The Effects of Toxicity on Disengagement in Open Source Projects

Ariel Kamen, Vijeth K L, Saisha Shetty, Thrisha Kopula, Kunal Pai

1 Introduction

Open source projects thrive on the active participation and commitment of developers, who play a pivotal role in shaping the success and sustainability of these collaborative initiatives. It is known that developers contribute to open source in waves and that toxic behavior reduces productivity in the workplace [4, 7]. The fragility of developer engagement poses a significant challenge to the longevity and effectiveness of open-source projects. Recent studies indicate that 41% of the failed open-source projects attribute their demise to issues related to the developer team, including factors such as a lack of interest or time from the main contributors [6]. Developer disengagement, therefore, emerges as a widely acknowledged and critical concern, bearing substantial economic and operational consequences [6, 25].

Given the multifaceted nature of developer disengagement, it becomes imperative to explore the factors contributing to this phenomenon. While some reasons for disengagement may be intrinsic and unavoidable, others may be mitigated through proactive measures, particularly those aimed at enhancing community support. Our research seeks to address the overarching question of whether toxicity in communications within open-source communities could be a significant driver of developer disengagement, thereby hindering the collaborative and innovative potential of these projects, while also answering certain questions about how toxicity propagates in a community.

Our research is guided by specific goals: to quantify the extent of toxicity within selected open-source projects, to assess its correlation with developer engagement, and to analyze the potential for community-driven policies to mitigate such negative interactions. Our research questions are structured to explore the nature of toxic behavior, its prevalence, and its impact on the continuity of developer contributions.

To answer questions about the propagation of toxicity, we are particularly interested in the psychological phenomenon of the "Chameleon Effect", which is the non-conscious mimicry of the behaviors of one's interaction partners, such that one's behavior passively and unintentionally changes to match that of others in one's current social environment [5]. We want to see if the sentiment of a comment has the potential to influence the sentiment of the subsequent comment, and thus propagate toxicity through the community.

Our research also explores potential differences in the extent and consequences of toxicity and disengagement among distinct groups of developers, including those from various programming communities, large personal projects, and those contributing to open-source projects associated with specific companies. Since the extent of the code of conduct is different for these projects [15], their interactions may correlate differently to disengagement and even to sentiment in general.

Considering all of these we decided to answer the following research questions:

Research Questions

1. What is disengagement, and how can we effectively measure it?
2. What are the trends for sentiment and toxicity in company and non-profit open-source repositories?
3. Is the sentiment of a commit message related to the sentiment of the previous comment?
4. Is there a relation between the sentiment of a commit comment and disengagement in company and non-profit repositories?
5. Does the effect of the sentiment vary across different developer groups?

By addressing these research questions, we aspire to provide valuable insights into the dynamics of open-source communities, offering a foundation for the need to develop strategies to mitigate toxicity and foster a more inclusive and collaborative environment. Through a comprehensive analysis, we aim to contribute to the broader understanding of the challenges associated with developer disengagement in open-source projects and, in turn, inform the development of effective measures to enhance the sustainability and success of these collaborative endeavors.

The code for our project is available at https://github.com/kunpai/ecs-260-project-toxicity-disengagement.

2 Background Work

Prior work has attempted to study contributor disengagement in open-source communities using a mixed-methods methodology [19]. This study identified disengaged developers from a dataset, sent out a survey to them, and validated the results using a survival model. The authors found out that the type of work a developer does on a team does not have a statistically significant impact on disengagement, but a transition in their life, like a job, has the most impact on disengagement from a project. Since the results of this study are from a survey, there is the threat of selection bias since developers who did not answer in the survey may have had different reasons for disengaging. Moreover, the potential impact of toxicity on disengagement was not highlighted, and the extent of the impact of disengagement on different "groups" of developers was also not studied. We would attempt to answer these questions through our quantitative study. Another work has attempted to uncover the reasons behind toxicity in open-source discussions [18]. The authors identified GitHub Issues toxicity as having insulting, arrogant, entitled, trolling, or unprofessional tonalities. They also identified failed use of the code, technical disagreement, ideologies, and past interaction reasons as behind toxicity. The study mentions insights about the harms of toxicity, but it does not delve deeply into the long-term impact on community dynamics, contributor retention, or project sustainability. While they touch upon the existence of "past interactions" as a reason for toxicity, this reason is limited to issues where users interact with each other and not the entire repository. We intend to study the long-term impact of toxicity on contributor retention and project sustainability while also seeing if a "past" toxic commit has the potential to make a developer's commit comment also toxic. There has been another study done on commit comments specifically [13]. The authors used a similar methodology to our proposed methodology, using GHTorrent [12] for repository data and SentiStrength [14] for sentiment analysis. They found out that while outliers exist, the overall sentiment of a repository tends to be neutral since comments tend to be more technical. They also found that Java had slightly more negative scores than projects implemented in other languages, the emotion of commit comments on Mondays tends to be more negative, commit comments from the afternoon were significantly more positive, and the number of country locations in a project is positively correlated to the amount of emotion present in positive commit comments. While this paper gives us the basis to investigate different variables, i.e., commit comments from a company's open-source repository versus a non-profit community's open-source repository, it does not highlight what impact negative emotions have on developer engagement and the overall culture of the repository. We intend to fill this gap with our study. Other studies on toxicity have shown a correlation with stress [23]. They claim that more stress results in the tendency to be more toxic. While this is a good investigation into one of the reasons why toxicity exists, we want to analyze if another developer's toxicity could also be one of the reasons why a developer can be toxic. Other studies on disengagement have shown a connection to taking breaks [4]. This paper claims that one of the reasons for disengagement is developers taking breaks. We want to analyze if toxicity is also one of the possible reasons for disengagement. However, none of these studies have tried to connect the two factors and also analyze toxicity itself as a reason for more toxicity. We intend to answer these questions.

3 Data and Methods

We primarily analyze commit messages to answer our research questions. We intend to use PyDriller [26] for extraction and NLTK [3] for sentiment analysis. Since NLTK negative sentiments may not be fully indicative of toxicity, we ran these comments through the OpenAI moderation API to detect if toxicity exists and its category if it does [21], and through a Hugging Face comment classifier model [24] to quantify toxicity and place it in categories. These categories are “Obscene”, “Severe Toxic”, “Insult”, “Threat” and “Identity Hate”.

For the repositories, we cherry-picked repositories so that they cover a wide variety of programming languages and are from different communities. These repositories are as follows:

Microsoft’s VSCode, an open-source project [17], is a Company-owned repository primarily written in TypeScript. Google’s Guava [11], another Company-owned repository, is implemented in Java. Facebook’s React [9], a Company-owned project, utilizes JavaScript as its primary language. Apple’s Swift [1], belonging to the Company, is predominantly coded in C++. The freeCodeCamp repository [10], associated with a non-profit organization, employs TypeScript. gem5 [2, 16], a non-profit project by gem5, is implemented in C++. Additionally, Node.js [20] from the Node non-profit organization uses JavaScript, and JDK [22] from OpenJDK, a non-profit initiative, is primarily coded in Java. These repositories showcase a mix of companies and non-profit organizations contributing to the open-source landscape with diverse programming languages.

