Analytics Dashboards and User Behavior: Evidence from GitHub

J. Schilpp*, F. Pethig* and H. Hoehle*
* University of Mannheim, Germany
jan.schilpp@uni-mannheim.de

Abstract – **In open source software development it is essential to stimulate developers' contribution behavior in the absence of monetary incentives. But research on potential secondary effects of these stimuli, such as on contribution sentiment, remains scarce. Thus, we investigate how a promising external stimulus, an analytics dashboard, influences developers' number of contributions as well as associated message length and sentiment. Building on social comparison and goal setting theory we hypothesize that adopting the analytics dashboard increases developers' number of contributions but decreases contribution message length and sentiment. To test these hypotheses, we leverage granular data from a matched sample of 43,434 GitHub developers over two years who adopted an analytics dashboard on their personal developer page. Our difference-in-differences analyses reveal that adopting the analytics dashboard increases a developers' number of contributions and associated message length. However, after adopting the analytics dashboard users write more negative messages and this effect is primarily driven by users with low initial GitHub activity. These findings suggest that the performance increase through analytics dashboards comes at the cost of potentially harmful consequences for the developers or the platform. We discuss implications for theory and practice.**

Keywords – open; source; software; developer; analytics; dashboard; contribution; length; sentiment

I. Introduction

Nowadays, 90 percent of companies rely on open source software (OSS), making it indispensable for their system landscape [1]. However, OSS is predominantly developed by volunteers. Thus, it is a constant struggle for OSS platforms to motivate users' contributions.

Current research on OSS developers investigates the sources of developers' motivation and commitment when contributing to OSS [2]–[4] as well as potential ways how their contribution behavior can be altered [5]. Most studies investigate primary effects, i.e., how the number or quality of contributions is affected. But there is little research on potentially unintended secondary effects of external stimuli on OSS developers' contribution behavior, e.g., the contribution sentiment. This is a central research gap, in particular because insights from organizational software developers show that they might personally suffer from the idiosyncrasies of software development [6]–[8].

We aim to approach this gap by investigating how OSS developers are affected by a promising external stimulus, an analytics dashboard. The latter's popularity heavily increased in the past years given the high data availability [9] and similar stimuli were implemented on popular OSS platforms such as GitHub [10]. Furthermore, analytics dashboards have shown to have positive effects in other contexts, e.g., improved decision-making and performance [9]. Yet, the increased transparency associated with dashboards might have negative effects on the individual developer such as strategic effort allocation or stress. Therefore, we investigate the impact of analytics dashboards on OSS developers' number of contributions as well as the associated message length and sentiment. Overall, our goal is to answer the following research question: *How does an analytics dashboard influence developers' contribution behavior?*

We exploit a unique extension of the OSS development platform GitHub, which enables users to display a descriptive dashboard on their profile page[1]. This allows adopters to track their own activities and easily compare them with others. To understand how this impacts adopters' contribution behavior as well as what underlying mechanism might be we build on social comparison and goal setting theory. These theories provide rich insights into how the comparison with others, enabled by the analytics dashboard, impact developers' number of contributions, the contribution message length, and their sentiment. Thereby, this research is expected to make two important contributions to research on OSS developers. First, we investigate primary effects of analytics dashboards on OSS developers' contribution behavior. Second, we shed light on potentially unintended secondary effects, i.e., the contribution message length and the associated sentiment.

II. Theoretical Background

In the following we first draw from research on OSS developers. Next, we introduce social comparison and goal setting theory to explain how analytics dashboards influence developer contribution behavior.

A. OSS Developers

The main difference between software developers in organizational and OSS settings is that software developers within organizations mainly contribute due to monetary incentives while OSS developers typically act on a voluntary basis. Therefore, OSS platforms must find alternative ways to influence their users' contribution behavior.

[1] https://github.com/anuraghazra/github-readme-stats

Correspondingly, studies on OSS developers focus on explaining why developers contribute to OSS. It is found that value and ideology congruence between a developer and the respective community is important [2], [11]. Moreover, OSS developers enjoy being an integral part of a community [3], [4]. Eventually, the contribution behavior of members in online communities can be altered by costly programs such as content contribution by the platform [5]. Overall, the identified determinants of OSS developer contribution behavior are mostly structural community features or costly efforts that cannot be easily influenced or implemented.

