Observing bots in the wild: A quantitative analysis of a large open source ecosystem

Natarajan Chidambaram
Software Engineering Lab
University of Mons
Mons, Belgium
natarajan.chidambaram@umons.ac.be

Tom Mens
Software Engineering Lab
University of Mons
Mons, Belgium
tom.mens@umons.ac.be

***Abstract*—The GitHub platform allows repository maintainers and contributors to automate their activities either through GitHub Actions workflows or by making use of bots. The second category concerns automated actors that interact with regular users in GitHub repositories, and includes GitHub's internal automation services (e.g., Dependabot), GitHub Apps, and bot accounts. There has been little to no empirical research on the presence and use of such bots in large software ecosystems. Also, no former studies have compared how bot accounts and GitHub Apps are used within and across GitHub repositories. This paper addresses this gap, through an empirical analysis of the use of bot accounts and GitHub Apps in *NumFOCUS*, a large open source software ecosystem for data science. We analyse, over a three-month period, the activity sequences of 853 contributors across 59 GitHub organisations hosting 1,169 repositories. Using state-of-the-art bot identification approaches we identify activities of 802 humans, 34 bot accounts, 13 GitHub Apps and 4 internal automation services. Based on this dataset we reveal behavioural differences in *NumFOCUS* between bots and humans on the one hand, and between different bot categories on the other hand.**

***Index Terms*—GitHub App, bot account, software ecosystem, collaborative development, software community**

I. Introduction

The GitHub collaborative software development platform has become omnipresent for open source software (OSS) projects in the last decades [1]. OSS contributors tend to perform a wide range activities in the GitHub repositories they are contributing to, such as committing and reviewing code, managing issues and pull requests (PR), publishing releases, creating and deleting branches and tags, and so on.

As some of these activities are repetitive or error-prone, developers tend to resort to one or more automation mechanisms: (1) GitHub's internal automation services such as Dependabot for dependency management and security scanning; (2) GitHub Apps in repositories that extend GitHub's functionality or provide an interface with third-party services; (3) bot accounts (a.k.a. machine users) that act on behalf of regular users to automate some of their activities; (4) GitHub Actions workflow configurations that are predominantly used for CI/CD automation.

The GitHub Actions mechanism is out of scope of the current paper since it has been investigated in earlier work [2], [3], [4], [5]. This paper therefore focuses on the first three automation mechanisms, which we collectively refer to as **bots**, and aims to study and compare their activities in GitHub organisations and repositories to regular human accounts. Ignoring the presence of bots in software repositories could bias the conclusions of socio-technical empirical analyses [6], or could lead funding organisations or sponsors of OSS projects to incorrectly accredit bots as top human contributors [6], [7].

Past research has primarily focused on identifying bot accounts [8], [9], [10], [11], [12]. There has been much less research on how the automation mechanisms in GitHub are used "in the wild", especially in the context of large software ecosystems composed of a large community of contributors collaborating within and across hundreds of GitHub repositories. Conducting such studies helps to understand the roles and dynamics of bots in such large ecosystems.

This paper therefore presents a first-of-its-kind quantitative observation of how bot mechanisms are used in a large OSS ecosystem hosted on GitHub. We selected as case study *NumFOCUS*[1]. At the time of writing, this non-profit organisation sponsored 50+ OSS projects for data science, including very popular ones such as *NumPy*, *Pandas*, *Matplotlib* and *Conda*.

To analyse its community of contributors, we rely on the public GitHub events made by contributors having participated in repositories belonging to the GitHub organisations of *NumFOCUS* projects. During three months, we observed their GitHub events to quantify the specific activity types of contributors (e.g., creating or deleting tags or branches, publishing releases, opening, closing or commenting on issues or PRs, reviewing commits or PRs) [13]. We explore and identify differences in activity patterns between the three GitHub bot mechanisms and the human contributors. A replication package can be found on https://doi.org/10.5281/zenodo.14415595.