We extracted these commits, along with other information like the timestamp of the commit, the SLOC of the change made, the sentiments of the immediate previous commit, what day and time (morning, afternoon, evening, or night) the commit was made, and whether it originated from a pull request or not. Then, we calculated the sentiment of the commit message, along with its toxicity score. We are following the definition of toxicity as “rude, disrespectful, or unreasonable language that is likely to make someone leave a discussion” [8]. To detect that, we passed commit messages through the OpenAI moderation API to flag any potentially toxic comments. This gives us the answer to what sentiment is and how we can effectively measure toxicity from it.

Our rationale for considering the sentiments of the immediately previous commit is based on the GitHub UI. When a developer logs into a repository on GitHub, the first thing they see, along with the directory structure of the root, is the last commit that was made, along with its message and when it was made relative to the time the developer is viewing the repository. So, we want to see if the sentiment of that commit could influence the user working on the repository. An assumption here is that development follows a linear pattern, i.e., a developer looks at the previous commit and the next commit in the repository is their commit, and there are none in between. However, even if there were commits in between, they could still have been influenced by the “previous commit”, and there could be a trickle-down effect of sentiment.

After this step, we sorted the commits on a developer-by-developer basis, in ascending order of the timestamp. Then, we calculated the time difference between two consecutive commits. In the case of the last commit of a developer, we calculated the time difference until the time the program was run, because the repository is still “waiting” for their next commit. This procedure gives us a period of disengagement. Once this dataset was ready, we used correlation on all our variables, with one-hot encoding for categorical variables like days of the week and time. We correlated through the inbuilt `pandas Dataframe corr` method which correlates by the Pearson method. To validate the correlation scores between variables, we performed linear regression analysis to see if we could use a correlated variable as a predictor for sentiment score or time difference.

Apart from measuring the impact of toxicity on an individual developer’s engagement, we also want to expand it to a community-based definition of engagement. While this definition will be expanded upon in the first R.Q., we primarily evaluate community engagement patterns using correlation and linear regression.

For the fifth R.Q., we want to divide the developers based on their commit experience for a repository, and then retry the analyses done for the third and fourth R.Q.s to observe if there is a correlation between the previous commit’s sentiment and their sentiment, and also if individual and community engagement is different for these different groups of developers.

4 Results

R.Q. 1: What is disengagement and how can we effectively measure it?

The measure of disengagement was quantified on two levels: individual (micro) and community-wide (macro). At the individual level, disengagement was quantified by the duration between consecutive commits, with extended intervals suggesting reduced engagement. Our analysis revealed notable periods of inactivity, potentially indicating disengagement. On a community scale, we measured engagement by tracking the number of unique contributors per year, with declining figures suggesting waning community health. This dual framework provides a robust metric for assessing the vitality and sustainability of open-source projects.

We implemented a systematic approach to analyze commits within software repositories. Key steps included sorting commits on a developer-by-developer basis and calculating time differences between consecutive commits. Additionally, we integrated sentiment analysis and toxicity detection algorithms to provide a contextual understanding of commit messages.

R.Q. 2: What are the trends for sentiment and toxicity in company and non-profit open-source repositories?

To address this research question, we analyzed the distribution of negative, positive, and neutral comments in both sets of repositories, as depicted in Figure 1. In company repositories, the sentiment distribution is 57.8% neutral, 25.6% positive, and 16.6% negative. For non-profit community repositories, the distribution is 56.4% neutral, 23% positive, and 20.7% negative. This finding aligns with the results of Guzman et al. [13], indicating a predominance of neutral sentiment in commit comments, primarily due to their technical nature. Notably, company open-source software tends to exhibit a higher proportion of positive sentiment compared to non-profit open-source software. This divergence could be attributed to more stringent guidelines in companies, ensuring commit messages remain free of toxic language.

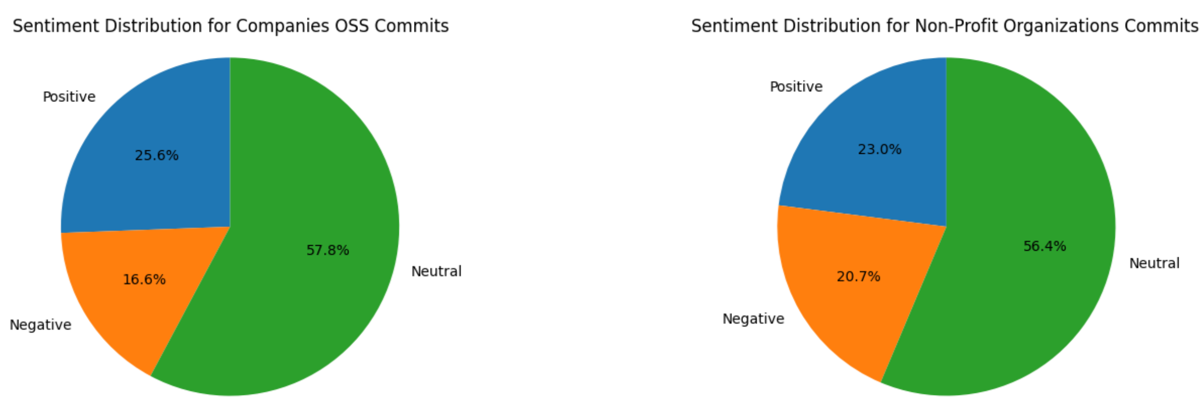


Figure 1: Distribution of sentiments for commit comments for both groups of repositories

Certain toxic comments for company repositories were "`dont be stupid isidor`" and "`a bit less ugly css rule`" from VSCode, "`remove some crazy slashes also`" and "`fix some crazy spacing, we don't force putting spaces around binary operators :-) Swift SVN r14990`" from Swift, among others. Certain toxic comments for non-profit repositories were "`build: fix windows build. Be very careful with forward declarations, MSVC is quite picky and rather stupid about it`" and "`mem: fix dumb typo in copyrights`" from gem5, "`Add small crappy manpage (please improve)`" from NodeJS, among others. The results show that the toxicity in the comments is very low, and the difference between the occurrence of toxicity in the two types of repositories is not significant. The OpenAI moderation API [21] flagged an even lesser number of toxic comments as toxic. This means that very few comments in an entire repository can be classified as harassment, profanity, or toxicity, and the difference between the two types of repositories is not significant. So, the code of conduct and guidelines for contributing to open-source repositories are effective in curbing toxic commit comments from becoming part of their commit history.

About the categories of toxicity that these comments fall into, we made a heatmap for both types of repositories, which can be seen in Figure 2. From the heatmaps, company repositories tend to show toxicity in the 'Obscene', 'Insult', 'Threat', and 'Identity_hate' categories, whereas non-profit repositories lack instances of 'Identity_hate.' Additionally, in the toxicity categories displayed by non-profit repositories, the intensity of toxicity surpasses that observed in company repositories. Non-profit communities may have contributors from diverse backgrounds and cultures, leading to variations in communication styles. The intensity of toxicity observed in non-profit repositories might be a reflection of passionate discussions or disagreements within a more loosely structured environment, while the toxicity observed in company repositories might be a reflection of a more formal and structured environment, with similar communication styles across contributors.

