Between JIRA and GitHub: ASFBot and its Influence on Human Comments in Issue Trackers

Ambarish Moharil, Dmitrii Orlov, Samar Jameel
Tristan Trouwen, Nathan Cassee, Alexander Serebrenik
Eindhoven University of Technology
Eindhoven, The Netherlands
{a.moharil,d.orlov,s.o.a.jameel,t.p.l.trouwen}@student.tue.nl
{n.w.casee,a.serebrenik}@tue.nl

ABSTRACT

Open-Source Software (OSS) projects have adopted various automations for repetitive tasks in recent years. One common type of automation in OSS is bots. In this exploratory case study, we seek to understand how the adoption of one particular bot (ASFBot) by the Apache Software Foundation (ASF) impacts the discussions in the issue-trackers of these projects. We use the SmartShark dataset to investigate whether the ASFBot affects (i) human comments mentioning pull requests and fixes in issue comments and (ii) the general human comment rate on issues. We apply a regression discontinuity design (RDD) on nine ASF projects that have been active both before and after the ASFBot adoption. Our results indicate (i) an immediate decrease in the number of median comments mentioning pull requests and fixes after the bot adoption, but the trend of a monthly decrease in this comment count is reversed, and (ii) no effect in the number of human comments after the bot adoption. We make an effort to gather first insights in understanding the impact of adopting the ASFBot on the commenting behavior of developers who are working on ASF projects.

KEYWORDS

bots, ASFBot, issue-trackers, Apache

ACM Reference Format:

Ambarish Moharil, Dmitrii Orlov, Samar Jameel and Tristan Trouwen, Nathan Cassee, Alexander Serebrenik. 2022. Between JIRA and GitHub: ASFBot and its Influence on Human Comments in Issue Trackers. In *19th International Conference on Mining SoftwareRepositories (MSR '22), May 23–24, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3524842.3528528

1 INTRODUCTION

With the growth of Open-Source Software (OSS) projects, smooth collaboration has become increasingly important. To facilitate collaboration many OSS projects also join foundations such as the Apache Software Foundation (ASF) or Eclipse Foundation [28]. To help developers, the ASF has introduced a software bot called the ASFBot that is active in their JIRA issue tracking system (ITS) [19, 20]. ITSs often become central places for idea exchange, bug tracking, and decision making for developers in OSS projects [1, 13] and many projects indicate the presence of ASF infrastructure, including ITS as a reason to join ASF in the first place [28]. At the same time, several developers feel that the "use of JIRA and poor integration with the PR process, etc. is more of a hindrance than a help"[1], discuss relative advantages and disadvantages of JIRA and GitHub[2] and pursue development using both JIRA and GitHub.[3] However, this also means that the discussions on GitHub and JIRA might diverge, e.g., when a pull request (PR) is merged on GitHub, and the associated issue in JIRA is not automatically updated. The ASFBot, a general secretarial bot in ASF, is responsible for updating JIRA when PRs are created referencing the JIRA issues, or when they are closed. The adoption of the ASFBot takes away the need for a human developer to remember to post an update if a PR related to an issue was created, updated, or merged.

Similar to the ASFBot, other bots have been actively used to support the software development process [6, 23]. Several studies have considered the impact of bots on the software development process [16, 23, 24, 27]. In these studies, the authors find that bots are frequently adopted by OSS projects and have a measurable effect on how developers communicate and review PRs. However, these studies focus on the impact that bots have on PRs, and not on the impact that bots have on ITS. While different studies have found that bots are also active in ITSs, these studies do not study the impact of these bots [10, 14].

By studying how OSS projects use the infrastructure provided by the ASF, such as the ASFBot, we hope to further understand how the infrastructure impacts the commenting behavior of humans in ITSs. Further, we seek to expand the understanding of automation adopted by OSS developers. For the ASFBot in particular we want to understand whether and how the automatic bookkeeping provided by ASF influences the commenting behavior of humans in the ITS.