II. Related work

Golzadeh et al. [10] developed *BoDeGHa*, a bot identification tool based on a Random Forest classifier to identify bots based on their comments in issues and PRs in GitHub repositories. The model used features like similarity and number of patterns in comments, number of empty comments, and string distance between comments. Using the same features, Golzadeh et al. [9] developed *BoDeGiC*, a model and tool to

[1]https://numfocus.org

identify bots based on their git commit messages. Abdellatif et al. [11] proposed *BotHunter*, another bot identification model based on a Random Forest classifier that extends the feature set of *BoDeGHa* with additional information from contributor's profile, bio, and events they are involved in. Chidambaram et al. [12] developed *RABBIT*, an efficient bot identification tool based on a Gradient Boosting classifier that relies uniquely on features derived from contributor activity sequences. Chidambaram [14] performed a detailed comparison of the accuracy and efficiency of these bot identification approaches, revealing that *BotHunter* and *RABBIT* are complementary and outperform the other approaches.

Golzadeh et al. [6] demonstrated the prevalence of bots in GitHub by considering top 20 committers in 10 large and active OSS projects in GitHub. For each project, up to 3 bots belonged to the top 20. Out of the 21 bots they found, 12 were bot accounts and 9 were GitHub Apps. On average, these bots were responsible for nearly one-fifth of all commits in the projects. In a follow-up study, Golzadeh et al. [15] considered top 20 committers in 27 popular GitHub repositories. They found all repositories to use bots, with 18 of them having a bot in the top three committers. Some repositories used 4 different bots, and many of these bots were responsible for most of the activities. Wang et al. [16] considered 1,000 frequently starred repositories in GitHub, and used *BoDeGHa* to detect bots in 613 of them. They grouped these bots into six task categories such as CI assistance, dependency and security analysis, and code review assistance. 60% of the considered projects used at least one bot to automate routine tasks, and 74 out of 201 bots belonged to more than one category. All the above studies were either conducted on a small number or on a random collection of repositories. In contrast, this paper focuses on the usage of bots in a large ecosystem (*NumFOCUS*) where repositories are likely to be interconnected, and hosts project that are developed with a common goal (advancing data science).

III. Data Extraction

To observe the use of bots in *NumFOCUS*, we rely on the public events in all repositories of GitHub organisations associated to *NumFOCUS* projects. From these public events, we identify all involved contributors (bots and humans). We exclude any events made by these contributors in repositories and organisations that do not belong to *NumFOCUS*, as they are considered out of scope. We consider a three-month observation period from July to September 2024.

A. Extracting activity sequences for contributors

We identified 60 GitHub organisations corresponding to *NumFOCUS* projects, and the 1,626 GitHub repositories belonging these organisations. For example, the dataset contains public events for all 12 GitHub repositories for the numpy GitHub organisation corresponding to the NumPy project.[2] We excluded the conda-forge organisation containing tens of thousands of repositories for the packages (called recipes) of the Conda package manager, since we consider it to be a separate packaging ecosystem.

As a data source we used GHArchive [3] which records and archives GitHub's public events. We extracted all 358,451 public events performed by 21,957 contributors in the identified repositories during the observation period. 14,351 contributors (65%) performed only a single event, of which 9,788 corresponded to *starring a repository* and 2,072 corresponded to *forking a repository*. We decided to exclude such peripheral contributors that contribute very little to the ecosystem. To do so, we removed a very long tail of 20,445 contributors (93.1%) involved in less than 10 events in *NumFOCUS* repositories. This resulted in a filtered dataset of 322,317 events (89.9%) performed by 1,512 contributors.

From the obtained events obtained we derived more meaningful higher-level activity types, using the mapping provided by Chidambaram et al. [13]. For example, we record activity types such as *Creating tag* and *Closing PR* from low-level events such as *CreateEvent* and *PullRequestEvent* that do not distinguish between the type of creation (e.g. tag, branch, release) or the type of PR event (e.g., opening, closing). The activity dataset contains 282,921 activities belonging to 23 different activity types, performed by 1,512 contributors in 1,315 repositories of 59 GitHub organisations.

B. Bot detection

Our aim is to explore differences in activities between various bot mechanisms and human contributors participating to *NumFOCUS* organisations and repositories on GitHub. This requires identifying the type of each contributor.

Determining contributors corresponding to GitHub Apps or internal automation services is easy. The GitHub API endpoint for users marks their type as "Bot", and their account name always ends with the [bot] suffix. In this way, we identified 4 internal automation services and 13 GitHub Apps.

