Appendix - Analytics Dashboards and User Behavior: Evidence from GitHub

Jan Schilpp*, Florian Pethig†, Hartmut Hoehle‡

January 10, 2025

CONTENTS

*University of Mannheim, Germany, jan.schilpp@uni-mannheim.de
†Tilburg University, The Netherlands, f.pethig@tilburguniversity.edu
‡University of Mannheim, Germany, hoehle@uni-mannheim.de

A GITHUB USER FILTERING

To implement the analytics dashboard on their 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.

We filtered the users included in the analysis based on multiple characteristics. First, we limited the sample to personal GitHub accounts of individual developers by removing organizational accounts and bots. This was achieved by filtering users based on their names and an unrealistically high number of contributions (Dey et al., 2020; Moldon et al., 2021). Second, we restricted the sample to users who had at least ten commits within our time frame to ensure active use of GitHub. Third, we narrowed down the sample to developers who adopted the analytics dashboard and did not remove it during our study period. This allows us to compare the impact of the dashboard among adopters. Fourth, we excluded the commits made by users to their own Readme.md. Including these commits could bias our estimations because adopter must modify their Readme.md to implement the dashboard. The collection of this data from GhArchive (https://www.gharchive.org) was only feasible thanks to the Python Multiprocess module (McKerns and Aivazis, 2010; McKerns et al., 2011).

B BERT MODEL PERFORMANCE

A fine tuned BERT model has shown to outperform other approaches in sentiment analysis within the software engineering domain (Wu et al., 2021). To fine tune a BERT model for our data we compiled a ground truth data set that is representative of the GitHub commit messages. This ground truth data set is based on five publicly available data sets and consists of short messages from software development that have been manually labeled as positive, negative, or neutral (Calefato et al., 2018; Lin et al., 2018; Novielli et al., 2020). We calculate a commit

message's sentiment by subtracting the predicted probability of a negative message from the predicted probability of a positive message.

Table A1: Fine Tuned BERT Model Performance

	Precision	Recall	F1
Negative messages	0.80	0.78	0.79
Neutral messages	0.85	0.83	0.84
Positive messages	0.80	0.85	0.82

Note: These performance statistics are calculated based on the test dataset that resulted from a 80/20 split of the assembled ground-truth dataset. The test data encompasses 3,582 short messages. Precision = TP / (TP + FP). Recall = TP / (TP + FN). F1 = 2PR / (P + R). TP = True Positive. FP = False Positive. FN = False Negative. P = Precision. R = Recall.

C SYNTHETIC DID BACKGROUND

For every cohort c our modified synthetic difference-in-differences (synthDiD, based on Berman and Israeli, 2022) algorithm solves the following minimization problem ten times (every time with a new random sample of non-adopters):

$$(\alpha_c, \beta_0, u_i, \tau_t) = \underset{\alpha_c, \beta_0, u_i, \tau_t}{\arg\min} \left\{ \sum_{i \in N_c} \sum_{t=c+m_{\min}}^{c+m_{\max}} \left(y_{it} - \beta_0 - u_i - \tau_t - \text{AfterAdoption}_{it} \cdot \alpha_c\right)^2 \omega_i \lambda_t \right\} \tag{1}$$

where α_c is the average treatment effect on the treated (ATT) of cohort c, β_0 is the intercept, u_i user fixed effects, and τ_t time fixed effects. Moreover, N_c are the units in the balanced panel of cohort c, m_{max} is the number of post-treatment months (5, adoption month is part of post-treatment period) while m_{min} represents the number of pre-treatment months (-6). Eventually, y_{it} is the respective dependent variable, $AfterAdoption_{it}$ indicates if user i adopted the dashboard in month t, and ω_i as well as λ_t represent unit and time weights.

To obtain an overall ATT estimate α showing the average treatment effect in the complete post-treatment period, all 260 (26 adoption months · ten estimation per cohort c) estimates are aggregated as:

$$\alpha = \frac{\sum_c N_c^{\text{tr}} \cdot \alpha_c}{\sum_c N_c^{\text{tr}}} \tag{2}$$

where N_c^{tr} represents the treated users in cohort c.

In the main paper, we differentiate between low- and high-activity users through a median split based on the total number of commits made by each user in the six months preceding the adoption of the analytics dashboard. The cutoff point is dynamically recalculated each month, as commit counts vary across the sample, as shown in Figure A1. The low commit count observed around September 2021 likely results from missing data, which may have been caused by partial query failures with the GitHub API experienced by GhArchive during that period.