Nonetheless, it is important to incentivize OSS developers in a manner that sustainably alters their contribution behavior, i.e., considering primary effects such as the number of contributions as well as secondary effects such as the associated message length and sentiment [12]. One way this can be achieved is through an external stimulus on the respective OSS platform [10]. Given the high availability of data, descriptive analytics dashboards that display a developers' contributions and thereby allow to compare oneself with others are a possible path.

B. Social Comparison and Goal Setting Theory

We build on social comparison and goal setting theory to explain how analytics dashboards lead to a change in OSS developers' contribution behavior. Social comparison theory argues that people want to evaluate themselves, e.g., through comparison with others, especially people with greater abilities [13]. An analytics dashboard allows OSS developers to easily compare their number of contributions with peers. More specifically, analytics dashboard adopters transparently see that other users contribute more than oneself. Thus, we can expect that adopting the analytics dashboard incentivizes developers to contribute more to match more active users.

To explain the underlying mechanism, we borrow from goal setting theory. It argues that "people are motivated to strive towards goals" [14, p. 509] and regulate their effort to achieve their objectives accordingly. Reconsidering the adopters, it is their goal to match more active users' number of contributions. Consequently, the adopters direct effort toward improving the statistics displayed in the analytics dashboard. Based on this mechanism we propose our first hypothesis.

Hypothesis 1: *Adopting an analytics dashboard increases developers' number of contributions.*

The increased effort necessarily reduces the adopters' effort put into other things. More specifically, one can suspect that adopters direct less effort toward tasks that are not made visible by analytics dashboards. As the latter are of descriptive nature, they only display "what happened" [9, p. 1029] on a high level and properties of contributions such as the associated message length are not exhibited. Furthermore, previous research shows that more active developers might document less because they perceive changes to be clear just from the software code [15]. Thus, we hypothesize the following:

Hypothesis 2: *Adopting an analytics dashboard decreases developers' message length per contribution.*

Eventually, social comparison theory argues that the comparison with other people shapes one's self-perception [13]. That is, humans are found to have a more negative self-perception due to frequent comparisons with others because they consider other people to be better [16], [17]. Thus, we can assume that the social comparison enabled by the analytics dashboard negatively influences developer's self-perception. This should be visible in the sentiment of adopters' contributions. Our third hypothesis follows from this.

Hypothesis 3: *Adopting an analytics dashboard decreases developers' sentiment of contributions.*

III. Method

A. Research Setting

The three hypotheses are tested with data from OSS developers on GitHub. It is the most popular website for OSS development with more than 100 million users in February 2023 [18]. Leveraging the version control system Git, the platform allows developers to perform all necessary software development activities, i.e., the organization of single-person or collaborative projects, called repositories, through features like commits (i.e., changes to one or multiple files in a repository, typically software code accompanied by a short message that normally describes implemented changes), pull requests (i.e., proposed changes to a repository that are to be accepted by collaborators) as well as issues (i.e., a discussion thread related to a repository) [19]. Furthermore, GitHub offers social features such as user profiles and the possibility to follow other users.

The paper exploits the opportunity that some GitHub developers adopted a descriptive analytics dashboard on their profile page. The dashboard is depicted in Fig. 1. It displays the number of users' most important contributions on GitHub and ranks these users accordingly relative to other users.

B. Research Data

The analytics dashboard's initial launch was on July 9, 2020, and to implement it on a profile page users must add a string to their *Readme.md* in a commit to their own repository. We exploited this to determine if and when GitHub users adopted the analytics dashboard. Thereby, we identified 137,178 adopters of the analytics element in

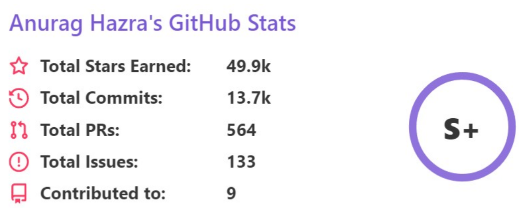

Figure 1. Analytics dashboard[2]

[2] Retrieved from https://github.com/anuraghazra/github-readme-stats, October 12, 2022

a two-year time window between July 9, 2020, and July 9, 2022. We then collected all public GitHub activities of these users between March 19, 2020, and October 29, 2022, from GhArchive[3].