Bot accounts are considerably more difficult to identify. The GitHub API marks their type as "User", making them indistinguishable from human accounts. Researchers have therefore proposed heuristics and classification models for bot identification (see Section II). Based on the comparison of [14] we combine the best three to identify bots: a name-based heuristic, *BIMBAS* and *BotHunter*. We started with the name-based heuristic to identify bots, by checking if their lowercased account name contains the string "bot". This resulted in 20 potential bot accounts. We manually excluded false positives (e.g., common human surnames such as "Abbot"), resulting in 19 confirmed bot accounts. We processed the remaining user accounts using *BIMBAS*, a bot identification model based on activity sequences [14]. The model provides, for each account, a prediction confidence score. We retained only those 817 accounts for which *BIMBAS* had at least 70% prediction confidence. We applied *BotHunter* [11] on them to check whether its prediction agreed with *BIMBAS*. In case of disagreement, we manually verified whether the account

[2]https://github.com/orgs/numpy/repositories

[3]https://www.gharchive.org/

should be classified as bot. This allowed us to identify 15 more bot accounts, leading to a total of 34 bot accounts (of which 19 containing "bot" in their name) and 802 human accounts.

Table I provides the characteristics of the final dataset of all 853 considered *NumFOCUS* contributors and their associated activity sequences, grouped by contributor type: 4 internal automation services, 13 GitHub Apps, 34 bot accounts and 802 human accounts. While 94% of all considered contributors are humans (802 out of 853), they account for only 53.4% of all activities (133,173 out of 249,185). This suggests that bots are more active, which is confirmed by their median number of activities, that is considerably higher than for humans.

TABLE I
BREAKDOWN OF CONTRIBUTOR TYPES IN DATASET

	internal services	GitHub Apps	bot accounts	human accounts	**total**
#contributors	4	13	34	802	853
#activities	91,381	4,859	19,772	133,173	249,185
median #activities	1,116	115	104	56	58
#repositories	414	239	234	1,108	1,169
median #repositories	84	5	2	4	4
#organisations	55	46	45	58	59
median #organisations	24.5	4	1	1	1

IV. EMPIRICAL ANALYSIS OF CONTRIBUTOR ACTIVITIES

This section explores two research questions pertaining to the prevalence and difference in behaviour of different types of contributors in the *NumFOCUS* ecosystem.

RQ1: How prevalent are bots in *NumFOCUS* repositories and organisations?

58 of the 59 organisations (98.3%) automate their tasks with bots. These bots are involved in 583 of the 1,169 repositories (49.9%) belonging to these organisations, and they are the most active contributor in 155 of them (26.6%). The median values in Table I reveal that internal automation services and Apps participate in more organisations and repositories than human accounts. In contrast, bot accounts tend to restrict themselves to fewer repositories belonging to a single organisation.

To observe the difference in number of activities, repositories and organisations that contributor types are involved in, we plot the distribution of all contributors (per contributor type) in Fig. 1 with *boxen plots*. We excluded the 4 internal automation services as there are not enough to have a meaningful visualisation.

The distributions are skewed for the number of repositories contributors are involved in, with a median of 5 repositories for Apps, 2 for bot accounts, and 4 for human accounts. The 75^{th} percentile confirms that Apps are involved in considerably more repositories, with a value of 19 compared to only 5.5 for bot accounts and 6 for human accounts. This is likely because Apps are readily available and can be installed easily from the GitHub Marketplace to automate activities in any repository.

The findings for the distributions of the number of organisations are similar to those for repositories, with bot accounts and human accounts being involved in very few organisations (median of 1), and Apps being involved in considerably more organisations (median of 4 and 75^{th} percentile of 10).

Focusing on the 4 internal automation services, we observe that: (1) github-actions[bot], acting on behalf of the GitHub Actions workflows used by a repository, accounted for 89,117 activities in 319 repositories of 45 organisations; (2) dependabot[bot] that automates dependency updates had 1,937 activities in 163 repositories of 47 organisations; (3) github-merge-queue[bot] that provides a merge queue for PRs had 295 activities in 4 repositories of a single organisation; and (4) github-advanced-security[bot] that improves and maintains code quality and security performed 32 activities in 5 repositories of 4 organisations.