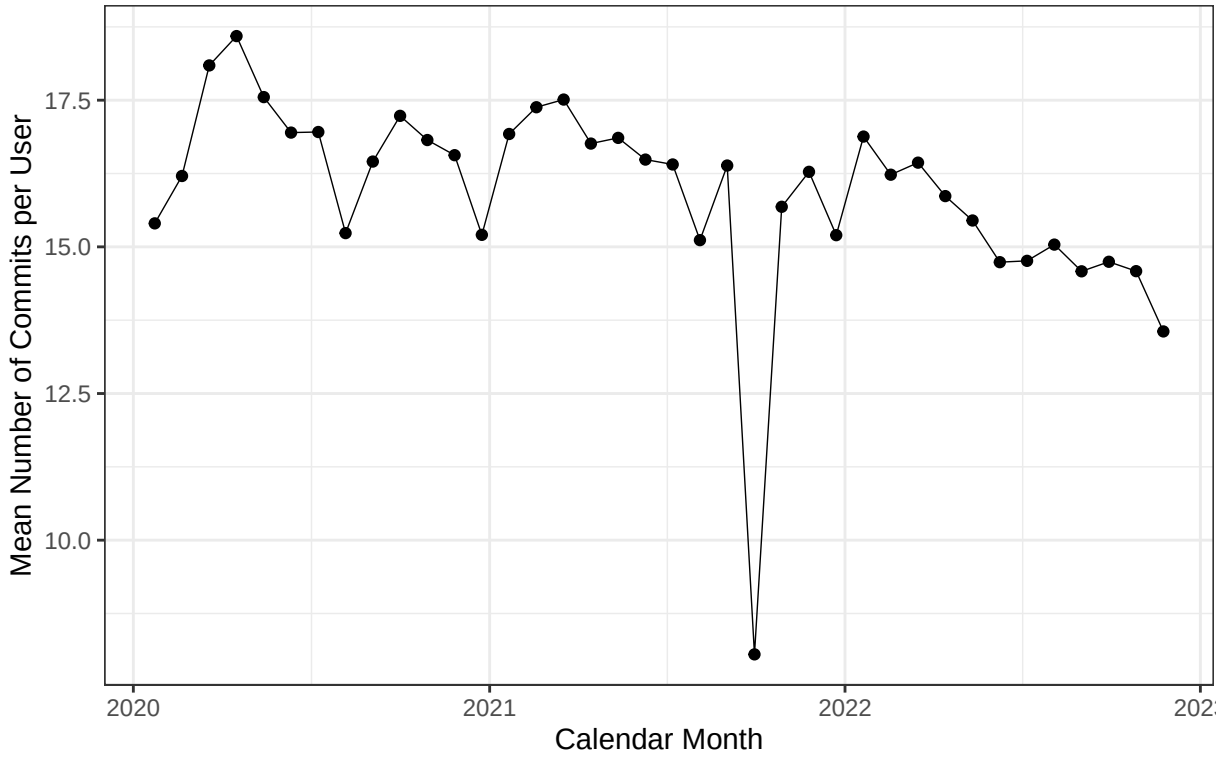


Figure A1: Mean Number of Commits per User and Month

D INSTRUMENTAL VARIABLE APPROACH BACKGROUND

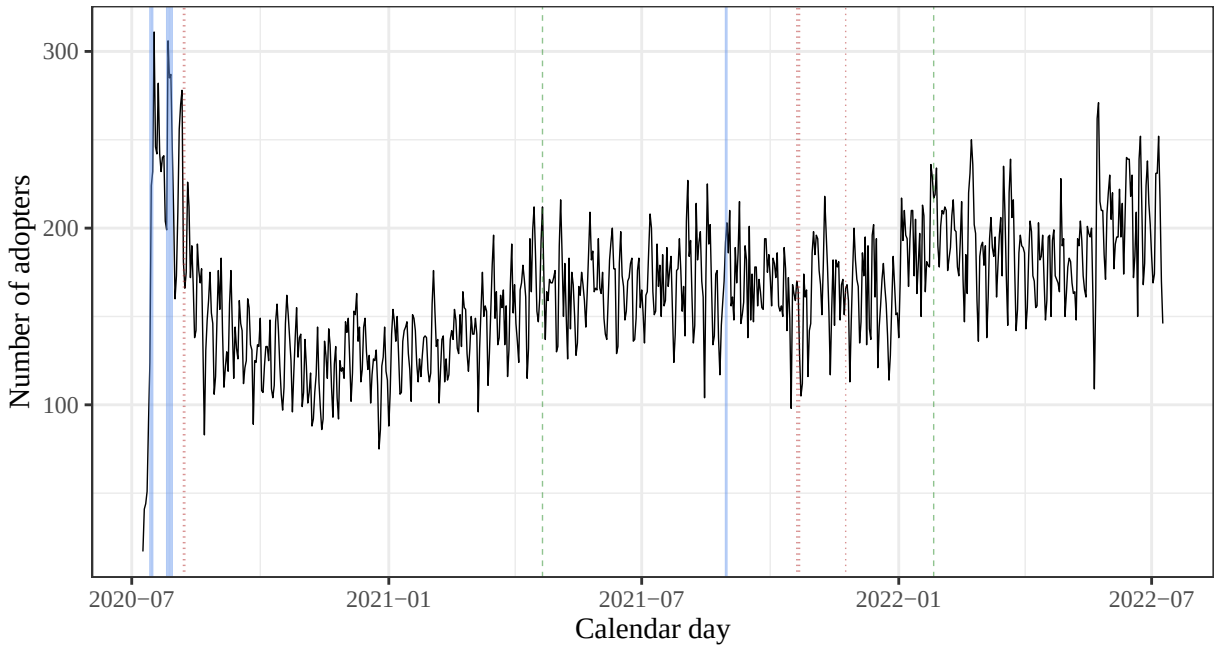
The information on GitHub Trending features was obtained through the Internet Archive's Wayback Machine (https://archive.org/web/). Detailed information about the analytics dash-

board's limited availability is documented in the repository issue discussions (https://github.com/anuraghazra/github-readme-stats/issues/325, https://github.com/anuraghazra/github-readme-stats/issues/1406, https://github.com/anuraghazra/github-readme-stats/issues/1471, Retrieved December 01, 2023). An overview is provided in Table A2. For information on how the exogenous shocks affected the number of adopters see Figure A2.