We limit our analysis to commits for two reasons. First, commits are the most important type of contribution on GitHub as they are mostly software code, the main goal to be produced on an OSS platform. Second and given the enormous amount of data, this makes the analysis computationally more feasible and accessible for researchers and readers alike.

Furthermore, we filtered the users included in the analysis along multiple characteristics. First, we limited the sample to personal GitHub accounts of single developers by removing organizational accounts and bots. The latter is achieved by filtering users based on names and an unrealistic high number of contributions [10]. Second, we limited the sample to users that had at least ten commits in our timeframe to ensure that users were actively using GitHub. Third, we reduced the sample to developers who adopted the analytics dashboard and did not remove it within our study period. This allowed us to compare the effect of the dashboard across adopters. Fourth, we removed the commits users made to their own repository as this could bias our estimation because adopters must modify their repository to implement the dashboard.

We deployed a Bidirectional Encoder Representations from Transformers (BERT) machine learning model to determine the commit message sentiment [22]. A finetuned BERT model has shown to outperform other approaches to sentiment analysis in the software engineering domain [23]. Thus, we compile a ground truth dataset that is representative of the to be classified commit messages to finetune a BERT model for our data. This ground truth dataset is based on five publicly available datasets and encompasses 17,908 short messages from software development that are manually labelled as positive, negative, or neutral [24]–[28]. Our finetuned BERT model achieves an accuracy of 97%. To capture the nuances of a commit message we calculate the message's net sentiment by subtracting the predicted probability of having a negative message from the predicted probability of having a positive message.

Eventually, we aggregated the data on users' commits, i.e., the number of commits, the commit message length measured in number of characters, and net sentiment, on a monthly level to make the analysis computationally feasible and reduce noise in the data that stems from irregular contribution behavior. Thereby, we obtain a panel data set with one observation per user and month.

C. Research Method

We use a difference-in-differences (DiD) approach to compare the behavior of analytics dashboards adopters with the behavior of a suitable control group [29]. To account for the staggered adoption of the analytics dashboard we normalize the months relative to the treatment group's month of adoption.

As users deliberately choose to adopt the analytics dashboard, there is a high danger of self-selection into the treatment group of adopters. This might bias the results because users might have changed their behavior before adopting the analytics dashboard. For instance, users could become more active and then adopt the analytics dashboard to show their increased activity. To account for this issue, we deploy look ahead coarsened exact matching to identify a suitable control group [30]–[32]. This approach encompasses two elements that are best illustrated with an example of two GitHub users, A and B, with A adopting the analytics dashboard in month m and B in month m+10. We can expect these users to be similar along time-invariant unobservable characteristics such as their dedication to contribute to GitHub because both adopted the analytics dashboard eventually. Additionally, we match these users based on observable characteristics, i.e., the number of commits (log), mean commit message length, mean net sentiment per commit message, and tenure in months (log) with the help of coarsened exact matching. This matching is based on the activity of both users in month m-6 to account for the possibility that the users might have changed their contribution behavior before adopting the analytics dashboard [31]. Thus, our identification strategy aims to reduce differences between user A and B along unobservable as well as observable characteristics. In our setting, user B can be considered a suitable control group for user A between month m-5 and m+4. Consequently, the DiD analysis is based on data from month m-5 to m+4 (including m as part of the after-treatment period). We choose to analyze data from five months before and after the adoption to investigate the lasting impact of the analytics dashboard. Applying this matching procedure, we obtain a sample of 43,434 users (21,717 adopters and 21,717 future adopters) with 434,340 user-month observations[4].

IV. Results

A. Number of Contributions

To test whether the analytics dashboard influences developers' number of contributions (Hypothesis 1), we analyze how the number of commits changes before and after the adoption. Fig. 2. illustrates the mean number of commits for adopters and future adopters of the dashboard

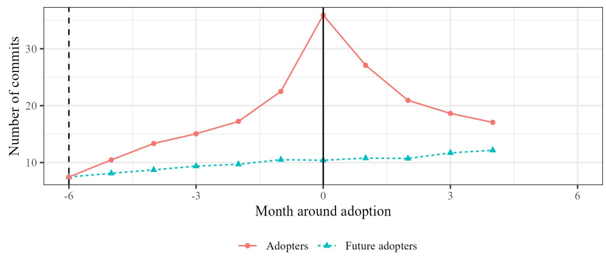

Figure 2. Number of commits around the adoption of the analytics dashboard

[3] The collection of this data was only feasible thanks to the Python Multiprocess module [20], [21].