Among the 13 GitHub Apps, the three most active ones were: (1) codecov[bot] that reports on code coverage, with 1,683 activities in 88 repositories of 25 organisations; (2) pre-commit-ci[bot] that provides a CI service for pre-commit framework, with 1,509 activities in 133 repositories of 26 organisations; (3) renovate[bot] that automates dependency updates, with 530 activities in 5 repositories of 3 organisations.

Among the 34 bot accounts, the three most active ones were: (1) editorialbot with 13,346 activities in only 4 repositories; (2) bioc-issue-bot with 841 activities in a single repository; (3) conda-bot with 673 activities in 26 repositories. Each of them was active in a single organisation.

Finding: Bots are prevalent in *NumFOCUS*, contributing to 98% of its organisations and 50% of its repositories, and being the most active contributors in 27% of these repositories. Human accounts and bot accounts tend to restrict their activities to few organisations. GitHub Apps tend to be involved in more repositories belonging to more organisations than bot accounts. The most active bots are the two internal automation services github-actions[bot] and dependabot[bot].

RQ2: Which activity types are bots involved in?

While RQ1 focused on the number of repositories and organisations, RQ2 focuses on the number of activities and activity types per contributor type. Table I revealed that bots have a higher median number of activities than human accounts. Fig. 1(a) reveals a skewed distribution of number of activities performed by contributors, grouped by contributor type. We limited the y-axis to 2,500 to aid readability.

For each contributor type there are outliers that are considerably more active than all others. The internal automation service github-actions[bot] was responsible for 89,117 activities, accounting for 97.5% of all activities performed by internal automation services. codecov[bot] and pre-commit-ci[bot] were responsible for 1,683 and 1,509 activities (34.6% and 31.1% of all activities performed by Apps). A single bot account, editorialbot was responsible for 13,346 activities (67.5%) of all activities performed by bot accounts. Two human accounts were involved in 12,321 and 5,206 activities (respectively 9.3% and 3.9% of all human activities).

To determine whether different contributor types tend to be

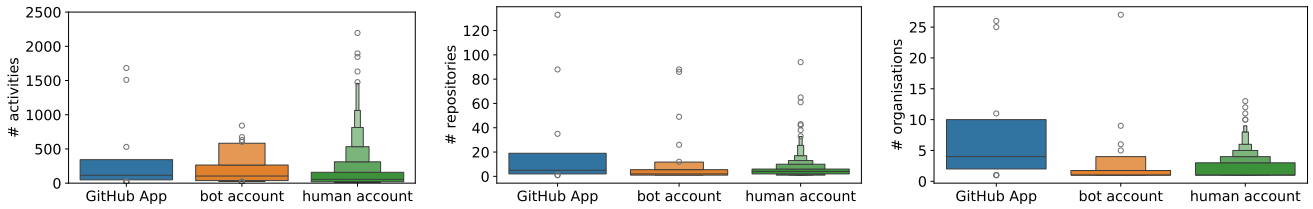

Fig. 1. Boxen plots of number of activities (left), number of repositories (middle) and number of organisations (right) for *NumFOCUS* contributors grouped by contributor type.

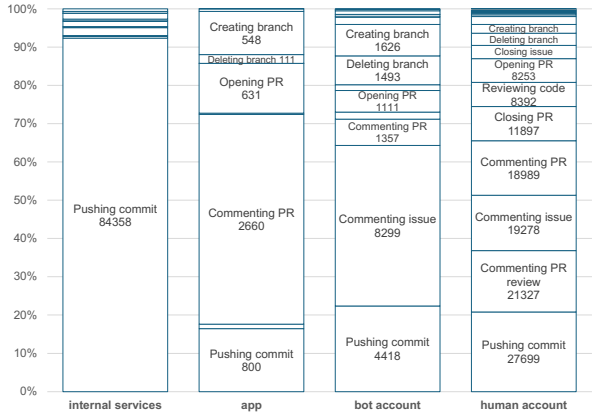

Fig. 2. Proportion of activity types performed per contributor type. Only activity types with a high enough proportion of activities are labeled along with the number of activities of that type.