Therefore, we pose the following two RQs:

(1) **RQ1**: *Does the ASFBot impact the number of comments made by humans which mention PRs and fixes across the issue-trackers of Apache projects?*
(2) **RQ2**: *Does the ASFBot impact the number of comments made by humans in the issue-trackers of Apache projects?*

MSR '22, May 23–24, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9303-4/22/05.
https://doi.org/10.1145/3524842.3528528

[1]https://lists.apache.org/thread/mnm6zo5knzgzr90jkjm1vd8tgtqtz0o5
[2]https://cwiki.apache.org/confluence/display/OFBIZ/Question%3A+GitHub+or+Jira+or+both
[3]For example, see the Kafka pull request 2648 on https://github.com/apache/kafka/pull/2648 and the related JIRA issue https://issues.apache.org/jira/browse/KAFKA-4849?.

In this paper, we perform a case study in which we study the impact of one specific bot (ASFBot) on the ITSs of nine different Apache projects. To gather the data needed for this study, we use the SmartSHARK dataset, version 2.1 [22]. We apply a methodology that is conceptually similar to that of Wessel *et al.* [24]: First, we zoom in on one specific project that adopted the ASFBot, and we analyze how activity in the ITS of the project evolved. Based on this analysis, we form two hypotheses, which we test using Regression Discontinuity Design (RDD) [3, 21]. However, while Wessel *et al.* study how bots impact code reviews, we study how one particular bot (ASFBot) impacts ITSs. Hence, we study different dependent variables in this work. The scripts and data used for this study are available as a replication package on Figshare.[4]

2 EXPLORATORY CASE STUDY

Since little is known about the effects of adopting the ASFBot in OSS, we start by conducting an exploratory case to formulate two hypotheses [15]. We aim to gain new insights and generate hypotheses for the research questions based on this case study.

Case study Methodology. To conduct our exploratory case study, we investigate the Maven project. Maven[5] has a considerable activity of the ASFBot (8.129% comment rate) over the two ITSs used by the Maven project. The project has 1156 unique human developers and 5073 unique issues.

To investigate **RQ1** and **RQ2**, we collect data related to the activity indicators given below. For each activity indicator, we take the median value per month for several months before and after the adoption of the ASFBot in Maven. Both the activity indicators and aggregation of the indicators are based on previous work [5, 24].

- **Number of developers (NO_DEVS)**: We compute the number of monthly developers who commented on an issue.
- **Number of issues (NO_ISSUES)**: We collect the number of unique monthly issues.
- **Project age (YEARS)**: We compute this as the time in years since the project has been active until the last available comment.
- **Dependent variable (COMMENTS)**: For **RQ1**, this refers to the median number of monthly human comments that contain keywords related to PRs and fixes. For **RQ2**, COMMENTS is the median number of human comments per issue in the ITS.

As the date of the ASFBot adoption in Maven, we pick the first day on which a comment of the ASFBot appears in the issue-tracker: 3 December 2014. To account for a period of instability immediately following the introduction of the ASFBot, we exclude the 30 days after the adoption of ASFBot. To study the impact of the ASFBot on the activity indicators, we collect data 12 months before the adoption and 12 months after the period of instability following the introduction of ASFBot. A similar step was taken into account for this instability by other studies that investigate the impact of interventions in Software Engineering (SE) [2, 24, 29, 30].

Case Study Results. For both dependent variables, we created a line plot showing the evolution of the variables. Figure 1 plots the median number of human comments per month that mention PRs or fixes, and Figure 2 plots the median number of human comments per month. We use the Mann-Whitney Wilcoxon test [26] to investigate the p-value for our distributions (non-parametric distributions), as was applied by Wesel *et al.* for the case study [24]. The null hypothesis H_0 assumes that *the distributions are the same*. We report both the p-value and the Cliff's Delta statistics.

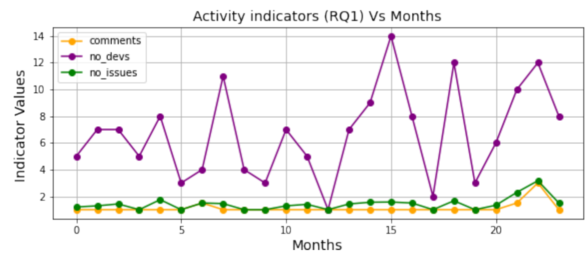

Figure 1: Monthly count of issues, developers and median comments which mention PRs and fixes for Maven.