[4] We repeat our main analysis with a matched sample of adopters and non-adopters as well with all adopters using an instrumental variable approach. We obtain similar results as for the look ahead matched sample.

around the adoption of the analytics dashboard. At the first glance, it is obvious that adopters commit most in the month of adoption. This intuitively makes sense as users must have been active in the adoption month to implement the dashboard, increasing the probability that they also contributed to other projects. Furthermore, the mean number of commits by adopters increases in the months leading up to the adoption. However, the graph suggests that the increased number of commits persists even after adoption as can be seen by the widening the gap between adopters and future adopters. That is, considering the same temporal distance from the adoption month, users commit more in the post- than in the respective pre-adoption period.

To further test this assumption, we estimate a Poisson DiD regression on the monthly number of commits with user and calendar month fixed effects. Using a Poisson model is the appropriate specification as the dependent variable follows a count data distribution. Moreover, we estimate separate regressions depending on the users' number of commits during the matching month to account for the idiosyncratic properties of user activity on GitHub. This contribution behavior follows a power law distribution where a minority of users is responsible for a majority of commits [33], [34]. The results are displayed in model 1 and 2 of Tab. I. The DiD coefficient (Adoption x After) is positive and significant for the low and high activity group which suggests that users commit more after adopting the analytics dashboard.

Consequently, we consider the first hypothesis to be supported and conclude that adopting the analytics dashboard increases developers' number of contributions.

B. *Message Length per Contribution*

As hypothesized, one might suspect that the higher number of commits comes at the cost that the adopters write shorter commit messages (Hypothesis 2). We investigate this proposition by considering the mean commit message length per commit. Fig. 3 displays the mean commit message length for adopters and future adopters around the adoption of the analytics dashboard. Surprisingly, we observe an increase in the mean commit message length per commit after the adoption of the analytics dashboard for the group of adopters.

Following the same structure as before, model 3 and 4 in Tab. I display the results of an ordinary least squared (OLS) regression on the mean commit message length. Again, we find both DiD estimators to be positive and significant suggesting that the commit message length increases after adopting the analytics dashboard.

From these findings, we conclude that adopting the analytics dashboard positively influences developers' message length per contribution and we reject the second hypothesis.

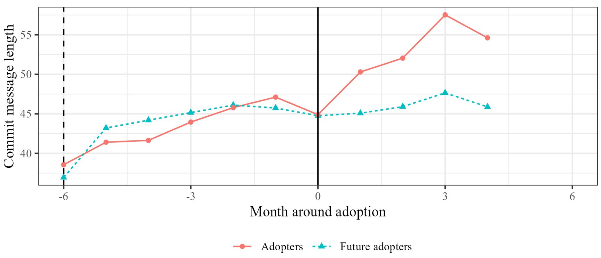

Figure 3. Mean commit message length around the adoption of the analytics dashboard

C. *Sentiment of Contributions*

Next, we examine the impact of the analytics dashboard on the commit message net sentiment as computed by our fine-tuned BERT model (Hypothesis 3). Model-free evidence in Fig. 4 suggests that the mean net sentiment of the commit messages among adopters is lowest during the month of adoption. This negativity partly diminishes in the following months but the commit message net sentiment for adopters stagnates on a visibly lower level than before the adoption and compared to the group of future adopters.

We complement this model-free evidence with an OLS regression on the mean net sentiment of the commit messages. The results are displayed in model 5 and 6 of Tab. I. Interestingly, the DiD estimator is negative and significant for low activity users but not for high activity users.

Overall, these findings provide partial support for our third hypothesis. We infer that adopting the analytics dashboard decreases the sentiment of contributions by users with low initial activity.