Table A2: Exogenous Shocks on Analytics Dashboard Visibility and Availability

Calendar day	Exogenoues shock
2020-07-14	Featured on GitHub Trending
2020-07-15	Featured on GitHub Trending
2020-07-16	Featured on GitHub Trending
2020-07-26	Featured on GitHub Trending
2020-07-27	Featured on GitHub Trending
2020-07-28	Featured on GitHub Trending
2020-07-29	Featured on GitHub Trending
2020-07-30	Featured on GitHub Trending
2020-08-07	Limited availability due to technical issues
2020-08-08	Limited availability due to technical issues
2021-04-21	Featured on Catalin's Tech
2021-08-30	Featured on GitHub Trending
2021-08-31	Featured on GitHub Trending
2021-10-20	Limited availability due to technical issues
2021-10-21	Limited availability due to technical issues
2021-10-22	Limited availability due to technical issues
2021-11-24	Limited availability due to technical issues
2022-01-26	Featured on Sitepoint

Table A3 reports the results of a probit regression of the instrumental variables on the users' analytics dashboard adoption. All coefficients are significant which documents the instrumental relevance of the chosen instrumental variables.



The graph shows the number of daily analytics dashboard adopters (including the ones who removed the dashboard again). The solid blue lines mark days when the dashboard was featured on GitHub Trending. The dashed green lines mark days when the dashboard was featured on external blogs. The dotted red lines mark days when the dashboard had limited availability due to technical issues

Figure A2: Analytics Dashboard Adopters per Day

Table A3: Instrument Relevance

	Dependent variable: *Adoption* (Probit)	
	(1) Low	(2) High
TotalDashboardRepositoryInteractions	0.006***	0.006***
	(0.000)	(0.000)
TotalDashboardRepositoryStars	−0.011***	−0.011***
	(0.000)	(0.000)
TotalDashboardRepositoryForks	−0.006***	−0.006***
	(0.000)	(0.000)
DashboardRepositoryFeatured	0.043***	0.039***
	(0.008)	(0.008)
DashboardRepositoryDown	−0.339***	−0.339***
	(0.008)	(0.008)
Observations	163,722	166,344
Pseudo R2	0.040	0.040

Note: Activity levels are displayed in the second row. Users are defined as high-activity if the number of commits during the pre-treatment period is equal to or above the sample median. Because the instruments only approximate adoption (not continuous use) we include only the adoption month from the after-treatment time window. Furthermore, we include only data from the time when the analytics dashboard repository was already created, i.e., July 09, 2020. The constant is not displayed. Robust standard errors in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

E ALTERNATIVE SPECIFICATION

E.1 Identification Strategy

As alternative to the synthDiD employed in the main paper, we repeat our analysis leveraging a two-way fixed effects specification where we compare the behavior of analytics dashboard adopters with the behavior of a suitable control group, i.e., a matched sample of non-adopters whom the adopters are following (Angrist and Pischke, 2008). We deploy coarsened exact matching (CEM) based on the logged number of commits per month (*NumCommits*), the logged mean message length per commit and month (*MesLength*), the logged number of unique words in commit messages per month (*MesUniqueWords*), and the mean sentiment per commit message and month (*MesSentiment*), and logged tenure in months (*Tenure*). CEM has shown to perform better than other prominent matching methods such as propensity score matching (Bapna et al., 2018; Blackwell et al., 2009; Wang et al., 2022). We opt for one-on-one matching based on the mahalanobis distance of adopters and non-adopters. This matching is based on the activity of adopters and non-adopters six months before the adoption to account for the possibility that the adopters might have changed their contribution behavior before adopting the analytics dashboard (Wang et al., 2022).

Table A4 displays the summary statistics for the main variables of interest of a random matched sample and CEM sample of adopters and non-adopters during the matching month, which is six months before the adoption of the analytics dashboard. Due to the staggered adoption of the analytics dashboard, a random matched sample, where each adopter is randomly matched to one non-adopter, serves as a suitable benchmark to assess the performance of CEM. T-tests are used to evaluate the statistical significance of mean differences between adopters and non-adopters, and the results show that CEM performs well in constructing a comparable control group of non-adopters. In the random matched sample, four variables exhibit significant differences between the two