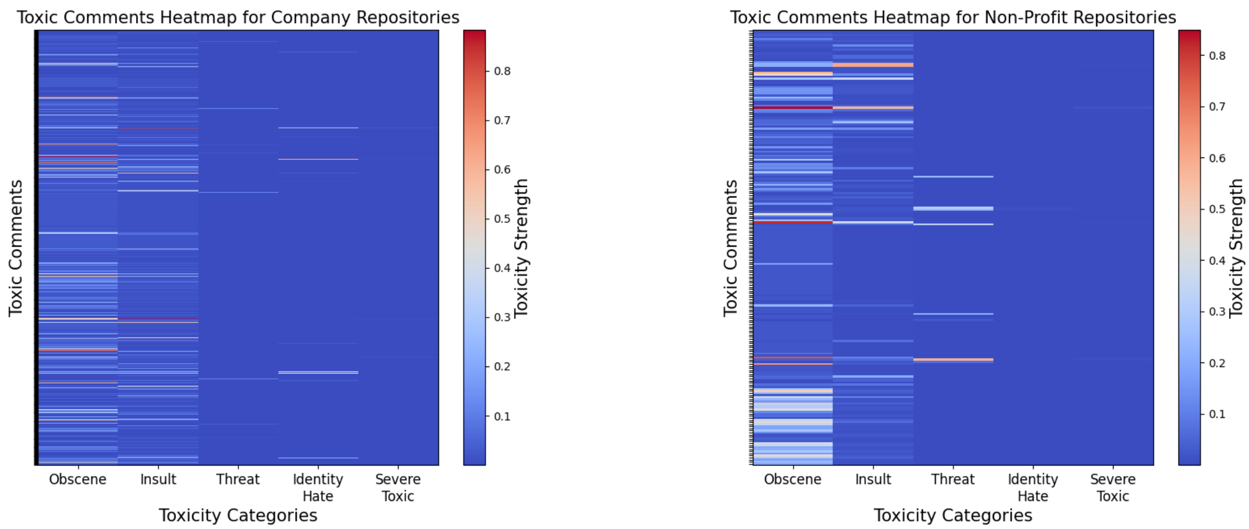


Figure 2: Distribution of toxicity for both groups of repositories

R.Q. 3: Is the sentiment of a commit message related to the sentiment of the previous comment?

We first correlated all the commit messages' NLTK sentiment with the previous commit message's NLTK sentiment and received the correlation matrices as shown in Tables 2 and 3. For non-profit open-source software, the emotion of a particular commit message has a weak positive correlation with the previous commit's sentiment, i.e., the developers in a repository tend to follow the behavior they observe in the commit messages. The correlation is the strongest for positive sentiment, i.e., positivity tends to flow down easier than other sentiments, but that could very well be a marker of the code of conduct of the repositories.

For company repositories, a similar correlation exists for positive and neutral sentiments. However, negative sentiment in the previous commit message doesn't tend to influence a developer's commit message. This could be indicative of either stronger enforcement of the code of conduct or less negativity, to begin with, due to professional behavior being mandated for the employees. SLOC and days and time of day did not have any correlation with sentiment.

To further validate the impact of previous sentiments on the current commit's sentiment, we aggregated the commits by developers and tried to fit sentiment into a regression model.

To investigate the influence of previous sentiments on current commit messages, we aggregated commits by the developer and conducted regression analyses. We attempted to predict the commit message's sentiment based on the previous commit message's sentiment and toxicity.

The coefficients for company repositories reveal that a one-unit increase in the previous sentiment is associated with a modest 0.0622 increase in the current sentiment, while a similar increase in previous toxicity results in a smaller 0.0060 uptick in sentiment. The intercept, representing the baseline sentiment when all variables are zero, is 0.0385. However, the low R^2 value of 0.0034 indicates that the model explains only a minimal portion of the variance in the sentiment score. Consequently, the previous commit message's sentiment and toxicity may have limited predictive power for sentiment, suggesting that other factors or a more comprehensive model may be necessary for a more accurate understanding of sentiment dynamics.

On the other hand, the coefficients for non-profit repositories reveal that a one-unit increase in the previous sentiment is associated with a 0.146 increase in the current sentiment, while a similar increase in previous toxicity results in a 0.236 uptick in sentiment. The intercept representing the baseline sentiment when all variables are zero is 0.0492. However, the low R^2 value of 0.019 indicates that the model explains only a minimal portion of the variance in the sentiment score. Consequently, the previous commit message's sentiment and toxicity may have limited predictive power for sentiment, suggesting that other factors or a more comprehensive model may be necessary for a more accurate understanding of sentiment dynamics.

In comparing the regression analyses of company and non-profit repositories, it becomes evident that the influence of previous sentiments on current commit messages varies between the two contexts. While company repositories show a modest impact of previous sentiment and toxicity, non-profit repositories exhibit more pronounced effects. However, across both scenarios, the limited predictive power of these factors alone suggests the necessity of considering additional variables or adopting a more comprehensive model to achieve a nuanced understanding of sentiment dynamics in software development.

Therefore, our results indicate that the sentiment of a commit message is related to the sentiment of the previous commit message, but the impact is limited. This suggests that other factors may also play a role in shaping the sentiment of commit messages.

	Current Commit NLTK Sentiment				**Toxic**	**Previous Commit NLTK Sentiment**			
	Positive	**Negative**	**Neutral**	**Compound**	**Toxic**	**Positive**	**Negative**	**Neutral**	**Compound**
Positive	1.0	-0.0797	-0.6563	0.5695	0.0055	**0.1624**	-0.0069	-0.1114	0.0845
Negative	-0.0797	1.0	-0.6998	-0.6198	0.0297	-0.0076	**0.1213**	-0.0863	-0.0759
Neutral	-0.6563	-0.6998	1.0	0.0610	-0.0264	-0.1107	-0.0869	**0.1451**	-0.0031
Compound	0.5695	-0.6198	0.0610	1.0	-0.0149	0.0854	-0.0749	-0.0046	**0.1266**
Toxic	0.0055	0.0297	-0.0264	-0.0149	1.0	-0.0039	0.0019	0.0013	-0.0037
Prev_Positive	0.1624	-0.0076	-0.1107	0.0854	-0.0039	1.0	-0.0797	-0.6558	0.5695
Prev_Negative	-0.0069	0.1213	-0.0863	-0.0749	0.0019	-0.0797	1.0	-0.6992	-0.6198
Prev_Neutral	-0.1114	-0.0863	0.1451	-0.0046	0.0013	-0.6558	-0.6992	1.0	0.0610
Prev_Compound	0.0845	-0.0759	-0.0031	0.1266	-0.0037	0.5695	-0.6198	0.0610	1.0

