systems

MDPI

Article

Fostering Productive Open Source Systems: Understanding the Impact of Collaborator Sentiment

Joonhaeng Lee [1] and Keuntae Cho [2,*]

[1] Graduate School of Management of Technology, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon 16419, Republic of Korea; joonhaeng@skku.edu
[2] Department of Systems Management Engineering, Graduate School of Management of Technology, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon 16419, Republic of Korea
* Correspondence: ktcho@skku.edu; Tel.: +82-31-290-7602

Abstract: Open Source Software (OSS) development is a complex socio-technical system in which collaborator attitudes influence the outcomes. This study empirically analyzes the impact of participant sentiment (positive, neutral, negative) on productivity, defined by Pull Requests (PR), Lines of Code (LoC), and interactions (as indicated by comment volume). Data on PRs, LoC, and comments, were collected from 20 top GitHub repositories. SentiStrength-SE was used to classify participant sentiment based on average comment sentiment. Appropriate nonparametric statistical and correlation analyses were performed. The results showed that contributors with positive sentiments have the highest productivity and interaction. Negative-sentiment contributors also significantly outperform the neutral group in both areas. The neutral group consistently ranks the lowest. The general patterns are as follows: positive > negative > neutral. The strongest positive correlations between productivity and interaction are observed in the positive-sentiment group. These findings empirically demonstrate that the sentiment levels of collaborators are significantly associated with OSS productivity and engagement, offering insights into socio-technical dynamics. Fostering a positive environment is a key strategy for enhancing OSS performance and sustainability.

Keywords: open source software; sentiment analysis; productivity; socio-technical systems

check for updates

Academic Editor: Emmanuel D. Adamides

Received: 11 April 2025
Revised: 4 June 2025
Accepted: 5 June 2025
Published: 6 June 2025

Citation: Lee, J.; Cho, K. Fostering Productive Open Source Systems: Understanding the Impact of Collaborator Sentiment. *Systems* **2025**, *13*, 445. https://doi.org/10.3390/systems13060445

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Open Source Software (OSS) development has become a cornerstone of modern technological advancement, representing a paradigm in which code is publicly accessible for anyone to view, modify, and distribute [1]. More than mere code repositories, OSS projects function as complex socio-technical systems, or STS. Developers of these systems from diverse backgrounds and affiliations (social elements) collaboratively create software using digital platforms such as GitHub along with various development tools (technical elements) [2,3]. The success and sustainability of community-driven or corporate-supported system initiatives [4] has increasingly draw attention, particularly in the context of open innovation and global sustainability goals such as those outlined in the United Nations' Roadmap for Digital Cooperation [5] and assessed by entities such as the World Benchmarking Alliance [6,7].

Within these intricate socio-technical environments, the attitudes and emotional states (sentiments) of collaborators can significantly shape interactions, and consequently also project outcomes. This assertion is grounded in established psychological research; emotions are not simply transient feelings but powerful regulators of essential cognitive func-

tions such as attention, memory, decision-making, and problem solving, all of which are critical for the demanding work of software development. For instance, Broaden-and-build theory suggests that positive emotions can broaden an individual's thought–action repertoires, fostering creativity and adaptability [8], whereas negative emotions may narrow focus, potentially hindering innovative solutions.

Although the importance of collaboration quality is acknowledged, the direct impact of participant sentiment on measurable productivity and interaction levels within OSS projects requires deeper empirical investigation. The need for more direct empirical scrutiny is evident when surveying existing research landscapes. The detrimental effects of negative interactions and the general impact of developer attitudes have been long-standing concerns within OSS communities [9,10]. However, rigorous quantitative investigations directly correlating a spectrum of participant emotional states (positive, neutral, and negative) with granular productivity metrics such as Lines of Code (LoC) and Pull Request (PR) volumes are less prevalent [11,12].

Previous research has often approached the role of sentiments indirectly. For instance, studies have analyzed sentiments concerning bug fix times [13,14] or explored emotional expressions during specific community events such as the temporary absence of key maintainers [15]. While these studies offer valuable insights, they typically do not provide a quantification of how ongoing collaborator sentiment correlates with core productivity outputs such as the volume of code contributions or the frequency of successful code integrations across a broad spectrum of OSS activities.