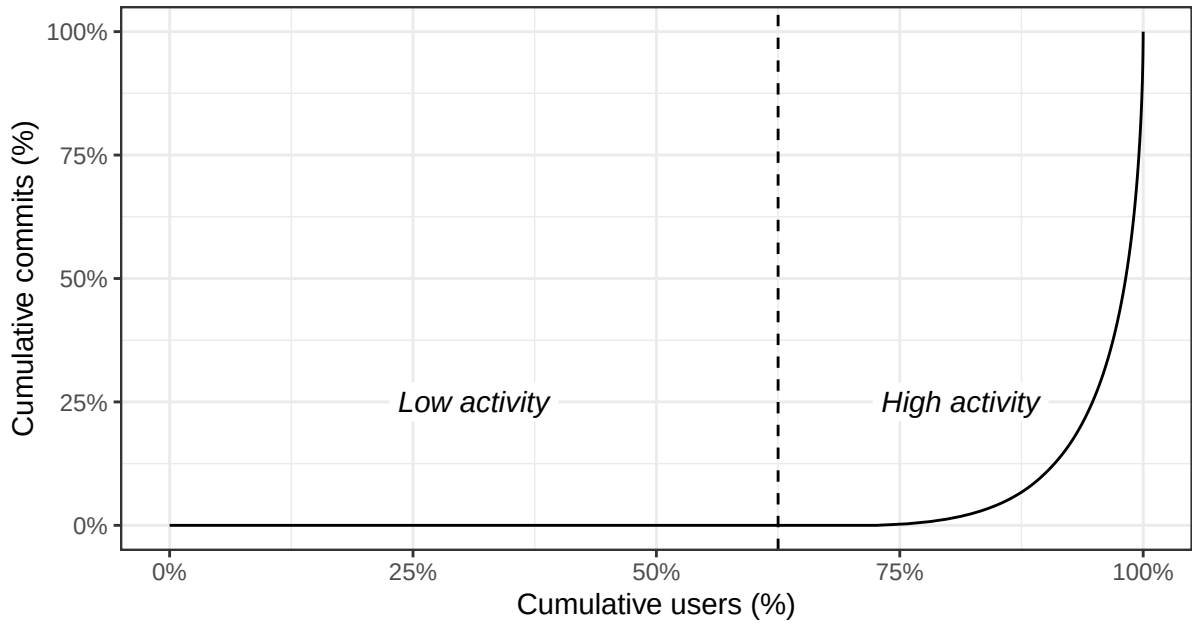
groups at a 99% level, whereas there is only one significant difference observed in the CEM sample (*MesLength*, $p < 0.05$). Based on this, we conclude that our CEM sample of non-adopters is a suitable control group for the adopters.

Table A4: Summary Statistics

	Adopters				Non-adopters				
Variable	Mean	SD	Min	Max	Mean	SD	Min	Max	t-statistics
Random matching									
NumCommits	0.834	1.421	0.000	8.992	0.739	1.374	0.000	8.694	21.435***
MesLength	3.311	0.717	0.000	9.337	3.546	0.841	0.000	9.082	−52.346***
MesUniqueWords	−0.001	0.084	−0.999	0.999	−0.001	0.094	−0.999	0.999	0.594
MesSentiment	3.254	1.398	0.693	9.998	3.448	1.545	0.693	9.547	−23.086***
Tenure	3.234	0.998	0.693	5.198	3.764	1.002	0.693	5.220	−167.875***
CEM									
NumCommits	0.758	1.352	0.000	7.330	0.759	1.350	0.000	7.158	−0.099
MesLength	3.298	0.659	0.693	8.530	3.307	0.658	0.693	6.929	−2.257**
MesUniqueWords	−0.001	0.051	−0.999	0.999	0.000	0.052	−0.999	0.999	−0.777
MesSentiment	3.207	1.367	0.693	9.038	3.211	1.370	0.693	8.747	−0.500
Tenure	3.253	0.992	0.693	5.198	3.252	1.000	0.693	5.198	0.254

Note: The table displays variable summary statistics for a random (CEM) matched sample during the matching month (i.e., 6 months before adoption). N = 67,082 (63,635) Adopters, 67,082 (63,635) Non-adopters. The variables *NumCommits*, *MesLength*, *MesUniqueWords*, and *Tenure* are logged. SD = Standard deviation. The t-statistics are the result of a t-test testing the statistical significance of mean differences. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Furthermore, we leverage the users' number of commits during the matching month to differentiate low and high-activity developers based on a split along the 60th percentile (see Figure A3). We choose to analyze data from five months before and after the adoption to investigate the lasting impact of the analytics dashboard. Additionally, we remove the adoption month as well as one month before and after to ensure that our results are not driven by short-term effects (cf., Foerderer, 2020). Applying the outlined matching procedure, we obtain a sample of 127,270 users (63,635 adopters and 63,635 non-adopters) with 1,145,430 user-month observations. Eventually, we normalize the months relative to the adopters' month of adoption to account for the staggered adoption of the analytics dashboard.



This data is based on the complete sample, i.e., 47,594 analytics element adopters (between 2020−07−09 and 2022−06−09) and a random sample of 103,333 users the adopters are following. All of the users must have been created before the relevant month. This is the data for the first month in the dataset, i.e. 2020−01−23.

Figure A3: Cumulative Users and Commits

E.2 Results

Our models estimate users' number of commits, mean commit message length, number of unique words in commit messages, and mean commit message sentiment per month. To facilitate interpretation and understanding, we use a ordinary least squared (OLS) regressions for all dependent variables. Our regression specification is

$$y_{it} = \beta_0 + \beta_1 Adoption_i x After_{it} + \beta_2 Tenure_{it} + \beta_3 Adopters\ in\ network_{it} + u_i + \tau_t + \epsilon_{it} \quad (3)$$

where the dependent variable y_{it} denotes user i's logged number of commits (*NumCommits*, H1 in the main paper), logged mean commit message length (*MesLength*, H2 in the main paper), logged number of unique words in commit messages (*MesUniqueWords*, H3 in the main paper), or mean commit message sentiment (*MesSentiment*, H4 in the main paper) in calendar month t. β_0 represents the constant of the OLS regression. The variable $Adoption_i$ is a dummy indicating whether user i is part of the adopter or non-adopter group (1 = adopter group) and $After_{it}$ indicates