Table 1: Correlation matrix for non-profit open-source repositories

	Current Commit NLTK Sentiment				**Toxic**	**Previous Commit NLTK Sentiment**			
	Positive	**Negative**	**Neutral**	**Compound**	**Toxic**	**Positive**	**Negative**	**Neutral**	**Compound**
Positive	1.0	-0.0318	-0.7155	0.5469	0.0005	**0.1427**	0.0114	-0.1130	0.0843
Negative	-0.0318	1.0	-0.6659	-0.5590	0.0550	0.0096	0.0748	-0.0585	-0.0369
Neutral	-0.7155	-0.6659	1.0	-0.0179	-0.0384	-0.1117	-0.0599	**0.1250**	-0.0367
Compound	0.5469	-0.5590	-0.0179	1.0	-0.0273	0.0823	-0.0348	-0.0366	0.0869
Toxic	0.0005	0.0550	-0.0384	-0.0273	1.0	0.0011	-0.0008	-0.0002	-0.0009
Prev_Positive	0.1427	0.0096	-0.1117	0.0823	0.0011	1.0	-0.0318	-0.7153	0.5469
Prev_Negative	0.0114	0.0748	-0.0585	-0.0348	-0.0008	-0.0318	1.0	-0.6657	-0.5590
Prev_Neutral	-0.1130	-0.0599	0.1250	-0.0366	-0.0002	-0.7153	-0.6657	1.0	-0.0178
Prev_Compound	0.0843	-0.0369	-0.0367	0.0869	-0.0009	0.5469	-0.5590	-0.0178	1.0

Table 2: Correlation matrix for companies' open-source repositories

R.Q. 4: Is there a relation between the sentiment of a commit comment and disengagement in company and non-profit repositories?

Firstly, we will look at the impact of sentiment and toxicity on the engagement of a particular developer. In this scenario, we define disengagement as the *increase in the time between two consecutive commits by a particular developer.*

To calculate this impact, we first correlated our sentiment and toxicity measures with the difference in the time it takes for every developer's next commit.

	NLTK				Additional Metrics						
	Positive	**Negative**	**Neutral**	**Compound**	**Toxic**	**Obscene**	**Threat**	**Insult**	**Identity_hate**	**Severe_toxic**	**Time Diff.**
Positive	1.0	-0.0318	-0.7155	0.5469	0.0005	0.0023	0.0006	0.0030	0.0016	0.0020	0.1703
Negative	-0.0318	1.0	-0.6659	-0.5590	0.0550	0.0180	0.0212	0.0269	0.0015	0.0012	0.0427
Neutral	-0.7155	-0.6659	1.0	-0.0179	-0.0384	-0.0141	-0.0151	-0.0208	-0.0022	-0.0020	-0.1538
Compound	0.5469	-0.5590	-0.0179	1.0	-0.0273	-0.0098	-0.0122	-0.0120	0.0009	0.0034	0.0884
Toxic	0.0005	0.0550	-0.0384	-0.0273	1.0	0.6818	0.1654	0.6519	0.2088	0.5023	0.0082
Obscene	0.0023	0.0180	-0.0141	-0.0098	0.6818	1.0	0.0246	0.4361	0.1243	0.5650	0.0070
Threat	0.0006	0.0212	-0.0151	-0.0122	0.1654	0.0246	1.0	0.0393	0.0196	0.0982	0.0031
Insult	0.0030	0.0269	-0.0208	-0.0120	0.6519	0.4361	0.0393	1.0	0.3413	0.6523	0.0046
Identity_hate	0.0016	0.0015	-0.0022	0.0009	0.2088	0.1243	0.0196	0.3413	1.0	0.4546	0.0049
Severe_toxic	0.0020	0.0012	-0.0020	0.0034	0.5023	0.5650	0.0982	0.6523	0.4546	1.0	0.0039
Time Diff.	**0.1703**	0.0427	**-0.1538**	0.0884	0.0082	0.0070	0.0031	0.0046	0.0049	0.0039	1.0

Table 3: Correlation matrix for sentiment and time difference in companies' open-source software

	NLTK				Additional Metrics						
	Positive	**Negative**	**Neutral**	**Compound**	**Toxic**	**Obscene**	**Threat**	**Insult**	**Identity_hate**	**Severe_toxic**	**Time Diff.**
Positive	1.0	-0.0797	-0.6563	0.5695	0.0055	-0.0005	0.0069	0.0039	0.0080	0.0084	-0.0248
Negative	-0.0797	1.0	-0.6998	-0.6198	0.0297	0.0133	0.0174	0.0141	0.0233	0.0195	-0.0008
Neutral	-0.6563	-0.6998	1.0	0.0610	-0.0264	-0.0097	-0.0182	-0.0135	-0.0233	-0.0208	0.0184
Compound	0.5695	-0.6198	0.0610	1.0	-0.0149	-0.0075	-0.0072	-0.0066	-0.0106	-0.0017	-0.0376
Toxic	0.0055	0.0297	-0.0264	-0.0149	1.0	0.7085	0.3584	0.6621	0.5765	0.5987	0.0261
Obscene	-0.0005	0.0133	-0.0097	-0.0075	0.7085	1.0	0.0147	0.5176	0.2521	0.5129	0.0021
Threat	0.0069	0.0174	-0.0182	-0.0072	0.3584	0.0147	1.0	0.0700	0.5586	0.5906	0.0094
Insult	0.0039	0.0141	-0.0135	-0.0066	0.6621	0.5176	0.0700	1.0	0.3200	0.6264	0.0090
Identity_hate	0.0080	0.0233	-0.0233	-0.0106	0.5765	0.2521	0.5586	0.3200	1.0	0.4458	0.0252
Severe_toxic	0.0084	0.0195	-0.0208	-0.0017	0.5987	0.5129	0.5906	0.6264	0.4458	1.0	-0.0141
Time Diff.	-0.0248	-0.0008	0.0184	-0.0376	0.0261	0.0021	0.0094	0.0090	0.0252	-0.0141	1.0

Table 4: Correlation matrix for sentiment and time difference in non-profit open-source software

It seems like for a company's repository, positive sentiment is positively correlated with time difference and neutral sentiment is negatively correlated with time difference. This implies that more positivity in a comment or less neutrality in a comment can cause a higher period of disengagement. There does not seem to be any correlation between negative sentiment or toxicity with disengagement. For a non-profit repository, there is no correlation between sentiment and disengagement. It implies that developers tend to have other reasons for disengaging from these repositories. However, it is important to note that toxicity in a non-profit open-source setting has a greater impact on disengagement than a company's repository.

To further validate the impact of sentiments on the time difference between two commits from the same developer, we aggregated the commits by the developer and tried to fit the time difference into a regression model. For all regression analyses, VIF was calculated to make sure we avoid collinearity.

To investigate the influence of sentiment and toxicity on current commit messages, we aggregated commits by the developer and conducted regression analyses. We attempted to predict the commit message's time difference concerning the next commit based on the previous commit message's sentiment and toxicity.