V. Discussion

Sustainably incentivizing OSS developer contribution behavior is essential but current research does not offer deeper insights into the impact of external stimuli such as analytics dashboards. To fill this important research gap, we build on social comparison and goal setting theory and argue that social comparison between developers impacts developers' number of contributions as well as associated message length and sentiment. Analyzing data from 43,434 developers, we find evidence for a positive effect of an analytics dashboard on developers' number of contributions and, surprisingly, associated message length. The latter might be driven by developers experiencing the need for longer documentation when committing more and collaborating with others [35]. Nonetheless, empirical evidence suggests a negative effect on contribution sentiment for initially less active developers. Two alternative mechanisms might explain the latter. First, less active adopters might experience higher stress or pressure mirrored in a more negative message sentiment. Second, these developers might adapt themselves to the tone of the platform.

We contribute to research on OSS developers by uncovering negative effects of altering contribution behavior as well as potential underlying mechanisms. For practitioners, our results imply that one should be careful

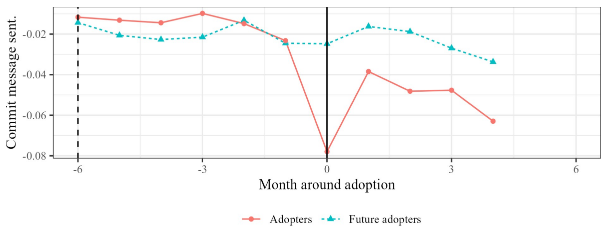

Figure 4. Mean commit message net sentiment around the adoption of the analytics dashboard

TABLE I. REGRESSIONS ON THE NUMBER OF COMMITS, MEAN COMMIT MESSAGE LENGTH, AND MEAN NET SENTIMENT

	Number of Commits (Poisson)		**Commit Message Length (OLS)**		**Net Sentiment (OLS)**	
	(1) Low	***(2) High***	***(3) Low***	***(4) High***	***(5) Low***	***(6) High***
After	0.4362*** (0.0370)	0.1101*** (0.0313)	0.3866 (0.5582)	1.4723 (0.9847)	0.0006 (0.0004)	0.0004 (0.0009)
Adoption x After	0.1043** (0.0444)	0.1875*** (0.0377)	4.0185*** (0.5428)	2.1188** (0.8865)	-0.0012** (0.0004)	-0.0000 (0.0008)
log(1+Tenure)	0.2644*** (0.0904)	-0.0384 (0.0938)	1.2579 (0.9629)	1.0388 (1.5167)	-0.0003 (0.0007)	-0.0032*** (0.0018)
Number of commits			0.0991*** (0.0263)	0.0584*** (0.0108)	-0.0000* (0.000)	-0.0000*** (0.0000)
User fixed effect	Yes	Yes	Yes	Yes	Yes	Yes
Calendar month fixed effect	Yes	Yes	Yes	Yes	Yes	Yes
Observations	284,960	104,030	328,700	105,640	328,700	105,640
Users	28,496	10,403	32,870	10,564	32,870	10,564
Adj. R^2			0.007	0.004	0.000	0.000

Note: The dependent variables are displayed in the first row. Users' activity level during the matching month according to a split along the 75th percentile are displayed in the second row. The number of users in column 1 and 2 differ because groups with zero observations are automatically dropped in a fixed effects poisson regression. Robust standard errors in parentheses. * p<0.05, ** p<0.01, *** p<0.001.

when incentivizing OSS developer contributions as this might come at a cost for the developers or the platform.

We are currently working on deepening these initial results to strengthen causal claims and to scrutinize underlying mechanisms. First, we are conducting additional robustness tests to test for parallel trends and the stable unit treatment value assumption. Second, we intend to conduct a randomized controlled trial with developers who are incentivized to adopt the analytics dashboard. Third, we are collecting additional qualitative evidence through interviews with adopters.

VI. CONCLUSION

Overall, our study suggests that the positive influence on contribution behavior triggered by analytics dashboards come at the cost of potentially harmful secondary consequences for the developers or the platform.

ACKNOWLEDGMENT

This work was supported by the University of Mannheim's Graduate School of Economic and Social Sciences. Furthermore, the authors acknowledge support by the state of Baden-Württemberg through bwHPC

REFERENCES

[1] GitHub Inc., "Octoverse 2022: The state of open source," *Octoverse 2022*, 2022. https://octoverse.github.com/ (accessed Feb. 02, 2023).

[2] L. M. Maruping, S. L. Daniel, and M. Cataldo, "Developer centrality and the impact of value congruence and incongruence on commitment and code contribution activity in open source software communities," *MIS Quarterly*, vol. 43, no. 3, pp. 951–976, Sep. 2019, doi: 10.25300/MISQ/2019/13928.