whether the observation month is part of the pre- or post-adoption period (1 = post-adoption). Their interaction $Adoption_i x After_{it}$ is our DiD term. Thus, β_1 is our DiD estimator showing how adopting the analytics dashboard influences the respective dependent variable. With $Tenure$ we control for user i's tenure in logged months on GitHub for month t. *Adopters in network* is our second control variable and represents the percentage of users in developer i's network who adopted the analytics dashboard in month t. By adding this variable we control for potential spillover effects, i.e., that the adoption of one user affects the behavior of another user, and guarantee that our results are unbiased even if such spillover effects exist (cf., stable unit treatment values assumption; Sinclair et al., 2012). Furthermore, the number of commits are highly user- and time-dependent (Guzman et al., 2014; Kalliamvakou et al., 2016). Therefore, we add user fixed effects u_i as well as time fixed effects τ_t that capture calendar month effects as well as relative month effects being the distance between observation and adoption month. ϵ_{it} represents the error term. The coefficients for the variables $Adoption_i$ and $After_{it}$ are only present in their interaction because the variables perfectly correlate with the user or time fixed effects respectively. To adequately test our hypotheses we run separate regressions for users with initial low and high activity.

Table A5 presents the results of the corresponding regressions. The results from the two regressions on users' number of commits are displayed in column 1 and 2. The DiD coefficient (*Adoption x After*) is positive and significant ($p<0.01$) for low and high-activity users indicating that users commit more after adopting the analytics dashboard. These findings suggest that adopters with low initial activity commit approximately 27% more after the adoption, while adopters with high initial activity commit about 16% more. Columns 3 and 4 present the results of the regressions on the mean commit message length. Both DiD estimators are positive and significant ($p<0.01$), indicating that users write longer commit messages after adopting the analytics dashboard. Specifically, the mean number of characters per message for both groups increases by

approximately 3% after the adoption of the analytics dashboard. Next, the results in column 5 and 6 display the results of the regressions on the unique words in commit messages. Again, both DiD coefficients are positive and significant indicating an increase by about 8% and 11%, for low- and high-activity adopters respectively. Eventually, we examine the impact of adopting the analytics dashboard on the commit message sentiment. The results of the corresponding OLS regressions are presented in columns 7 and 8. While the DiD estimator for users with high initial activity is positive but not statistically significant, the one for users with low initial activity is negative, with a point estimate of -0.002.

Table A5: Difference-in-Differences

	NumCommits		MesLength		MesUniqueWords		MesSentiment	
	(1) Low	(2) High	(3) Low	(4) High	(5) Low	(6) High	(7) Low	(8) High
Adoption x After	**0.270*** **	**0.164*** **	**0.033*** **	**0.027*** **	**0.078*** **	**0.105*** **	**−0.002**	**0.001**
	(0.009)	**(0.013)**	**(0.007)**	**(0.006)**	**(0.015)**	**(0.012)**	**(0.001)**	**(0.001)**
Tenure	0.268***	−0.531***	−0.017	0.017	0.086***	−0.089***	−0.002	−0.003*
	(0.019)	(0.032)	(0.015)	(0.013)	(0.032)	(0.030)	(0.002)	(0.001)
Adopters in network	0.038***	0.016	0.009	0.017	0.003	0.018	0.000	−0.001
	(0.012)	(0.025)	(0.012)	(0.012)	(0.024)	(0.026)	(0.001)	(0.002)
User Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Month Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Observations	637,392	380,768	219,756	242,530	220,128	242,804	219,742	242,528
Users	79,674	47,596	63,762	45,814	63,787	45,818	63,756	45,814
Adj. R2	0.037	0.038	0.004	0.003	0.014	0.010	0.000	0.000
Mean Adopters Pre	8.474	33.210	34.289	40.038	49.128	99.554	0.000	−0.001

Note: The dependent variables are displayed in the first row. The means of adopters before for Number of commits and Commit message length are not logged. Users' activity level during the matching month according to a split along the 60th percentile of the number of commits are displayed in the second row. The constants of the OLS regressions are not displayed. Robust standard errors in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

The DiD estimators in our main specification vary in magnitude and significance between low- and high-activity users for all three dependent variables. To assess whether the adoption of the analytics dashboard has different impacts on users based on their initial activity levels, we follow the approach of Oberfichtner and Tauchmann (2021), who develop a test to determine whether the coefficients from two regressions with different subsamples significantly differ.

This test involves running a single regression using stacked data from the distinct OLS regressions (building on Wooldridge, 2010), making it compatible with panel data and fixed effects. The test then compares the outcomes against a chi-squared distribution, allowing one to determine whether differences in the coefficients are statistically significant. If the test reveals significant differences in the DiD estimators, this implies that low and high activity users are affected differently from the analytics dashboard adoption.

Table A6 present the corresponding results. The differences in the DiD coefficients between low and high activity users in the regressions on number of commits and commit message sentiment are statistically significant ($p < 0.01$ and $p < 0.1$), while there is no significant difference in the DiD coefficients in the regressions on commit message length and unique words in commit messages. These findings clearly support our hypotheses H1 (number of code contributions) and H4 (developer sentiment) postulated in the main paper suggesting that the adoption of the analytics dashboard exerts a greater effect on users with low initial activity. Specifically, adopters with low initial activity commit about eleven percentage points more and write more negative commit messages after adopting the analytics dashboard compared to adopters with high initial activity.