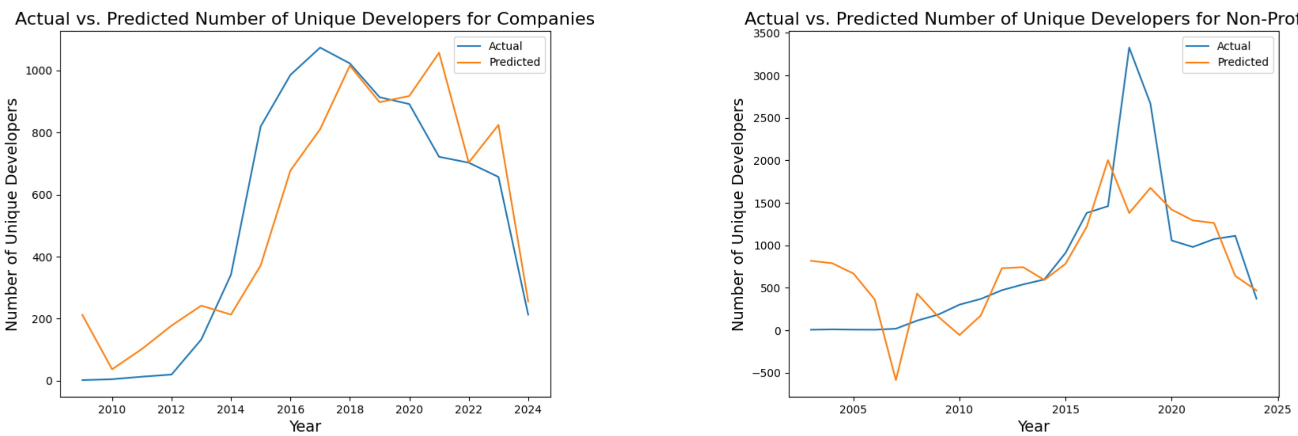
The coefficients for company repositories reveal that an increase in sentiment is associated with a decrease in time difference, while a similar increase in toxicity results in a decrease in time difference. The intercept, representing the baseline time difference when all variables are zero, is around 1942 days. However, the low R^2 value of 0.00034 indicates that the model explains only a minimal portion of the variance in the time difference. Consequently, the commit message's sentiment and toxicity may have limited predictive power for time difference, suggesting that other factors or a more comprehensive model may be necessary for a more accurate understanding of time difference dynamics for a particular developer in a company repository.

On the other hand, the coefficients for non-profit repositories reveal that an increase in sentiment is associated with a decrease in time difference, while a similar increase in toxicity results in a decrease in time difference. The intercept, representing the baseline time difference when all variables are zero, is 2048 days. However, the low R^2 value of 0.0038 indicates that the model explains only a minimal portion of the variance in the time difference. Consequently, the commit message's sentiment and toxicity may have limited predictive power for time difference, suggesting that other factors or a more comprehensive model may be necessary for a more accurate understanding of time difference dynamics for a particular developer in a non-profit repository.

In comparing the regression analyses of company and non-profit repositories, it becomes evident that the influence of sentiment and toxicity on time difference varies between the two contexts. While company repositories show a modest impact of sentiment and toxicity, non-profit repositories exhibit more pronounced effects. However, across both scenarios, the limited predictive power of these factors alone suggests the necessity of considering additional variables or adopting a more comprehensive model to achieve a nuanced understanding of time difference dynamics in software development.

Therefore, our results indicate that the sentiment of a commit message is related to the time difference between two commits from the same developer for company repositories, but the impact of sentiment on the time difference in both repositories is limited. This suggests that other factors may also play a role in shaping the time difference between two commits from the same developer.

On the macro level, a more engaged community is one where *there are more unique developers contributing to the codebase in a particular year.*



(a) Company repositories

(b) Non-profit repositories

Figure 3: Regression model for the number of unique developers for a year based on total SLOC, mean toxicity, and mean sentiment

We hypothesize that a more positive, less toxic, and larger codebase would have more engagement. To test this, we aggregated the commits by year and attempted to predict the number of unique developers for that year based on the mean sentiment, mean toxicity, and the total lines of code worked on for that year. The coefficients for company repositories reveal that an increase in SLOC is associated with an increase in the number of unique developers. An increase in toxicity results in a decrease in the number of unique developers. The increase in sentiment (i.e., moving more towards positivity), is associated with an increase in the number of unique developers. The intercept, representing the baseline number of unique developers when all variables are zero, is around 238. The high R^2 value of 0.76 indicates that the model explains a significant portion of the variance in the number of unique developers. Consequently, the SLOC, sentiment, and toxicity have a strong predictive power for the number of unique developers, suggesting that these factors play a significant role in shaping the engagement of a community in a company repository.

On the other hand, the coefficients for non-profit repositories reveal that an increase in SLOC is associated with an increase in the number of unique developers. An increase in toxicity results in a decrease in the number of unique developers. The increase in sentiment (i.e., moving more towards positivity), is associated with an increase in the number of unique developers. The intercept, representing the baseline number of unique developers when all variables are zero, is around 1170. The moderate R^2 value of 0.48 indicates that the model explains a moderate portion of the variance in the number of unique developers. Consequently, the SLOC, sentiment, and toxicity have a moderate predictive power for the number of unique developers, suggesting that these factors play a moderate role in shaping the engagement of a community in a non-profit repository.

In comparing the regression analyses of company and non-profit repositories, as seen in Figure 3, it becomes evident that the influence of SLOC, sentiment, and toxicity on the number of unique developers varies between the two contexts. While company repositories show a strong impact on these factors, non-profit repositories exhibit a moderate effect. However, across both scenarios, the predictive power of these factors suggests that these factors play a significant role in shaping the engagement of a community in software development.

Therefore, our results indicate that the sentiment of a commit message is related to the engagement of a community and that the sentiment of a commit message, along with the toxicity and the size of the codebase, can be used to predict the engagement of a community in software development. However, the impact of sentiment and toxicity on an individual developer's engagement is limited.

R.Q. 5: Does the effect of the sentiment vary across different developer groups?

We employed a data-driven approach, utilizing commit data from the GitHub repository, to investigate the impact of interaction sentiments on different developer groups within software development projects. We categorized these developers into groups of two: "Professionals" and "Beginners", based on a threshold of 5 commits. Professionals were those with more than 5 commits, while Beginners had 5 or fewer commits.

In both types of repositories, the distribution between professionals and beginners was similar, with around 82-85% beginners and the remaining developers being professionals.

To answer our R.Q., we correlated the same variables as before with the separated developers for both non-profit and company repositories.

In non-profit repositories, novice developers with up to 5 commits exhibit a notable correlation of 0.15 between the positive sentiment of their commit messages and the sentiment of the preceding commit messages. However, their commit messages' negative sentiment scores show a weaker correlation of 0.07 with the preceding negative scores. Similarly, their neutral sentiment displays a modest positive correlation of 0.12 with the previous neutral sentiment scores, suggesting a tendency toward positivity in their contribution messages. Toxicity is not correlated with the previous commit message's toxicity.

In non-profit repositories, professional developers with more than 5 commits exhibit a notable correlation of 0.16 between the positive sentiment of their commit messages and the sentiment of the preceding commit messages. Moreover, their commit messages' negative sentiment scores show a correlation of 0.12 with the preceding negative scores. Similarly, their neutral sentiment displays a positive correlation of 0.15 with the previous neutral sentiment scores, suggesting a tendency toward positivity in their contribution messages. Toxicity is not correlated with the previous commit message's toxicity.