[3] G. von Krogh, S. Haefliger, S. Spaeth, and M. W. Wallin, "Carrots and rainbows: motivation and social practice in open source software development," *MIS Quarterly*, vol. 36, no. 2, pp. 649–676, Jun. 2012, doi: 10.2307/41703471.

[4] C. Zhang, J. Hahn, and P. De, "Research note—continued participation in online innovation communities: does community response matter equally for everyone?," *Information Systems Research*, vol. 24, no. 4, pp. 1112–1130, Dec. 2013, doi: 10.1287/isre.2013.0485.

[5] Y. Deng, J. Zheng, W. Khern-am-nuai, and K. Kannan, "More than the quantity: the value of editorial reviews for a user-generated content platform," *Management Science*, vol. 68, no. 9, pp. 6355–7064, Sep. 2022, doi: 10.1287/mnsc.2021.4238.

[6] S. Sarker, M. Ahuja, and S. Sarker, "Work–life conflict of globally distributed software development personnel: an empirical investigation using border theory," *Information Systems Research*, vol. 29, no. 1, pp. 103–126, Jan. 2018, doi: 10.1287/isre.2017.0734.

[7] V. Venkatesh, A. Rai, and L. M. Maruping, "Information Systems Projects and Individual Developer Outcomes: Role of Project Managers and Process Control," *Information Systems Research*, vol. 29, no. 1, pp. 127–148, Jan. 2018, doi: 10.1287/isre.2017.0723.

[8] V. Venkatesh, J. Y. L. Thong, F. K. Y. Chan, H. Hoehle, and K. Spohrer, "How agile software development methods reduce work exhaustion: Insights on role perceptions and organizational skills," *Information Systems Journal*, vol. 30, no. 4, pp. 733–761, Jul. 2020, doi: 10.1111/isj.12282.

[9] R. Berman and A. Israeli, "The value of descriptive analytics: evidence from online retailers," *Marketing Science*, vol. 41, no. 6, pp. 1074–1096, Nov. 2022, doi: 10.1287/mksc.2022.1352.

[10] L. Moldon, M. Strohmaier, and J. Wachs, "How gamification affects software developers: cautionary evidence from a natural experiment on GitHub," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, May 2021, pp. 549–561. doi: 10.1109/ICSE43902.2021.00058.

[11] L. M. Daniel, L. Maruping, M. Cataldo, and J. Herbsleb, “The impact of ideology misfit on open source software communities and companies,” *MIS Quarterly*, vol. 42, no. 4, pp. 1069–1096, Dec. 2018, doi: 10.25300/MISQ/2018/14242.

[12] S. Matook, G. Lee, and B. Fitzgerald, “Information systems development,” *MIS Quarterly Research Curations*, Dec. 2021. http://misq.org/research-curations (accessed Jan. 13, 2022).

[13] L. Festinger, “A theory of social comparison processes,” *Human Relations*, vol. 7, no. 2, pp. 117–140, May 1954, doi: 10.1177/001872675400700202.

[14] R. N. Landers, K. N. Bauer, and R. C. Callan, “Gamification of task performance with leaderboards: A goal setting experiment,” *Computers in Human Behavior*, vol. 71, pp. 508–515, Jun. 2017, doi: 10.1016/j.chb.2015.08.008.

[15] E. A. Alomar, A. Peruma, M. W. Mkaouer, C. D. Newman, and A. Ouni, “Behind the scenes: On the relationship between developer experience and refactoring,” *Journal of Software: Evolution and Process*; e2395, Oct. 2021, doi: 10.1002/smr.2395.

[16] S. Y. Lee, “How do people compare themselves with others on social network sites?: The case of Facebook,” *Computers in Human Behavior*, vol. 32, pp. 253–260, Mar. 2014, doi: 10.1016/j.chb.2013.12.009.

[17] D. A. de Vries and R. Kühne, “Facebook and self-perception: Individual susceptibility to negative social comparison on Facebook,” *Personality and Individual Differences*, vol. 86, pp. 217–221, Nov. 2015, doi: 10.1016/j.paid.2015.05.029.