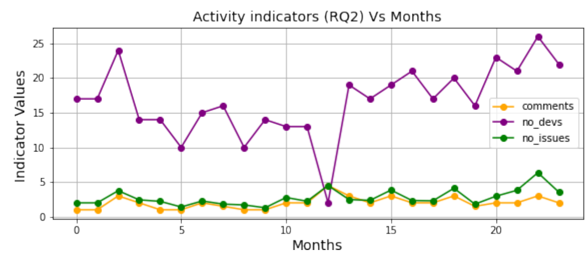

Figure 2: Monthly count of issues, developers and median comments of Maven.

From the analysis for **RQ1**, we find there is no effect on COMMENTS (p-value = 0.41421, δ = 0.09) that contain references to PRs and fixes after the adoption of ASFBot. For **RQ2** we find that there is an increase in the COMMENTS (p-value = 0.013000, δ = 0.64) after the adoption of the ASFBot. Based on these findings, we formulate the following hypotheses:

(1) $\mathbf{H_1}$: *There is no effect on the comments mentioning PRs and fixes after the adoption of the ASFBot.*
(2) $\mathbf{H_2}$: *There is an increase in the human comment rate after the adoption of the ASFBot.*

3 MAIN STUDY

In the following section, we discuss the statistical approach and data collection methods for the main study. Subsequently, we present the results of the main study.

3.1 Main Study Design

Statistical Approach. Based on the hypotheses formulated in the case study, we employed a regression discontinuity design (RDD) to study the impact of the ASFBot. Applying RDD to study the impact of interventions in SE is a common technique [2, 24, 29, 30]. We applied RDD as the following linear mixed-effects model to investigate $\mathbf{H_1}$ & $\mathbf{H_2}$ along with the longitudinal effects of the ASFBot. We implemented the model in R [12] and SAS [8]. The statistical model behind RDD is

$$
\begin{aligned}
y_i = \alpha \cdot + \beta \cdot time_i + \gamma \cdot intervention_i + \delta \\
\cdot time\_after\_intervention_i + \eta \cdot controls_i + \varepsilon_i
\end{aligned}
\tag{1}
$$

[4]https://figshare.com/s/9e9bcdcab730801dab4b
[5]https://github.com/apache/maven

where i denotes the observations for a given project. The *time* variable is measured as a certain time j in months from the start to the end of our observation period. The *intervention* variable is a binary value to indicate whether the time j occurs before (*intervention* = 0) or after (*intervention* = 1) the bot adoption. The *time_after_intervention* variable is the count in months after the ASFBot adoption. It is set to 0 before the bot adoption. Additionally, we considered the following $controls_i$ variables: the NO_DEVS, and NO_ISSUES. All of the variables mentioned in Section 2 are computed for each of these 9 projects and modeled into the mixed-effects linear regression. To account for variability originating from the fact that each project is unique, we modeled an additional variable called PROJECT_NAME, which represents the names of the nine projects as a random effect. All the other parameters were modeled as fixed effects. We fit two models, one for each dependent variable (*Median number of monthly human comments per issue that contain keywords related to PRs and fixes* & *Median number of monthly human comments per issue*), to investigate our RQs.

Data Collection. We focus on the issue comments present in the *issue comments* collection of the SmartSHARK database [22]. The database contains information about 77 projects.

Identification of the ASFBot. We identify the accounts used by ASFBot by finding users with *"ASF"* in their name. From the obtained list of 45 users, two authors inspected the comments of these bots, looking for activity confirming whether this account was human or bot. This was done by investigating their issue comments for the presence of templates such as `'This is an automated message from the Apache Git Service'`. Two additional authors confirmed the inspection results. In this way, we identified two IDs used by the ASFBot.

Identification of other bots. To answer the research questions, we separate human activity from bot activity and identify other bots present in the ITSs of the OSS projects. Bot detection has been studied previously [4, 7]. We applied two different detection techniques to detect bots other than the ASFBot.

