com/eslint/eslint/blob/54080dad4f77bb39a1a843933d4ff3a2b7c175e2/CONTRIBUTING.md>.
- [37] AI policy of envoyproxy/envoy, <https://github.com/envoyproxy/envoy/blob/fe746515871f06ba5b811714c9eeeca8eaf582654/CONTRIBUTING.md>.
- [38] AI policy of ClickHouse/ClickHouse, https://github.com/ClickHouse/ClickHouse/blob/cf81ad1ddee2d5376f28f33bd4afe750c850319/AI_POLICY.md.
- [39] AI policy of seleniumhq/selenium, <https://github.com/seleniumhq/selenium/blob/1ab572a13ddf5e47e8a0b82da3e37557270df391/CONTRIBUTING.md>.
- [40] AI policy of run-llama/llama_index, https://github.com/run-llama/llama_index/blob/0a6c90bfd610dcc66dc89ed3e1d905c5e9b6dc/CONTRIBUTING.md.
- [41] AI policy of sipeed/picoclaw, <https://github.com/sipeed/picoclaw/blob/8d51d306b32ae2024e02b7a97c9d051b3d8b25be/CONTRIBUTING.md>.
- [42] AI policy of astral-sh/uv, https://github.com/astral-sh/.github/blob/main/AI_POLICY.md.
- [43] AI policy of facebook/docusaurus, <https://github.com/facebook/docusaurus/blob/9929ac0f6bc13c9ffbc6aa2cca419ff62234d3c6/CONTRIBUTING.md>.
- [44] AI policy of hashicorp/terraform, <https://github.com/hashicorp/terraform/blob/844f216569901c0f8142136e9e47fe62e336b9ca/.github/CONTRIBUTING.md>.
- [45] AI policy of badlogic/pi-mono, <https://github.com/badlogic/pi-mono/blob/0b271a2c4f539c24d33a6f3e8f8e9e7abe625823/CONTRIBUTING.md>.
- [46] AI policy of anuken/mindustry, <https://github.com/anuken/mindustry/blob/327464ba82fca614671d86b1357c99f182783493/CONTRIBUTING.md>.
- [47] Auto-closing pull requests, <https://github.com/earendil-works/pi/issues/4390>.
- [48] Approved contributors, <https://github.com/earendil-works/pi/pull/4409>.
- [49] Checking implementation, <https://github.com/openclaw/openclaw/pull/80637>.
- [50] Limiting contributors, <https://github.com/openclaw/openclaw/issues/38283>.
- [51] AI policy of jesseduffield/lazygit, <https://github.com/jesseduffield/lazygit/blob/0ecf818a739606d4255e009b66309e6e508ce08a/CONTRIBUTING.md>.
- [52] AI policy of pocketbase/pocketbase, <https://github.com/pocketbase/pocketbase/blob/83e44a7cfbc7312c3f5e3d2f643da45512fcbc52/CONTRIBUTING.md>.
- [53] A. Hora and R. Robbes, “Are coding agents generating over-mocked tests? an empirical study,” in *International Conference on Mining Software Repositories*, 2026.
- [54] H. V. F. d. Santos, V. Costa, J. E. Montandon, and M. T. Valente, “Decoding the configuration of ai coding agents: Insights from claude code projects,” in *International Workshop on Agentic Engineering*, 2026, pp. 63–67.
- [55] J. Tsay, L. Dabbish, and J. Herbsleb, “Influence of social and technical factors for evaluating contribution in github,” in *International Conference on Software Engineering*, 2014, pp. 356–366.
- [56] O. Elazhary, M.-A. Storey, N. Ernst, and A. Zaidman, “Do as I do, not as I say: Do contribution guidelines match the GitHub contribution process?” in *International Conference on Software Maintenance and Evolution*, 2019, pp. 286–290.