The present study addresses this gap by directly examining the relationship between collaborator sentiment and tangible contributions. To accomplish this, we analyze data from multiple GitHub repositories, focusing on direct indicators of productivity such as PRs and LoC alongside interaction levels as measured by comment exchanges. We employ sentiment analysis techniques to classify the comments of participants into positive, neutral, and negative sentiment categories. This approach allows for more direct assessment of how participants' emotional states relate to their engagement and output in an OSS ecosystem. Understanding this relationship is crucial because OSS collaboration dynamics differ from typical organizational settings owing to diverse participant affiliations, flexible engagement, and horizontal structures, all of which can influence how sentiment translates into productivity.

Moreover, the contemporary landscape of software development, including that of OSS, is characterized by rapid change and inherent uncertainties. In such contexts, research on positive organizational behavior suggests that the emotional resilience of collaborators becomes critical for sustained engagement and contribution, including their ability to bounce back from setbacks and adapt to evolving challenges [16].

Therefore, the primary goal of this study is to empirically investigate the impact of collaborator sentiment on productivity and interaction levels within OSS development environments. By clarifying this relationship using direct quantitative indicators, we aim to provide evidence-based insights that can foster more effective and sustainable OSS communities.

To achieve the above goal, this study seeks to answer the following research questions:

- **RQ1:** Do significant differences exist in system outcomes, specifically productivity indicators (Number of PRs, Total LoC, LoC/PR), based on the internal system variable of OSS participant sentiment state (positive/neutral/negative)?
- **RQ2:** Which sentiment group has a more positive impact on the systemic property of community activation through active communication?

- **RQ3:** What correlation patterns emerge between participant attitudes and productivity indicators, and what patterns suggest the potential existence of feedback loops [17,18] within the system?

This paper makes several contributions to the field. First, it provides robust empirical evidence directly linking collaborator sentiment to key productivity metrics (PRs, LoC) and interaction frequency in OSS projects. Second, it uncovers the nuanced finding that participants exhibiting negative sentiment can demonstrate higher productivity than those with neutral sentiment, challenging overly simplistic interpretations of the role. Third, it provides a methodological demonstration of the utility of combining large-scale data mining from platforms such as GitHub with sentiment analysis and nonparametric statistical techniques in order to investigate socio-technical dynamics in OSS. Finally, our findings offer actionable insights for OSS community managers and organizations seeking to cultivate more productive, engaged, and sustainable collaborative environments.

The remainder of this paper is organized as follows. Section 2 discusses the theoretical background, including the systemic understanding of OSS development, then reviews the related literature on collaboration processes, sentiments, and productivity in Software Engineering (SE). Section 3 details the research methodology, including repository selection, data collection, procedures, sentiment analysis techniques, data processing, and statistical analysis methods. Section 4 presents the empirical findings addressing our research questions. Section 5 interprets the results, discusses their implications, and acknowledges the limitations of the study. Finally, Section 6 summarizes the study and proposes directions for future research.

2. Theoretical Background and Literature Review

2.1. Theoretical Background

2.1.1. Systemic Understanding of the OSS Development Environment

Applying an STS perspective is useful for understanding the OSS development environment. STS theory emphasizes that technical subsystems (code, tools, platforms, etc.) and social subsystems (developers, community norms, collaboration culture, etc.) are inseparable, and that optimal performance requires harmonious interaction between these two systems [19]. OSS development is a classic example of STS [20] in which technical elements (Git, GitHub features, etc.) and social elements (communication styles, emotions, etc.) are intricately intertwined, thereby influencing project success. Participant "attitude" as addressed in this study is a crucial aspect of the social subsystem, and "productivity" can be viewed as a system performance indicator resulting from techno-social interactions.