Table A6: Difference-in-Differences Estimator Differences

	NumCommits	MesLength	MesUniqueWords	MesSentiment
Adoption x After (Chi2)	47.463***	0.469	2.002	3.079*

Note: Compared against a Chi2 distribution, the computed values show the statistical significance of the DiD estimator *Adoption x After* differences between low and high activity users (following Oberfichtner and Tauchmann, 2021). *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Overall, the alternative two-way fixed effects model with a matched sample of adopters and non-adopters corroborates the findings from the synthDiD and instrumental variable approach presented the main paper. They suggest that the analytics dashboard adoption positively impacts the number of commits (supporting H1), the commit message lenth (rejecting H2) and the unique

words in commit messages (rejecting H3), but low-activity users write more negative commit messages (partly supporting H4).

F BTM BACKGROUND

The commonly used Latent Dirichlet Allocation (LDA) algorithm for topic modeling does not perform well when applied to a short text corpus because it relies on word co-occurrences between messages, which requires longer messages to deliver a good topic model (Mazarura and de Waal, 2016; Weisser et al., 2022). In contrast, biterm topic modelling (BTM) relies on biterm co-occurrences across messages, thereby solving the problem of extracting meaningful topics from texts that consist of only a few words (Yan et al., 2013). Furthermore, BTM has been shown to outperform other topic modeling approaches applicable to short texts, such as BERTopic (Miyazaki et al., 2023).

It is reasonable to treat the commit messages of adopters and non-adopters before and after the adoption as one corpus because, due to the complexity of text data and the stochastic nature of topic models, there is no one single topic model for a corpus of text. Thus, running a topic modeling approach with separate corpora (e.g., one topic model for a corpus of commit messages by adopters before and one after the adoption of the analytics dashboard) to capture topic shifts over time, might lead to entirely different topic models that are not comparable across corpora (Furman and Teodoridis, 2020).

Determining the optimal number of topics is a crucial step in every topic modeling process. To find the best fit, we ran separate models with topic counts ranging from three to ten and selected the final model with T = 8. This selection was made after carefully evaluating and balancing the commonly used three key metrics: coherence (maximized), perplexity (minimized), and entropy (minimized). Table A7 displays the complete metrics for the computed topic models.

Table A7: BTM Performance with Different Numbers of Topics

T	Coherence	Perplexity	Entropy
3	−639.38	405.22	3.98
4	−663.38	375.04	3.57
5	−658.49	354.01	3.35
6	−654.83	345.10	3.23
7	−670.75	335.40	3.15
8	**-649.69**	**316.56**	**3.10**
9	−660.11	318.70	3.05
10	−669.50	305.34	3.02

Note:
T = Number of Topics. The topic models are run on the complete corpus of commit messages by low-activity users.

G SYNTHETIC DID BY TOPICS ALL DEPENDENT VARIABLES

Tables A8, A9, and A10 present the regression results segmented by topic for low-activity users for the dependent variables not included in the main paper. Tables A9 and A10 offer intriguing insights into the unexpected positive effects of the analytics dashboard adoption on the commit message length and the number of unique words in commit messages. Notably, both variables show the steepest increase for Topic 8, "branch management" (0.141, $p < 0.01$, and 0.322, $p < 0.01$).

This suggests that the overall rise in the commit message length and the unique words in commit messages is likely driven by an increased documentation effort necessitated by the higher number of commits. Put differently, as the analytics dashboard incentives low-activity developers to commit more frequently, a greater number of software versions is created. As a result, the need to organize and track these versions increases, requiring more detailed commit messages to enable developers to retrospectively monitor their progress. For example, if a developer commits only once a week, it is relatively easy to remember the implemented changes without extensive documentation. However, as the number of commits rises, more comprehensive documentation becomes essential to track progress and fully understand the changes made. Similarly, the need for branch management increases

Table A8: Number of Commits Synthetic Difference-in-Differencey by Topics

	Core OSS Activities			Peripheral OSS Activities			
	(1) Topic 1	(2) Topic 2	(3) Topic 3	(4) Topic 5	(5) Topic 6	(6) Topic 7	(7) Topic 8
Adoption x After	**0.191*** **	**0.250*** **	**0.253*** **	**0.233*** **	**0.209*** **	**0.149*** **	**0.218*** **
	(0.004)	**(0.005)**	**(0.004)**	**(0.005)**	**(0.004)**	**(0.003)**	**(0.004)**
Observations	241,416	241,416	241,416	241,416	241,416	241,416	241,416
Users	20,118	20,118	20,118	20,118	20,118	20,118	20,118
Mean Adopters Pre	2.931	4.318	2.601	8.885	3.156	1.658	2.024

Note: The dependent variable is the logged number of commits separate by topics. The topic identified by Biterm Topic Modelling is displayed in the top row. Only initially less active users are considered in the regressions. Users are defined as low-activity if the number of commits during the pre-treatment period is below the sample median. The pre-treatment means of adopters are presented in their raw form (not logged). Robust standard errors in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A9: Commit Message Length Synthetic Difference-in-Differencey by Topics