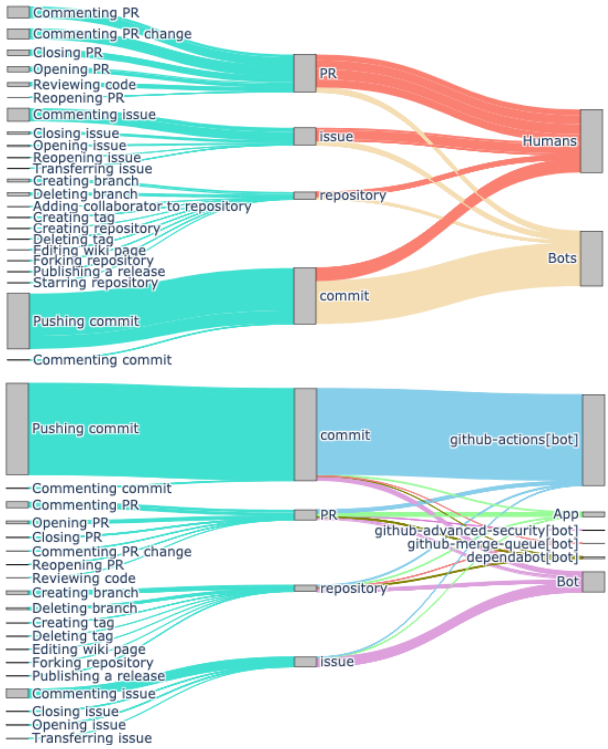

Fig. 3. Sankey diagram of the distribution of activities performed by (top) humans and bot contributors, and (bottom) different internal automation services, Apps and bot accounts.

involved in different types of activities in *NumFOCUS*, (i) we quantify, per contributor type, the proportion of all activities carried out per activity type as visualised in Fig. 2, and (ii) we analyse the difference in proportion of activities per activity type group between humans and bot contributors in Fig. IV and between different internal automation services, Apps and bot accounts in Fig. IV. We grouped activity types into four different groups, namely PR, issue, repository and commit for ease of visualisation and understanding.

The GitHub internal automation services are involved in 15 activity types, with *pushing commits* being the predominant type (92.3%). As observed in Fig. IV, 99.8% of *pushing commits* (84,200 activities) is due to github-actions[bot]. dependabot[bot] is more diverse, being involved in seven activity types, of which 32.6% *creating branch* and 32.3% *opening PR*.

GitHub Apps perform ten activity types. The most frequent type is *commenting PR*, performed by ten Apps. Five Apps exclusively perform this activity. For example, codecov[bot] accounts for 63.3% of all *commenting PR* activity performed by Apps, mainly for posting code coverage analysis reports computed by Codecov. Four Apps are involved in a combination of *creating branch*, *opening PR* and *pushing commits*.

Bot accounts are involved in 16 activity types. They are mostly *commenting issues* (42% by 10 bots) while this type only accounted for 1.2% of all activities for Apps. The secondmost frequent activity for bot accounts was *pushing commits* (22.3% by 13 bots). Both activity types are predominantly performed by editorialbot, accounting for 77.3% of all *commenting issues* and 61.6% of all *pushing commits*. Other major activity types are *creating branch* (8.2%), *deleting branch* (7.6%), *commenting PR* (6.9%), and *opening PR* (5.6%). As observed from Fig. IV, bot accounts majorly perform issue-related activities (45.5%) compared to that of commit- (23.3%), repository- (16.9%) and PR-related activities (14.3%).

Human accounts are involved in all 23 activity types. A single human accounted for 43.9% of all *pushing commits*, an activity carried out by 401 humans. 1,928 human accounts performed *commenting* (on issue, PR or PR review) (44.7% of all human activities). Review-related activities (i.e., *commenting PR review* and *reviewing code*) were virtually absent for bots,

while they account for 22.3% of all human activities. Overall, as observed from Fig. IV, humans majorly perform PR-related activities (52%) compared to that of commit- (20.9%), issues- (20.2%) and repository-related activities (6.9%).

Finding: Bots are more active than humans. Humans are involved in more activity types. Bot accounts are involved in more activity types than Apps. The activity distribution is heavily skewed for all contributor types.

V. Conclusion

While many bot identification tools have seen the light in recent years, there is little large-scale empirical evidence on how bots act "in the wild" in large ecosystems of interrelated software repositories on GitHub. We therefore carried out a case study on the *NumFOCUS* ecosystem for data science.