These findings collectively suggest a prevailing inclination towards constructive and positive communication among developers in non-profit settings, regardless of their experience level. This positivity may foster a collaborative and supportive environment conducive to effective project development and community engagement. However, it is crucial to acknowledge the need for monitoring negative sentiment trends among professionals to mitigate any potential adverse impacts on the collaborative atmosphere.

In company repositories, novice developers with up to 5 commits exhibit weak correlations with all types of sentiment and even toxicity. For positive sentiment, the correlation score is 0.005. For negative sentiment, the correlation score is 0.01. For neutral sentiment, the correlation score is -0.02. This implies that novice developers may not be committing code with more positivity in their commit message. Toxicity is not correlated with the previous commit message's toxicity, which means that they do follow the code of conduct in that respect.

In company repositories, professional developers with more than 5 commits exhibit a notable correlation of 0.15 between the positive sentiment of their commit messages and the sentiment of the preceding commit messages. However, their commit messages' negative sentiment scores show a weaker correlation of 0.07 with the preceding negative scores, implying negativity does not flow down. Similarly, their neutral sentiment displays a modest positive correlation of 0.12 with the previous neutral sentiment scores, suggesting a tendency toward positivity in their contribution messages. Toxicity is not correlated with the previous commit message's toxicity.

Overall, while positive communication trends are evident among experienced developers within company repositories, the disparate patterns observed between novice and professional developers in companies highlight the importance of tailored support mechanisms to nurture positive communication practices across all experience levels within corporate development teams. By recognizing and addressing these nuances, companies can cultivate an environment conducive to innovation, collaboration, and sustained project success.

While there is a correlation between communication trends, regression analysis reveals that we cannot predict the sentiment of a commit message using the previous commit message's sentiment or toxicity. The R^2 values are less than 0.01 for both types of developers in both types of repositories. This implies that the previous commit message's sentiment and toxicity do not cause the sentiment of a commit message.

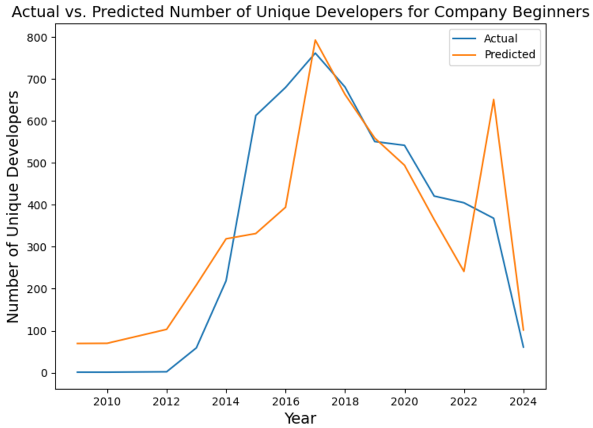
Regarding individual developer disengagement, correlating the time difference with all the variables reveals that for both categories of developers, in both types of repositories, toxicity does not have a strong correlation with the time difference between two successive commits by the same developer. Also, the sentiment of the commit message overall does not have an impact on individual developer engagement. However, positive sentiment in the commit message for a company professional has a positive correlation of 0.17 with the time difference, and negative sentiment in the commit message has a correlation of -0.16 with the time difference, implying that more positivity causes more individual developer disengagement. However, the regression analysis did not show that we could predict this period of disengagement using these variables, implying that while they are correlated, positive sentiment is not one of the drivers of a longer disengagement period.

On a community-wide engagement metric, we used regression analysis to predict the unique number of beginners and professional developers in both categories of repositories. We used total SLOC, mean toxicity, and mean sentiment for that year as predictor variables. The regression model plots can be seen in Figure 4. For company repositories, the R^2 value for beginners 0.70 is, and for professionals is 0.88. For non-profit repositories, the R^2 value for beginners is 0.42 and for professionals is 0.55.

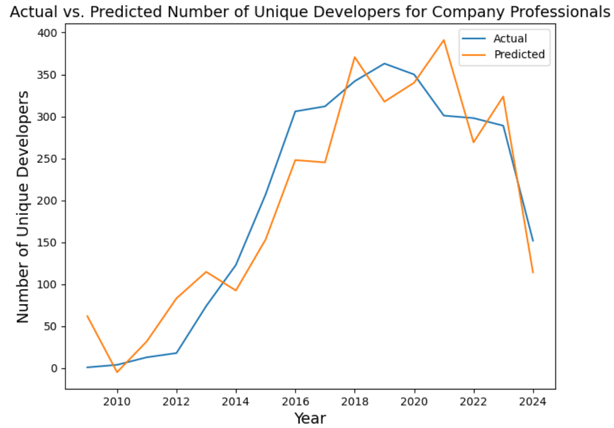
The R^2 value observed for professional developers across company repositories, standing at emphasizes the significant impact of variables like SLOC, toxicity, and sentiment on their engagement levels. This suggests that these factors play a pivotal role in shaping the interactions and contributions of seasoned developers within the professional context. Conversely, while the R^2 values for beginners in company repositories are slightly lower at 0.70, they still exhibit a considerable level of predictability, indicating that these individuals are also influenced by similar factors, albeit to a lesser extent.

However, within non-profit repositories, the predictability diminishes for beginners and professionals, as reflected in the comparatively lower R^2 values. This implies that while these factors remain influential, the engagement dynamics for both beginner and professional developers in non-profit settings are subject to additional complexities or perhaps different motivational drivers, and require further research. But, we can still conclude that SLOC, toxicity, and sentiment are more impactful on professional developers than beginner developers.

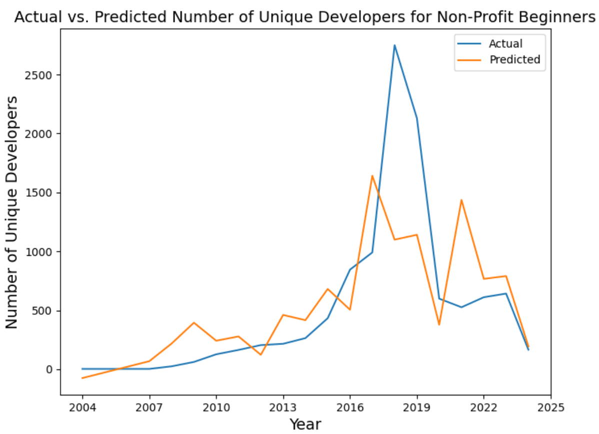
Overall, the results suggest that the engagement of developers in company repositories is more predictable using our variables than in non-profit repositories and that the engagement of professional developers is more predictable than that of beginners. A larger codebase, coupled with less toxicity and more positivity, helps professional developers for a community to continue returning to commit to the community, and for beginners in company repositories to start committing to a repository. Therefore, it is integral for community managers to focus on our predictors to ensure the continued engagement of professional developers in company repositories, particularly in company repositories where predictability is higher, and thus, maintain the health of the repository.