We first implemented a completely autonomous, supervised machine learning approach suggested by Golzadeh *et al.* [7]. Goldazeh *et al.* presented a systematic approach for identifying bots based on GitHub PR and issue comments using TF-IDF vectorization and the Naive Bayes Classifier. The model predicted the presence of around 1289 bots in our issue comments. Of these predicted bots, only 8 were identified as *true* bots after a manual analysis of the names, e-mails, comments and number of comments of the account by the first two authors of the study.

We also applied *regex* filters such as "[bot]" to identify the presence of other bots and found 148 potential bot accounts using these regular expressions. The first two authors checked the activity of these accounts (name, email, comments, and number of comments) to verify whether these accounts are actually bots. This resulted in the identification of a total of 100 bots that are active in the issue trackers of the project present in the SmartSHARK dataset.[6]

Project selection. The ASFBot was found to be active in 64 out of the 77 projects in the SmartSHARK dataset. For our main study we selected projects in which the ASFBot is the only bot active in the ITS, filtering out the projects where any other identified bot was present. This allowed us to study the impact of the ASFBot in isolation. After this step, only 21 projects were left. From this sample we selected projects that were active at least one year before and after the adoption of the ASFBot, this threshold is commonly used for studies that apply RDD [2, 24, 29]. The final nine Apache projects are: *["maven", "mina-sshd", "santuario-java", "commans-bcel", "jackrabbit", "roller ,"gora", "openwebbeans", "directory-studio"].*

[6]The results of this manual labeling, and the labels given to the accounts are available as part of the replication package in the Data Cleaning folder.

3.2 Main Study Results

RQ1: Effects on the human comments mentioning PRs and fixes. Table 1 summarizes the results of the RDD analysis. There is an immediate decrease in the number of median monthly comments mentioning PRs and fixes after the adoption of the ASFBot. However, a monthly trend of a decrease in the number of comments that mention PRs and fixes is reversed after the adoption of ASFBot. The bot intervention significantly affects the trend of the number of median monthly comments mentioning PRs and fixes. As expected, the number of monthly developers and number of monthly issues also explain the number of comments that mention PRs and fixes, i.e., more comments are produced when there are more developers and more issues to be discussed. The marginal and conditional variability R_m^2 and R_c^2 proposed by [17] are 0.113 and 0.442 respectively. This means that there is a large difference between the projects. Based on these results we <u>refute</u> our $\mathbf{H_1}$.

RQ2: Effect on the human comment rate. From Table 1 we observe that there is no significant effect (p-value > 0.05) on the number of monthly median human comments (COMMENTS) after the adoption of the ASFBot. The time-series parameters do not affect the human comment rate. The only parameters that explain the human comment rate are the control variables, number of monthly contributing developers and the number of monthly issues. We find the marginal (R_m^2) and conditional variability as (R_c^2) to be 0.334 and 0.403 respectively. The observed standard errors are relatively low and the less marginal and conditional variability could be attributed to our small sample space. From this discussion we <u>refute</u> our $\mathbf{H_2}$.

4 RELATED WORK

To better understand the various types of bots and study their impact, Storey *et al.* published a call in which they list several types of bots and ask attention to the fact that researchers should study potential challenges related to bots [18]. Another classification of SE bots was proposed by Wessel *et al.*, who found that GitHub bots can be classified based on the work that they perform [23]. Erlenhov *et al.* studied existing software bots and introduced a taxonomy based on bots' purpose, communication style, and level of their intelligence [5]. Recently, they extended their taxonomy and proposed a classification framework based on three bot personas [6].

Specific effects of adopting software bots to OSS projects have been studied from different perspectives: Peng *et al.* address how the mention bot is used, finding that it saves developers' effort even if it does not optimally divide work between team members [11]. Like Peng *et al.* we focus on one specific bot in this work, however, the bot we study ASFBot is different from the mention bot studied by Peng *et al.*. Wessel *et al.* studied the stability of the

Table 1: Effect of the ASFBot on general issue comments and comments with PRs and fixes.