We analysed 249,185 activities performed by 853 contributors in 1,169 GitHub repositories belonging to 59 GitHub organisations. We identified 51 of these 853 contributors as bots, of which 4 corresponded to GitHub internal automation services, 13 to GitHub Apps, and 34 to bot accounts. Analysing differences in the type and frequency of activities being performed by these contributors, we observed that bots behave differently than humans, and that the three types of bots exhibit different activity patterns. GitHub Apps were also observed to be active in considerably more repositories and organisations than the other contributor types.

These promising preliminary insights call for the need for more in-depth studies on the use and complementarity of the different automation practices provided by GitHub. This may help to uncover the roles and values of such practices during collaborative development, and how their use evolves over time and across organisations and projects. There is also a need to carry out case studies on other large ecosystems, since the observed findings may not generalise beyond *NumFOCUS*.

Acknowledgment

We thank Youness Hourri for extracting the *NumFOCUS* event data from GH Archive. This work is supported by DigitalWallonia4.AI research project ARIAC grant 2010235, and the Fonds de la Recherche Scientifique - FNRS under grants T.0149.22 and J.0147.24.

References

[1] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, "Social coding in GitHub: Transparency and collaboration in an open software repository," in *International Conference on Computer Supported Cooperative Work (CSCW)*. ACM, 2012, pp. 1277–1286.

[2] A. Decan, T. Mens, P. R. Mazrae, and M. Golzadeh, "On the use of GitHub Actions in software development repositories," in *International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2022, pp. 235–245.

[3] M. Golzadeh, A. Decan, and T. Mens, "On the rise and fall of CI services in GitHub," in *International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2022, pp. 662–672.

[4] A. Decan, T. Mens, and H. Onsori Delicheh, "On the outdatedness of workflows in the GitHub Actions ecosystem," *Journal of Systems and Software*, vol. 206, p. 111827, 2023.

[5] P. Rostami Mazrae, T. Mens, M. Golzadeh, and A. Decan, "On the usage, co-usage and migration of CI/CD tools: A qualitative analysis," vol. 28, no. 2, p. 52.

[6] M. Golzadeh, T. Mens, A. Decan, E. Constantinou, and N. Chidambaram, "Recognizing bot activity in collaborative software development," *IEEE Software*, vol. 39, no. 5, pp. 56–61, 2022.

[7] N. Cassee, C. Kitsanelis, E. Constantinou, and A. Serebrenik, "Human, bot or both? A study on the capabilities of classification models on mixed accounts," in *International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2021.

[8] T. Dey, S. Mousavi, E. Ponce, T. Fry, B. Vasilescu, A. Filippova, and A. Mockus, "Detecting and characterizing bots that commit code," in *International Conference on Mining Software Repositories (MSR)*. ACM, 2020, pp. 209–219.

[9] M. Golzadeh, A. Decan, and T. Mens, "Evaluating a bot detection model on git commit messages," in *Belgium-Netherlands Software Evolution Workshop (BENEVOL)*, vol. 2912. CEUR Workshop Proceedings, 2020.

[10] 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," *Journal of Systems and Software*, vol. 175, 2021.

[11] A. Abdellatif, M. Wessel, I. Steinmacher, M. A. Gerosa, and E. Shihab, "BotHunter: An approach to detect software bots in GitHub," in *International Conference on Mining Software Repositories (MSR)*, 2022, pp. 6–17.

[12] N. Chidambaram, T. Mens, and A. Decan, "RABBIT: A tool for identifying bot accounts based on their recent GitHub event history," in *International Conference on Mining Software Repositories (MSR)*. ACM, 2024.

[13] N. Chidambaram, A. Decan, and T. Mens, "A dataset of bot and human activities in GitHub," in *International Conference on Mining Software Repositories (MSR)*. IEEE/ACM, 2023, pp. 465–469.

[14] N. Chidambaram, T. Mens, and A. Decan, "A bot identification model and tool based on GitHub activity sequences," *Journal of Systems and Software*, vol. 221, March 2025.

[15] M. Golzadeh, A. Decan, and N. Chidambaram, "On the accuracy of bot detection techniques," in *International Workshop on Bots in Software Engineering (BotSE)*. IEEE, 2022.

[16] Z. Wang, Y. Wang, and D. Redmiles, "From specialized mechanics to project butlers: The usage of bots in open source software development," *IEEE Software*, vol. 39, no. 5, pp. 38–43, 2022.