2.1.2. The Collaboration Process in OSS

Collaboration in OSS typically follows a fork-and-pull model, as illustrated in Figure 1. The contributor first duplicates (forks) the program source code from the main repository for their own repository. After making additions, deletions, or modifications to the code, the contributor makes a request (pull) that these changes be incorporated back into the main repository. This request to merge the changes into the main repository is called a PR, and serves as the unit of individual contribution [21].

After a PR has been submitted to the main repository, reviewers examine the modified source code for potential issues and ensure that it adheres to the established rules of the existing codebase. If the code passes review, they ask the maintainer to merge it with the main repository. However, even if the code passes review (is closed), the maintainer may choose not to merge it based on their judgment. They also might request further modifications by the contributors. If the maintainer finally approves the merge, the contributor's PR is integrated into the main repository (merged) [22,23]. This entire sequence represents

a system process combining code writing (technical activity) and peer review/communication (social activity). Communications between contributors and reviewers, contributors and maintainers, and reviewers and maintainers mostly occur online. Specifically, these messages are recorded in the conversation history of a GitHub PR, as shown in Figure 2. If the repository is set to public, anyone using the internet can view these records.

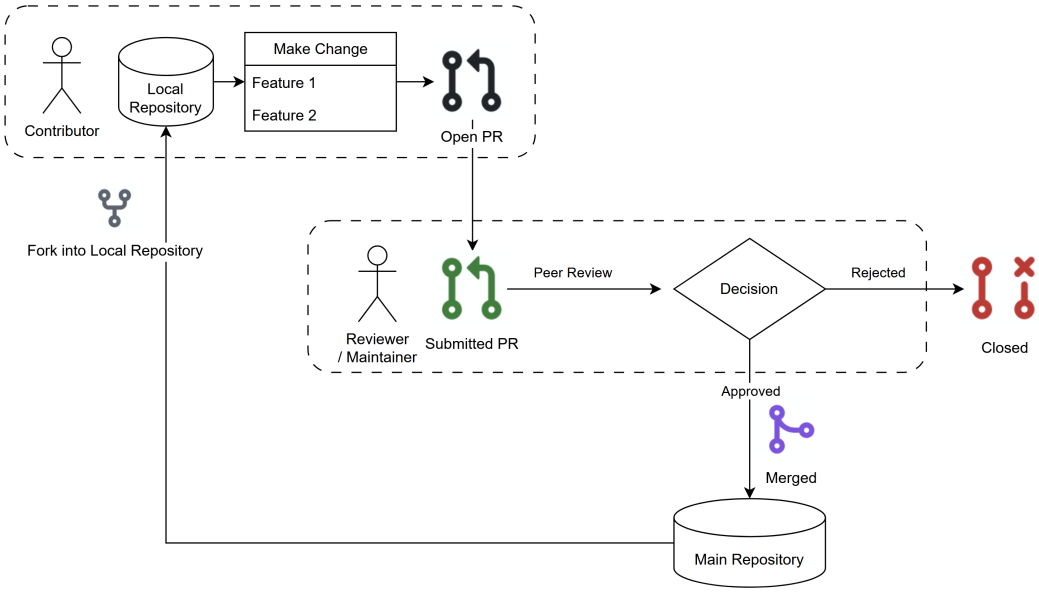

Figure 1. The flow until a PR is merged [24].

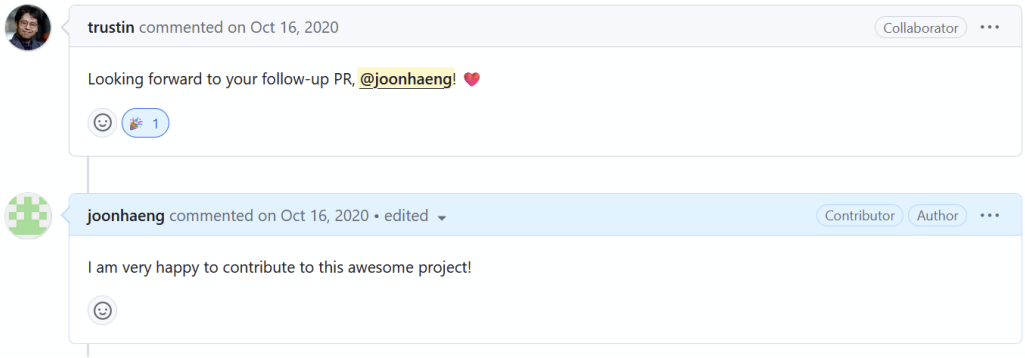

Figure 2. Example of conversation history in a PR [25].