	Comment count mentioning PRs and fixes		General comment count	
	Coefficients	Standard error	Coefficients	Standard error
intercept	0.2699*	0.3108	-0.3071	0.3008
intervention (1)	-0.1527*	0.07225	0.07045	0.07045
after_int	0.02089***	0.005010	-0.00312	0.005063
time	-0.00157***	0.000365	-0.00041	0.000243
log_no_devs	0.1291***	0.01948	0.2934***	0.02058
log_no_issues	0.2056***	0.01675	0.2061***	0.01289
years	-0.01700	0.02008	0.01941	0.01935

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

configuration of the stale bot, finding that OSS projects don't frequently update the bot configurations [25]. Additionally, Wessel *et al.* study how code review bots impact the code reviewing practices of OSS projects [24]. To determine the impact, they analyzed several productivity indicators before and after bot adoption, finding that adoption of code reviewing bots has a noticeable impact on human activity. In this work, we take a similar approach to that of Wessel *et al.* to study the impact of ASFBot on the issue trackers of projects that are part of ASF. However, instead of focusing on bots in general, we focus on one specific bot in one ecosystem.

Liu *et al.* perform a mixed-methods empirical research to understand how developers interact with bots in OSS projects [9]. As a result, they propose a set of guidelines for future bot developers. When it comes to bots that open PRs, Wyrich *et al.* study how these bots are perceived by developers. Wyrich *et al.* find that code changes proposed by bots are smaller than changes proposed by humans [27]. However, Wyrich *et al.* also find that bot changes are less likely, and take longer, to be merged. Saadat *et al.* built a bot classifier and compared PRs where bots were present with human-only PRs, finding that PRs where bots are present exhibit more complex interaction patterns [16].

5 THREATS TO VALIDITY

Construct Validity. SmartSHARK mentions no explicit date for the ASFBot adoption. Hence, we regard the date of the first *bot* issue comment as the date of bot adoption. Even though we implemented a machine learning approach to identify bots, most of the bots were identified using regex and manual filtering.

Internal Validity. To confirm the robustness of our model, we apply filtering to remove projects that mention no comments in the selected time-frame and that had the effect of bots other than the ASFBot. It is to be noted that we had no data on whether the ASFBot was disabled for a certain period of time within the project.

External Validity. The effect of the ASFBot found in this study is based on a sample of nine ASF projects. These nine projects are all projects where the ASFBot is the only bot present in the ITS. Thus, the effects observed might not generalize to other ASF projects that use the ASFBot together with other bots in the ITS.

6 DISCUSSION

Addressing **RQ1**, we found insufficient evidence in the Maven dataset in the exploratory case study to conclude that there was an effect on comments mentioning PRs and fixes after the adoption of the ASFBot. Hence, we hypothesized that such effects would not exist in the main study. However, while studying the behavior of nine different OSS projects that are part of the ASF, we found a considerable impact of the ASFBot. This concurs with the previous observation that OSS projects join ASF to benefit from the infrastructure provided [28]. In this work, we find empirical support for the advantages provided by the ASF, and we find that developers can be more productive when the ASFBot is present as they no longer need to perform the repetitive task of manually updating issues in JIRA when merging a PR on GitHub.

As for **RQ2**, we did observe more developer participation in issue comments after the adoption of ASFBot during the exploratory case study. However, we did not confirm this in the mixed-effects model. Instead, we saw that an increase in the number of comments was mostly explained by an increased number of developers or issues. This might indicate that the length of developer discussions was unaffected by the information provided by the ASFBot. Unlike the results obtained by Wessel *et al.*, who found that the adoption of PR bots leads to a decrease in the number of comments made on PRs [24]. One possible explanation might be that providing links to PRs or potential fixes generally act as auxiliary information having little influence on developers' engagement, unlike bots that are active in PRs that were studied by Wessel *et al.*

7 CONCLUSION

We investigate the influence of the ASFBot on OSS projects that are part of the ASF. We differentiated between comments mentioning PRs and fixes and general human comments. For both the comment types, we formulated corresponding research questions that address the effects of the ASFBot adoption in a project.

Firstly, we did an exploratory case study on the data from the project Maven. Based on this, we hypothesized that the adoption of the ASFBot increases the number of human comments and does not affect the number of comments mentioning PRs and fixes. In the main study, we gathered data from nine projects which adopt the ASFBot. Applying regression discontinuity design on this data allowed us to reject both hypotheses. We discovered that the increase in general comments is only associated with an increase in the number of issues and participating developers. The number of human comments mentioning PRs and fixes does diminish.