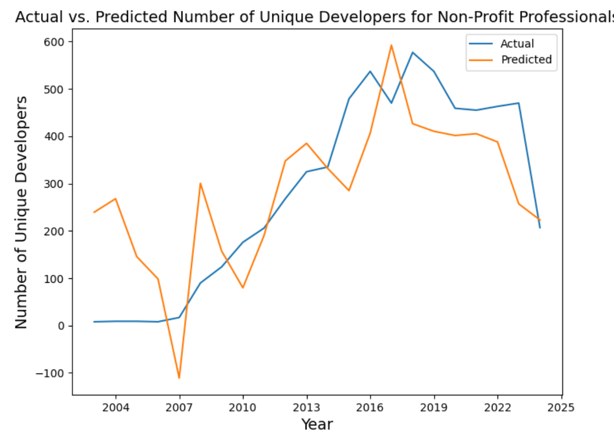
(a) Beginners in Company Repositories



(b) Professionals in Company Repositories



(c) Beginners in Non-Profit Repositories



(d) Professionals in Non-Profit Repositories

Figure 4: Regression model for the number of unique developers of both categories for a year based on total SLOC, mean toxicity, and mean sentiment

We also performed T-tests on the toxicity scores of these developer groups. The test resulted in a T-statistic of -0.2595 and a p-value of 0.7952, indicating the lack of any significant difference between the two.

This structured methodology provided a comprehensive framework for understanding the influence of interaction sentiments on developer behavior within software development projects.

5 Discussion and Conclusion

Our study aimed to investigate the impact of interaction sentiments on developer engagement and contribution to software development projects.

Our findings revealed that the sentiment of a commit message is related to the engagement of a community and that the sentiment of a commit message, along with the toxicity and the size of the codebase, can be used to predict the engagement of a community in a company's open-source software. However, the impact of sentiment and toxicity on an individual developer's engagement is limited. We also found that the sentiment of a commit message is related to the sentiment of the previous commit message, but the impact is limited. This suggests that other factors may also play a role in shaping the sentiment of commit messages. Furthermore, we observed that the toxicity in the comments is very low, and the difference between the toxicity in company and non-profit repositories is insignificant. This indicates that the code of conduct and guidelines for contributing to open-source repositories are effective in curbing toxic commit comments from becoming part of their commit history.

Our study has several strengths. We used a large dataset of commit messages from repositories, which allowed us to draw generalizable conclusions about the impact of interaction sentiments on developer engagement and contribution. We also used multiple sentiment analysis techniques to analyze the commit messages to gauge the impact of interaction sentiments on developer behavior within software development projects. Additionally, we used regression analyses to investigate the influence of sentiment and toxicity on the time difference between two commits from the same developer to gain insights into the dynamics of engagement within software development projects. Finally, we used a case study to illustrate the impact of interaction sentiments on different developer groups within software development projects to gain insights into how interaction sentiments influenced developer groups within the project context.

Despite our contributions, we acknowledge the limitations inherent in our study. Our reliance on commit messages as the primary data source may not capture the full spectrum of interactions within open-source projects, such as discussions in issues and pull requests. We tried tackling this with case studies on toxicity in the git repository for a toxic repository like Bitcoin by looking at pull request, issues and commit comments. The results were in line with our other findings (Appendix). We could not replicate this in other repositories due to the rate-limiting issues from the Git APIs. Additionally, we only used NLTK sentiment analysis [3], the OpenAI Moderation API [21], and a hugging face classifier [24] to analyze the sentiment of commit messages, which may not capture the full range of interaction sentiments within software development projects. Furthermore, our case study may not capture the full range of developer groups. Finally, we only investigated the impact of interaction sentiments on developer engagement and contribution, which may not capture the full impact of interaction sentiments on other aspects of software development projects.

In the future, we plan to address these limitations by analyzing commit messages from different software development platforms and using more sentiment analysis techniques to capture the full range of interaction sentiments within software development projects. We also plan to extend our dataset to include issues and PRs since they also capture interactions between developers in a community. We also plan to conduct case studies to gain more insights into the impact of interaction sentiments on different developer groups within software development projects.

Ultimately, our study contributes to understanding the dynamics of developer interactions and emphasizes the broader implications for fostering healthier and more productive software development communities. By recognizing the impact of interaction sentiments, project maintainers and community leaders can implement strategies to cultivate positive interactions and enhance overall project engagement and collaboration.

6 Team Membership and Attestations

Team Members Thrisha Kopula, Kunal Pai, Saisha Shetty, Ariel Kamen, and Vijeth KL participated sufficiently.

- Thrisha Kopula: Extraction of the commit and pull request dataset. Helped define the disengagement metrics and worked on processing the mined data. Worked on the report.
- Kunal Pai: Extraction of commit data pipeline, correlation matrices for all variables, regression analyses, and writing report.
- Saisha Shetty: Extraction of the commit and issues dataset, worked on defining the disengagement metrics, and helped write the report.
- Ariel Kamen: Writing the OpenAI Moderation API code and running the CSV files through it. Worked on the report.
- Vijeth K L: Defined and measured disengagement in repositories and compared it to the toxic and positive interactions across years. Worked on the case study and research question 5. Worked on the report.

References

[1] Apple. Swift. https://github.com/apple/swift.

[2] Binkert, N., Beckmann, B., Black, G., Reinhardt, S. K., Saidi, A., Basu, A., Hestness, J., Hower, D. R., Krishna, T., Sardashti, S., et al. The gem5 simulator. *ACM SIGARCH computer architecture news 39*, 2 (2011), 1–7.

[3] Bird, S., Klein, E., and Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[4] Calefato, F., Gerosa, M. A., Iaffaldano, G., Lanubile, F., and Steinmacher, I. Will you come back to contribute? investigating the inactivity of oss core developers in github. *Empirical Software Engineering 27*, 3 (2022), 76.

[5] Chartrand, T. L., and Bargh, J. A. The chameleon effect: The perception–behavior link and social interaction. *Journal of personality and social psychology 76*, 6 (1999), 893.

[6] Coelho, J., and Valente, M. T. Why modern open source projects fail. In *Proceedings of the 2017 11th Joint meeting on foundations of software engineering* (2017), pp. 186–196.

[7] De Choudhury, M., and Counts, S. Understanding affect in the workplace via social media. In *Proceedings of the 2013 conference on Computer supported cooperative work* (2013), pp. 303–316.

[8] Duggan, M. Online harassment 2017.

[9] Facebook. React. https://github.com/facebook/react.

[10] freeCodeCamp. freecodecamp. https://github.com/freeCodeCamp/freeCodeCamp.

[11] Google. Guava. https://github.com/google/guava.

[12] Gousios, G., and Spinellis, D. Ghtorrent: Github's data from a firehose. In *2012 9th IEEE Working Conference on Mining Software Repositories (MSR)* (2012), IEEE, pp. 12–21.

[13] Guzman, E., Azócar, D., and Li, Y. Sentiment analysis of commit comments in github: an empirical study. In *Proceedings of the 11th working conference on mining software repositories* (2014), pp. 352–355.

[14] Islam, M. R., and Zibran, M. F. Sentistrength-se: Exploiting domain specificity for improved sentiment analysis in software engineering text. *Journal of Systems and Software 145* (2018), 125–146.

[15] Li, R., Pandurangan, P., Frluckaj, H., and Dabbish, L. Code of conduct conversations in open source software projects on github. *Proceedings of the ACM on Human-computer Interaction 5*, CSCW1 (2021), 1–31.