	Core OSS Activities			Peripheral OSS Activities			
	(1) Topic 1	(2) Topic 2	(3) Topic 3	(4) Topic 5	(5) Topic 6	(6) Topic 7	(7) Topic 8
Adoption x After	**0.017**	**0.023*** **	**0.102*** **	**0.048*** **	**−0.024*** **	**0.049*** **	**0.141*** **
	(0.011)	**(0.005)**	**(0.011)**	**(0.003)**	**(0.008)**	**(0.014)**	**(0.016)**
Observations	20,016	37,440	16,920	85,032	18,072	8,064	9,720
Users	1,668	3,120	1,410	7,086	1,506	672	810
Mean Adopters Pre	61.284	31.241	44.997	17.995	42.540	56.954	133.787

Note: The dependent variable is the logged commit message length separate by topics. The topic identified by Biterm Topic Modelling is displayed in the top row. Only initially less active users are considered in the regressions. Users are defined as low-activity if the number of commits during the pre-treatment period is below the sample median. The pre-treatment means of adopters are presented in their raw form (not logged). Robust standard errors in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A10: Unique Words in Commit Messages Synthetic Difference-in-Differencey by Topics

	Core OSS Activities			Peripheral OSS Activities			
	(1) Topic 1	(2) Topic 2	(3) Topic 3	(4) Topic 5	(5) Topic 6	(6) Topic 7	(7) Topic 8
Adoption x After	**0.157*** **	**0.151*** **	**0.235*** **	**0.208*** **	**0.183*** **	**0.201*** **	**0.322*** **
	(0.017)	**(0.010)**	**(0.018)**	**(0.007)**	**(0.016)**	**(0.025)**	**(0.021)**
Observations	20,016	37,440	16,920	85,032	18,072	8,064	9,720
Users	1,668	3,120	1,410	7,086	1,506	672	810
Mean Adopters Pre	46.579	19.357	25.133	17.624	29.985	25.680	33.000

Note: The dependent variable is the logged number of unique words in commit messages separate by topics. The topic identified by Biterm Topic Modelling is displayed in the top row. Only initially less active users are considered in the regressions. Users are defined as low-activity if the number of commits during the pre-treatment period is below the sample median. The pre-treatment means of adopters are presented in their raw form (not logged). Robust standard errors in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

H QUALITATIVE EVIDENCE

To further understand the underlying mechanisms that drive developers' behavior changes after adopting the analytics dashboard, we additionally conducted 32 semi-structured interviews with analytics dashboard adopters. Among these interviews, 16 involved individuals with low initial activity, while the other 16 were part of the high activity group. The users in the study adopted the analytics dashboard either in May or June 2022, and the interviews took place between May and June 2023.

26 developers stated that the analytics dashboard had a positive impact on their contribution behavior. This does not only encompasses the previously observed higher number of commits but also a more desirable commit frequency without harming code quality.

> "I used to write codes, but don't push them usually. But nowadays, I try to commit sincerely and try to write meaningful commits whenever I commit and try to organize my codes, and everything I try to look like a professional." (Developer 19, low initial activity)

> "No, I don't think it [the quality or complexity of the code] changed that much [after adopting the analytics dashboard]." (Developer 4, high initial activity)

Furthermore, twelve developers explained that the analytics dashboard led them to engage in upward social comparison which impacted their contribution behavior and well-being.

> "I'm probably amazed first [when visiting the GitHub profile of a developer with higher numbers in the analytics dashboard]. [...] Then I kind of kick myself in the butt that you need to hurry up and learn more and do better stuff. [...] But then those people if you if you see their projects and their contributions, they are usually contributors and maintainers of an amount of open source project. And sometimes the code that they write or the language they use is just so low level." (Developer 26, low initial activity)

Eventually, five respondents (three with low and two with high initial activity) stated that they encountered stress due to the upward social comparison enabled by the analytics dashboard. Particularly interesting are the experiences of an adopter whose emotional response changed depending on their life situation. Initially, they were an intern as their main occupation and

felt stressed when comparing themselves to more active developers on GitHub, but they do not experience these negative emotions any more because now they work as a full-time employee.

> "I think I definitely have worried that [the] dashboard, something that shows raw stats will show that I may be not as experienced of a developer as other people who've been longer out in the industry. I just graduated about a month ago and so I've been working full time for a month, before that, I was an intern." (Developer 4, high initial activity)

This example shows that developers' emotional response to the upward social comparison enabled by the analytics dashboard highly depends on their personal situation. Specifically, if one cares about the comparison, perceives oneself at a disadvantage, and is unsure about their own abilities, this may result in a higher level of stress.

Overall, the qualitative evidence corroborates and enriches the findings from the quantitative data, indicating that adopting the analytics dashboard positively affects developers' number of commits as well as their professional appearance. However, the analytics dashboard may adversely impact developers' well-being, especially those with less experience and higher uncertainty about their abilities, as they may experience stress induced by excessive upward social comparison.