Our results can be used by OSS projects to decide if a bot referencing PRs and fixes in issue comments is beneficial. Future work can be done in analyzing the reasons for the observed human behavior more thoroughly.

REFERENCES

[1] Dane Bertram, Amy Voida, Saul Greenberg, and Robert Walker. 2010. Communication, Collaboration, and Bugs: The Social Nature of Issue Tracking in Small, Collocated Teams. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work* (Savannah, Georgia, USA) *(CSCW '10)*. Association for Computing Machinery, New York, NY, USA, 291–300. https://doi.org/10.1145/1718918.1718972

[2] Nathan Cassee, Bogdan Vasilescu, and Alexander Serebrenik. 2020. The Silent Helper: The Impact of Continuous Integration on Code Reviews. In *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 423–434. https://doi.org/10.1109/SANER48275.2020.9054818

[3] Thomas D Cook and D T Campbell. 1979. *Quasi-Experimentation: Design and Analysis Issues for Field Settings.* Houghton Mifflin.

[4] Tapajit Dey, Sara Mousavi, Eduardo Ponce, Tanner Fry, Bogdan Vasilescu, Anna Filippova, and Audris Mockus. 2020. Detecting and Characterizing Bots that Commit Code. In *MSR '20: 17th International Conference on Mining Software Repositories, Seoul, Republic of Korea, 29-30 June, 2020*, Sunghun Kim, Georgios Gousios, Sarah Nadi, and Joseph Hejderup (Eds.). ACM, 209–219. https://doi.org/10.1145/3379597.3387478

[5] Linda Erlenhov, Francisco Gomes de Oliveira Neto, Riccardo Scandariato, and Philipp Leitner. 2019. Current and Future Bots in Software Development. In *2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE)*. IEEE Press, Montreal, Quebec, Canada, 7–11. https://doi.org/10.1109/BotSE.2019.00009

[6] Linda Erlenhov, Francisco Gomes de Oliveira Neto, and Philipp Leitner. 2020. An Empirical Study of Bots in Software Development: Characteristics and Challenges from a Practitioner's Perspective. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Virtual Event, USA) *(ESEC/FSE 2020)*. Association for Computing Machinery, New York, NY, USA, 445–455. https://doi.org/10.1145/3368089.3409680

[7] Mehdi Golzadeh, Alexandre Decan, Eleni Constantinou, and Tom Mens. 2021. Identifying bot activity in GitHub pull request and issue comments. *CoRR* abs/2103.06042 (2021), 21–25. arXiv:2103.06042 https://arxiv.org/abs/2103.06042

[8] SAS Institute. 1985. *SAS user's guide: Statistics*. Vol. 2. Sas Inst.

[9] Dongyu Liu, Micah J. Smith, and Kalyan Veeramachaneni. 2020. Understanding User-Bot Interactions for Small-Scale Automation in Open-Source Development *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3334480.3382998

[10] Marco Ortu, Giuseppe Destefanis, Bram Adams, Alessandro Murgia, Michele Marchesi, and Roberto Tonelli. 2015. The JIRA Repository Dataset: Understanding Social Aspects of Software Development. In *Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering* (Beijing, China) *(PROMISE '15)*. Association for Computing Machinery, New York, NY, USA, Article 1, 4 pages. https://doi.org/10.1145/2810146.2810147

[11] Zhenhui Peng and Xiaojuan Ma. 2019. Exploring how software developers work with mention bot in GitHub. *CCF Transactions on Pervasive Computing and Interaction* 1, 3 (01 Nov 2019), 190–203. https://doi.org/10.1007/s42486-019-00013-2

[12] R Core Team. 2022. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

[13] Michael Rath and Patrick Mäder. 2020. *Request for Comments: Conversation Patterns in Issue Tracking Systems of Open-Source Projects.* Association for Computing Machinery, New York, NY, USA, 1414–1417. https://doi.org/10.1145/3341105.3374056

[14] Michael Rath and Patrick Mäder. 2020. Request for Comments: Conversation Patterns in Issue Tracking Systems of Open-Source Projects. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. Association for Computing Machinery, New York, NY, USA, 1414—-1417. https://doi.org/10.1145/3341105.3374056