[16] Lowe-Power, J., Ahmad, A. M., Akram, A., Alian, M., Amslinger, R., Andreozzi, M., Armejach, A., Asmussen, N., Beckmann, B., Bharadwaj, S., et al. The gem5 simulator: Version 20.0+. *arXiv preprint arXiv:2007.03152* (2020).

[17] Microsoft. Vscode. https://github.com/microsoft/vscode, 2013.

[18] Miller, C., Cohen, S., Klug, D., Vasilescu, B., and KaUstner, C. " did you miss my comment or what?" understanding toxicity in open source discussions. In *Proceedings of the 44th International Conference on Software Engineering* (2022), pp. 710–722.

[19] Miller, C., Widder, D. G., Kästner, C., and Vasilescu, B. Why do people give up flossing? a study of contributor disengagement in open source. In *Open Source Systems: 15th IFIP WG 2.13 International Conference, OSS 2019, Montreal, QC, Canada, May 26–27, 2019, Proceedings 15* (2019), Springer, pp. 116–129.

[20] Node.js. Node. https://github.com/nodejs/node.

[21] OpenAI. Openai moderation api. https://platform.openai.com/docs/guides/moderation, 2024.

[22] OpenJDK. Jdk. https://github.com/openjdk/jdk.

[23] Raman, N., Cao, M., Tsvetkov, Y., Kästner, C., and Vasilescu, B. Stress and burnout in open source: Toward finding, understanding, and mitigating unhealthy interactions. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results* (2020), pp. 57–60.

[24] Rimi98. Rimi98/negativecommentclassifier. https://huggingface.co/spaces/Rimi98/NegativeCommentClassifier/tree/main, 2024.

[25] Sharma, P. N., Hulland, J., and Daniel, S. Examining turnover in open source software projects using logistic hierarchical linear modeling approach. In *Open Source Systems: Long-Term Sustainability: 8th IFIP WG 2.13 International Conference, OSS 2012, Hammamet, Tunisia, September 10-13, 2012. Proceedings 8* (2012), Springer, pp. 331–337.

[26] Spadini, D., Aniche, M., and Bacchelli, A. Pydriller: Python framework for mining software repositories. In *Proceedings of the 2018 26th ACM Joint meeting on european software engineering conference and symposium on the foundations of software engineering* (2018), pp. 908–911.

7 Appendix

Case study: Impact of toxicity on the engagement of people involved in toxic interactions

For our case study, we chose to analyze Bitcoin's git repository since it had the most toxic comments among the ones we analyzed.

We chose the year with the most toxic comments(2020) and plotted a graph of the number of commits and pull requests for the users involved in toxic interactions.

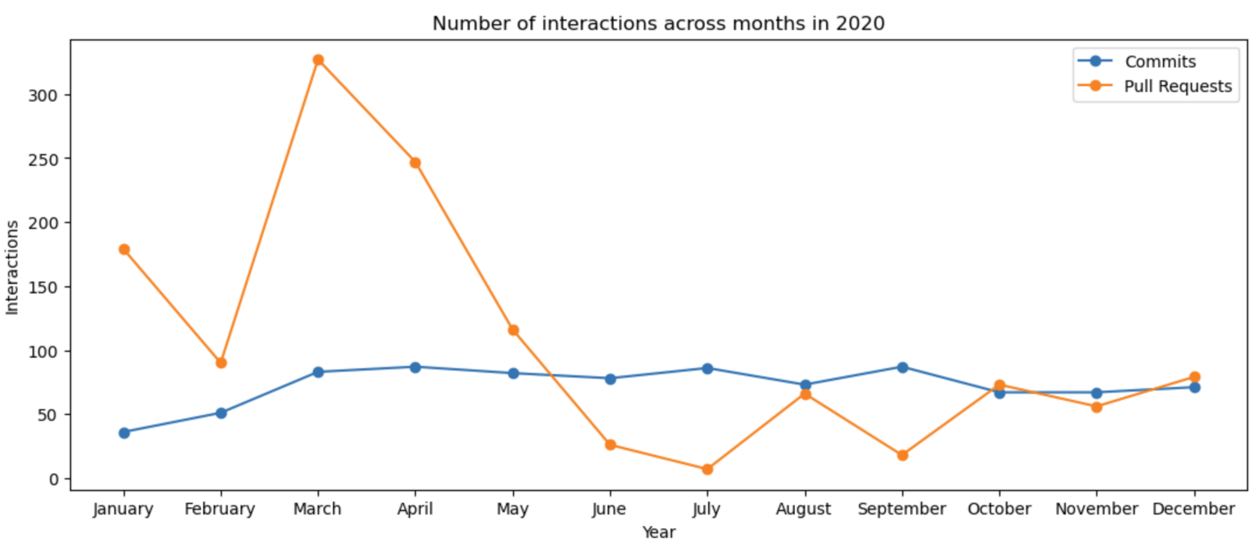
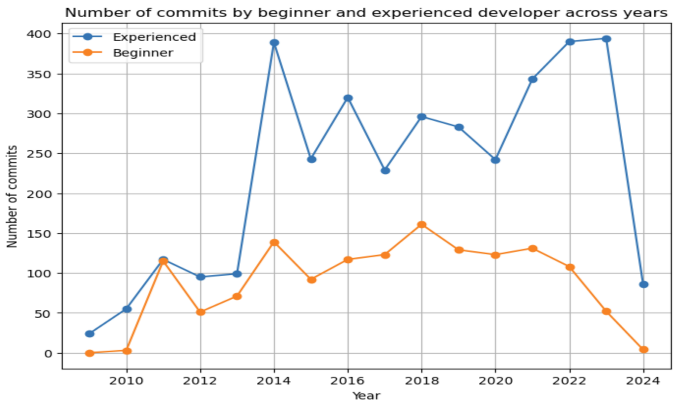


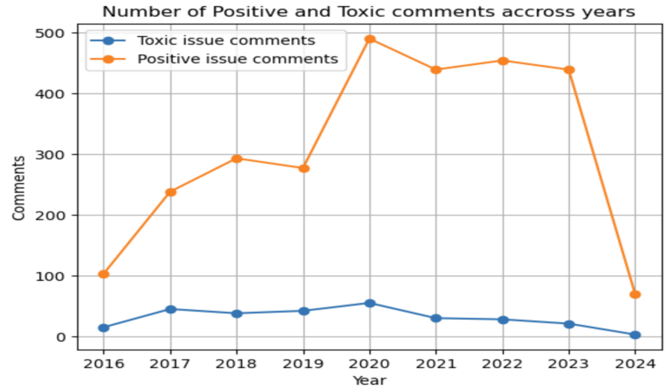
Figure 5: Company repositories

We then did a correlation analysis for the PRs and commits against the number of toxic comments each month. This test returned low correlation scores of -0.0973 for commits and 0.0453 for PRs.

This result helped us conclude that toxic comments don't impact engagement even among the people involved in these interactions.



(a) Beginners vs experienced developers



(b) Number of toxic and positive comments across years

Figure 6: Impact of Toxicity in Experienced vs Beginner Developers