2.1.3. Productivity in Software

Defining and measuring productivity in software systems is challenging given the nature of knowledge-based work, yet it is crucial for understanding system performance. Productivity is often quantified as the number of tasks processed per unit of time or amount of code added, deleted, or changed [26]. Zhou and Mockus [27] used monthly task throughput to measure the productivity of new developers in outsourced software development, noting potential individual differences among developers. Kieburtz et al. [28] proposed the amount of work completed per unit time as a productivity metric from a software-reuse perspective. Task throughput roughly corresponds to the number of finally merged PRs and has clear start and end dates, making it a convenient metric. However, it can overlook the importance of individual PRs, and may underestimate the productivity of developers who primarily handle large or difficult PRs.

Lines of Code (LoC) is a direct indicator of code contribution volume. Devanbu et al. [29] used the total LoC of a software project divided by the development time (analysis, design, implementation, and testing) from a software-reuse perspective. Blackburn

et al. [30] used LoC divided by man/month to compare the productivity among developers, indicating how much code one developer wrote per month. Other approaches measure productivity using function points [31], a metric for the size of information systems that can indirectly estimate productivity in development and maintenance activities [32]. However, while LoC is easily measurable by external researchers owing to the public nature of the OSS, function points often involve internal corporate information and are rarely disclosed or used in OSS, which limits their practical applications. The number of commits is another proxy for productivity [33]. Considering the ease of measuring them in the OSS environment, we used the numbers of PRs and LoC as our productivity metrics. Although LoC does not capture all facets of contributions, such as code complexity or quality, its objective and quantifiable nature across diverse projects makes it a suitable proxy for contribution volume in this large-scale study.

2.2. Related Work

2.2.1. Conversation and Attitude in PRs

Attitude is defined as a state of positive or negative feelings towards a specific object [34], and concurrently a tendency to respond positively or negatively towards it [35]. Greenwald [36] identified emotion as a core component and major driver of attitude, suggesting that strategies leveraging emotion could be effective in changing attitudes in fields such as business, politics, and education.

GitHub not only functions as a repository but also exhibits the characteristics of a social network in which anonymous developers collaborate [37]. In addition to simple information exchange, conversations within PRs serve as social interaction channels that reveal participants' emotions. Leng et al. [38] proposed that emotions influence attitudes; individuals with positive emotions tend to communicate openly and are willing to share knowledge, whereas those with negative emotions are more likely to adopt defensive and uncooperative attitudes. Furthermore, Dehbozorgi et al. [39] reported that emotions affect teamwork performance and individual achievement. Joshi's research [40] highlighted comments during the PR review process as a crucial mediator of communication and interaction among developers, demonstrating that analyzing the sentiment expressed in comments is effective in predicting PR acceptance.

2.2.2. Collaborator Attitude and Productivity

Guzman et al. [41] argued that software development generally relies on high levels of collaboration and attitudes toward collaboration influence multidimensional system outcomes such as productivity, code quality, creativity, and job satisfaction. Ferreira et al. [15]'s study analyzed community conversations during the absence of Linus Torvalds, lead developer of the Linux kernel, revealing that negative attitudes can adversely affect OSS productivity. Their findings suggest that the attitudes of highly involved developers affect overall system productivity. Our study broadens their approach by quantitatively analyzing sentiments from ongoing PR comments across multiple projects and linking them directly to code contribution metrics.

Asri et al. [42] found that PRs involving participants who expressed negative opinions during reviews took longer to close than those where participants expressed positive opinions. This result indicates that addressing the concerns of participants with negative views slows the OSS collaboration process because resolution often requires consensus among reviewers, or at least a majority. Thus, the PR system process can potentially reduce efficiency.