REFERENCES

Angrist, J. D., & Pischke, J.-S. (2008). *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press. https://doi.org/10.1515/9781400829828

Bapna, R., Ramaprasad, J., & Umyarov, A. (2018). Monetizing freemium communities: Does paying for premium increase social engagement? *MIS Quarterly*, *42*(3), 719–735. https://doi.org/10.25300/MISQ/2018/13592

Berman, R., & Israeli, A. (2022). The value of descriptive analytics: Evidence from online retailers. *Marketing Science*, *41*(6), 1074–1096. https://doi.org/10.1287/mksc.2022.1352

Blackwell, M., Iacus, S., King, G., & Porro, G. (2009). CEM: Coarsened exact matching in Stata. *The Stata Journal*, *9*(4), 524–546. https://doi.org/10.1177/1536867X0900900402

Calefato, F., Lanubile, F., Maiorano, F., & Novielli, N. (2018). Sentiment polarity detection for software development. *Empirical Software Engineering*, *23*(3), 1352–1382. https://doi.org/10.1007/s10664-017-9546-9

Dey, T., Mousavi, S., Ponce, E., Fry, T., Vasilescu, B., Filippova, A., & Mockus, A. (2020). Detecting and characterizing bots that commit code. *Proceedings of the 17th International Conference on Mining Software Repositories*, 209–219. https://doi.org/10.1145/3379597.3387478

Foerderer, J. (2020). Interfirm exchange and innovation in platform ecosystems: Evidence from Apple's worldwide developers conference. *Management Science*, *66*(10), 4772–4787. https://doi.org/10.1287/mnsc.2019.3425

Furman, J. L., & Teodoridis, F. (2020). Automation, research technology, and researchers' trajectories: Evidence from computer science and electrical engineering. *Organization Science*, *31*(2), 330–354. https://doi.org/10.1287/orsc.2019.1308

Guzman, E., Azócar, D., & Li, Y. (2014). Sentiment analysis of commit comments in GitHub: An empirical study. *Proceedings of the 11th Working Conference on Mining Software Repositories*, 352–355. https://doi.org/10.1145/2597073.2597118

Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D. M., & Damian, D. (2016). An in-depth study of the promises and perils of mining GitHub. *Empirical Software Engineering*, *21*(5), 2035–2071. https://doi.org/10.1007/s10664-015-9393-5

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

Mazarura, J., & de Waal, A. (2016). A comparison of the performance of latent dirichlet allocation and the dirichlet multinomial mixture model on short text. *2016 Pattern Recognition*

Association of South Africa and Robotics and Mechatronics International Conference, 1–6. https://doi.org/10.1109/RoboMech.2016.7813155

McKerns, M. M., & Aivazis, M. (2010). *Pathos: A framework for parallel graph management and execution in heterogeneous computing*. pathos. Retrieved February 2, 2023, from https://mmckerns.github.io/project/pathos/wiki.html

McKerns, M. M., Strand, L., Sullivan, T., Fang, A., & Aivazis, M. A. G. (2011). Building a framework for predictive science. *Proceedings of the 10th Python in Science Conference (SciPy 2011)*. https://doi.org/10.48550/arXiv.1202.1056

Miyazaki, K., Murayama, T., Uchiba, T., An, J., & Kwak, H. (2023). *Public perception of generative AI on twitter: An empirical study based on occupation and usage*. arXiv: 2305.09537 [cs]. https://doi.org/10.48550/arXiv.2305.09537

Moldon, L., Strohmaier, M., & Wachs, J. (2021). How gamification affects software developers: Cautionary evidence from a natural experiment on GitHub. *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 549–561. https://doi.org/10.1109/ICSE43902.2021.00058

Novielli, N., Calefato, F., Dongiovanni, D., Girardi, D., & Lanubile, F. (2020). Can we use se-specific sentiment analysis tools in a cross-platform setting? *Proceedings of the 17th International Conference on Mining Software Repositories*, 158–168. https://doi.org/10.1145/3379597.3387446

Oberfichtner, M., & Tauchmann, H. (2021). Stacked linear regression analysis to facilitate testing of hypotheses across OLS regressions. *The Stata Journal*, *21*(2), 411–429. https://doi.org/10.1177/1536867X211025801

Sinclair, B., McConnell, M., & Green, D. P. (2012). Detecting spillover effects: Design and analysis of multilevel experiments. *American Journal of Political Science*, *56*(4), 1055–1069. https://doi.org/10.1111/j.1540-5907.2012.00592.x

Wang, L., Lowry, P. B., Luo, X., & Li, H. (2022). Moving consumers from free to fee in platform-based markets: An empirical study of multiplayer online battle area games. *Information Systems Research*, *34*(1), 275–296. https://doi.org/10.1287/isre.2022.1127

Weisser, C., Gerloff, C., Thielmann, A., Python, A., Reuter, A., Kneib, T., & Säfken, B. (2022). Pseudo-document simulation for comparing LDA, GSDMM and GPM topic models on short and sparse text using Twitter data. *Computational Statistics*, *38*, 647–674. https://doi.org/10.1007/s00180-022-01246-z

Wooldridge, J. M. (2010). *Econometric analysis of cross section and panel data*. MIT Press.

Wu, J., Ye, C., & Zhou, H. (2021). BERT for sentiment classification in software engineering. *2021 International Conference on Service Science*, 115–121. https://doi.org/10.1109/ICSS53362.2021.00026

Yan, X., Guo, J., Lan, Y., & Cheng, X. (2013). A biterm topic model for short texts. *Proceedings of the 22nd International Conference on World Wide Web*, 1445–1456. https://doi.org/10.1145/2488388.2488514