[15] Per Runeson and Martin Höst. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering* 14, 2 (2009), 131–164. https://doi.org/10.1007/s10664-008-9102-8

[16] Samaneh Saadat, Natalia Colmenares, and Gita Sukthankar. 2021. Do Bots Modify the Workflow of GitHub Teams? arXiv:2103.09319 [cs.SE]

[17] Holger Schielzeth. 2013. A General and Simple Method for Obtaining R2 from Generalized Linear Mixed-Effects Models. *Methods in Ecology and Evolution* 4 (02 2013), 133–142. https://doi.org/10.1111/J.2041-210x.2012.00261.X

[18] Margaret-Anne Storey and Alexey Zagalsky. 2016. Disrupting Developer Productivity One Bot at a Time. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (Seattle, WA, USA) *(FSE 2016)*. Association for Computing Machinery, New York, NY, USA, 928–931. https://doi.org/10.1145/2950290.2983989

[19] The Apache Software Foundation. 2022. ASFBot: IRC services and archives. https://infra.apache.org/asfbot.html [Online; accessed 18-January-2022].

[20] The Apache Software Foundation. 2022. System Dashboard - ASF JIRA. https://issues.apache.org/jira/secure/Dashboard.jspa [Online; accessed 18-January-2022].

[21] Donald L. Thistlethwaite and Donald T. Campbell. 1960. Regression-discontinuity analysis: An alternative to the ex post facto experiment. *Journal of Educational Psychology* 51 (1960), 309–317.

[22] Alexander Trautsch, Fabian Trautsch, and Steffen Herbold. 2021. MSR Mining Challenge: The SmartSHARK Repository Mining Data. arXiv:2102.11540 [cs.SE]

[23] Mairieli Wessel, Bruno Mendes de Souza, Igor Steinmacher, Igor S. Wiese, Ivanilton Polato, Ana Paula Chaves, and Marco A. Gerosa. 2018. The Power of Bots: Characterizing and Understanding Bots in OSS Projects. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 182 (nov 2018), 19 pages. https://doi.org/10.1145/3274451

[24] Mairieli Wessel, Alexander Serebrenik, Igor Scaliante Wiese, Igor Steinmacher, and Marco Aurelio Gerosa. 2020. Effects of Adopting Code Review Bots on Pull Requests to OSS Projects. In *Proceedings - 2020 IEEE International Conference on Software Maintenance and Evolution, ICSME 2020*. IEEE Computer Society, United States, 1–11. https://doi.org/10.1109/ICSME46990.2020.00011

[25] Mairieli Wessel, Igor Steinmacher, Igor Wiese, and Marco A. Gerosa. 2019. Should I Stale or Should I Close? An Analysis of a Bot That Closes Abandoned Issues and Pull Requests. In *2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE)*. 38–42. https://doi.org/10.1109/BotSE.2019.00018

[26] D.S. Wilks. 2011. *Statistical Methods in the Atmospheric Sciences.* Elsevier Science. https://books.google.nl/books?id=fxPiH9Ef9VoC

[27] Marvin Wyrich, Raoul Ghit, Tobias Haller, and Christian Müller. 2021. Bots Don't Mind Waiting, Do They? Comparing the Interaction With Automatically and Manually Created Pull Requests. *CoRR* abs/2103.03591 (2021). arXiv:2103.03591 https://arxiv.org/abs/2103.03591

[28] Nan Yang, Isabella Fereira, Alexander Serebrenik, and Bram Adams. 2022. Why Do Projects Join the Apache Software Foundation?. In *44th IEEE/ACM International Conference on Software Engineering: Software Engineering in Society, ICSE (SEIS) 2022.* ACM, xx–xx.

[29] Yangyang Zhao, Alexander Serebrenik, Yuming Zhou, Vladimir Filkov, and Bogdan Vasilescu. 2017. The impact of continuous integration on other software development practices: A large-scale empirical study. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 60–71. https://doi.org/10.1109/ASE.2017.8115619

[30] Théo Zimmermann and Annalí Casanueva Artís. 2019. Impact of Switching Bug Trackers: A Case Study on a Medium-Sized Open Source Project. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 13–23. https://doi.org/10.1109/ICSME.2019.00011