Previous studies indirectly related to productivity have focused on software bugs. Research has indicated that issues exhibiting more positive sentiment during bug fix-

ing tended to be resolved faster. Furthermore, codes developed in processes in which happiness-related emotions were frequently expressed tended to be higher quality with fewer bugs. The opposite was true for negative emotions [13,14]. This outcome suggests that positive participant attitudes not only affect productivity but also software quality. Our study extends this finding by examining the direct impact of sentiment on core productivity outputs such as PR volume and LoC, which provide a different perspective on developer contributions beyond bug-related activities. Reducing bugs ultimately contributes indirectly to increased productivity by saving resources otherwise spent on bug fixing. In addition, code that contains bugs is, on average, associated with more negative comments than neutral ones [43]. This implies that negative participants might be more prone to introducing bugs, thereby indirectly negatively impacting software productivity.

Regarding community and collaboration, research has suggested that positive emotions enhance willingness to collaborate and improve overall community intimacy [44]. Another study found that student groups that expressed more positive emotions achieved significantly higher academic results [39]. Furthermore, impolite responses were identified as social barriers for newcomers wishing to contribute to OSS [45].

However, conflicting findings also exist. Some research has suggested that attitudes vary across different types of tasks within a team. Although emotional attitudes were expressed during bug fixing, linking sentiment to performance proved difficult [46]. Other studies have noted that productivity remains a challenge in software development and reliability, and that the relationship with attitude has not been clearly established [47]. A study applying cluster analysis to group developers concluded that factors influencing productivity differed among clusters [48]. In another, the obtained survey results indicated that developers do not rank attitude or emotion among the top factors related to perceived productivity, contrasting with general knowledge workers, who do prioritize emotional factors [49]. A study on university students engaged in pair programming found no correlation between attitude-related factors and performance metrics such as assignment/exam scores or implementation times [50]. Conversely, another study analyzed the emotions expressed while solving programming problems and found that positive emotions such as happiness tended to be correlated with increased productivity [51]. These diverse findings underscore the need to conduct further research.

2.3. Research Problem Statement

Although previous studies have attempted to measure productivity and link it to attitude or emotion, most have employed indirect methods. Research associating productivity directly with attitude or emotion from a systemic perspective remains scarce. This paucity likely stems from the nature of software development. As it relies heavily on human intellectual activity, quantification is difficult, and metrics (for example, sentiment scores, production volume, and LoC) are often not amenable to standard statistical analysis.

By contrast, this study explores the relationship between software productivity and participant sentiment using quantified direct indicators. By uniquely employing direct metrics and appropriate statistical methods, we seek to demonstrate the connection between productivity and attitude/emotion more clearly than indirect approaches. By applying system theory to practical problem-solving, our results should provide stakeholders leading or participating in OSS projects with empirical insights into strategies for enhancing overall system productivity.

3. Research Methods

This study empirically investigated the impact of collaborator sentiment on productivity and interaction levels within OSS development environments. We used direct quantitative indicators, and this section describes the approach used to achieve that objective.

OSS participants are highly diverse in terms of geographical location [52], organizational affiliation [53], cultural background [54], and experience levels [55]. Collaboration among such diverse individuals necessitates internet-based cooperation. Therefore, publicly accessible repositories are essential. Services such as GitHub, GitLab, and Bitbucket offer such repositories, with GitHub being the largest and most active repository service [56]. GitHub provides a wide range of information through its Application Programming Interface (API), which makes it possible to collect nearly all information related to PRs. After collection, the PR information was loaded into a database, processed into a format suitable for statistical analysis using Structured Query Language, and then analyzed.

The target metrics were as follows:

- **No. of PRs:** Number of merged PRs created by an individual
- **Total LoC:** Total LoC written by individuals. The LoC from the merged PRs were counted.
- **LoC/PR:** LoC per merged PR.
- **Comments:** Number of comments written by an individual in the PRs. To assess community activity level, comments from all PRs (open, closed, and merged) were collected.

The overall workflow of our research methodology is illustrated in Figure 3.