[18] GitHub Inc., “GitHub About,” *GitHub*, Feb. 2023. https://github.com/about (accessed Feb. 02, 2023).

[19] GitHub Inc., “GitHub glossary,” *GitHub Docs*, Jan. 2022. https://docs.github.com (accessed Jan. 16, 2023).

[20] M. M. McKerns and M. Aivazis, “pathos: a framework for parallel graph management and execution in heterogeneous computing,” *pathos*, 2010. https://mmckerns.github.io/project/pathos/wiki.html (accessed Feb. 02, 2023).

[21] M. M. McKerns, L. Strand, T. Sullivan, A. Fang, and M. A. G. Aivazis, “Building a framework for predictive science,” in *Proceedings of the 10th Python in Science Conference (SciPy 2011)*, Austin, Texas, Jul. 2011. doi: 10.48550/arXiv.1202.1056.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding.” arXiv, May 2019. doi: 10.48550/arXiv.1810.04805.

[23] J. Wu, C. Ye, and H. Zhou, “BERT for sentiment classification in software engineering,” in *2021 International Conference on Service Science (ICSS)*, May 2021, pp. 115–121. doi: 10.1109/ICSS53362.2021.00026.

[24] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, “Sentiment polarity detection for software development,” *Empirical Software Engineering*, vol. 23, no. 3, pp. 1352–1382, Jun. 2018, doi: 10.1007/s10664-017-9546-9.

[25] B. Lin, F. Zampetti, R. Oliveto, M. Di Penta, M. Lanza, and G. Bavota, “Two datasets for sentiment analysis in software engineering,” in *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Sep. 2018, pp. 712–712. doi: 10.1109/ICSME.2018.00084.

[26] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto, “Sentiment analysis for software engineering: how far can we go?,” in *Proceedings of the 40th International Conference on Software Engineering*, May 2018, pp. 94–104. doi: 10.1145/3180155.3180195.

[27] N. Novielli, F. Calefato, D. Dongiovanni, D. Girardi, and F. Lanubile, “Can we use se-specific sentiment analysis tools in a cross-platform setting?,” in *Proceedings of the 17th International Conference on Mining Software Repositories*, Sep. 2020, pp. 158–168. doi: 10.1145/3379597.3387446.

[28] G. Uddin and F. Khomh, “Automatic mining of opinions expressed about apis in stack overflow,” *IEEE Transactions on Software Engineering*, vol. 47, no. 3, pp. 522–559, Mar. 2021, doi: 10.1109/TSE.2019.2900245.

[29] J. D. Angrist and J.-S. Pischke, *Mostly harmless econometrics: An empiricist’s companion*. Princeton, NJ, USA: Princeton University Press, 2008. doi: 10.1515/9781400829828.

[30] R. Bapna, J. Ramaprasad, and A. Umyarov, “Monetizing freemium communities: Does paying for premium increase social engagement?,” *MIS Quarterly*, vol. 42, no. 3, pp. 719–735, Sep. 2018, doi: 10.25300/MISQ/2018/13592.

[31] J. Wang, G. Li, and K.-L. Hui, “Monetary incentives and knowledge spillover: Evidence from a natural experiment,” *Management Science*, vol. 68, no. 5, pp. 3549–3572, May 2022, doi: 10.1287/mnsc.2021.4048.

[32] M. Blackwell, S. Iacus, G. King, and G. Porro, “Cem: Coarsened exact matching in stata,” *The Stata Journal*, vol. 9, no. 4, pp. 524–546, Dec. 2009, doi: 10.1177/1536867X0900900402.

[33] S. L. Johnson, S. Faraj, and S. Kudaravalli, “Emergence of power laws in online communities: The role of social mechanisms and preferential attachment,” *MIS Quarterly*, vol. 38, no. 3, pp. 795–808, Sep. 2014.

[34] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian, “An in-depth study of the promises and perils of mining GitHub,” *Empirical Software Engineering*, vol. 21, no. 5, pp. 2035–2071, Oct. 2016, doi: 10.1007/s10664-015-9393-5.

[35] Y. Tian, Y. Zhang, K.-J. Stol, L. Jiang, and H. Liu, “What makes a good commit message?,” in *Proceedings of the 44th International Conference on Software Engineering*, Jul. 2022, pp. 2389–2401. doi: 10.1145/3510003.3510205