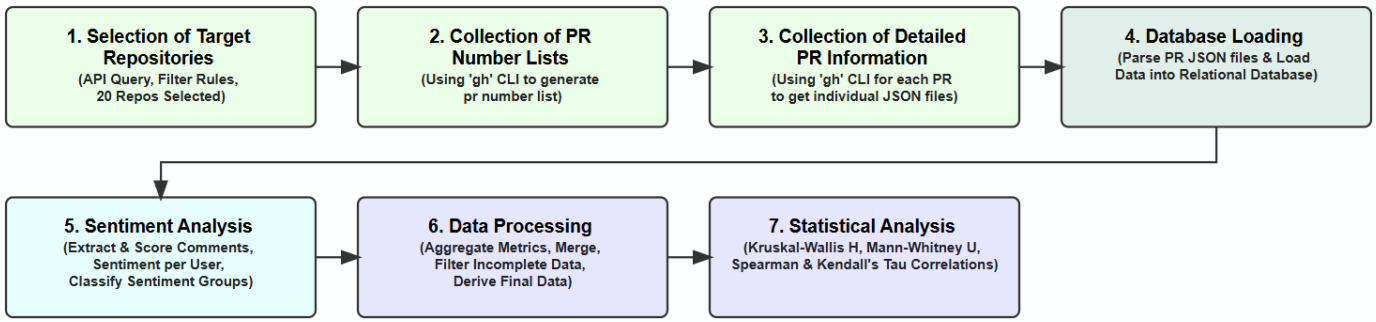

Figure 3. Research procedure.

3.1. Selection of Target Repositories

All repositories selected for this study were hosted on the GitHub platform. GitHub was chosen due to its prominence and data accessibility. Although countless repositories exist on GitHub, most are inactive projects [57]. To ensure meaningful results, we aimed to avoid those projects while identifying repositories with many contributors. Using the GitHub API, we collected lists of the top 100 repositories based on Stars and Forks. Stars indicate interest or satisfaction with a project [58], while Forks can be considered an intent to contribute to OSS via the fork-and-pull model [59]. These metrics are widely regarded as proxies for project popularity, community engagement, and visibility within the OSS ecosystem. From these lists, we selected the top ten repositories from each category (Star and Fork) based on rank, applying the following rules:

- Exclude projects that are clearly not software projects.
- If a repository appears in both of the top 100 lists, it is classified based on its higher rank. For example, “react” (ranked 10th by Star and 22nd by Fork) is classified as a Star repository.

- Exclude repositories with an "Archived" status.
- Exclude repositories not following the fork-and-pull model [21].

Based on these rules, ten projects for each of the Star and Fork categories were selected, as shown in Table 1.

Table 1. List of top ten Star and Fork repositories.

Star Top 10		Fork Top 10	
Repository Name	**Data Collection Date**	**Repository Name**	**Data Collection Date**
react	5 November 2023	tensorflow	31 October 2023
ohmyzsh	5 November 2023	bootstrap	25 October 2023
flutter	4 November 2023	opencv	30 October 2023
vscode	8 November 2023	kubernetes	29 October 2023
AutoGPT	2 November 2023	bitcoin	24 October 2023
transformers	7 November 2023	three.js	1 November 2023
next.js	4 November 2023	qmk_firmware	30 October 2023
react-native	6 November 2023	material-ui	29 October 2023
electron	2 November 2023	django	27 October 2023
stable-diffusion-webui	7 November 2023	cpython	26 October 2023

Table 1 summarizes the twenty projects selected for this study. The data collection date for each project represents the cutoff point for data collection, which encompasses all relevant data from the project's first PR on GitHub up to the specified date in 2023.

3.2. Collection of PR Number Lists

The list of PR numbers for each selected repository was collected using the GitHub API. This process utilized gh, the official GitHub command line interface, which allows for scripted interactions with GitHub features including direct API calls. The specific commands employed to retrieve and extract the PR numbers are presented in Listing 1.

Listing 1. Collecting PR number list using the "gh" command.

```
gh pr list --state all --json number --jq '.[].number' --limit
    999999999 > pr-num-list.txt
```

The execution of this command involves several key operations. The "gh pr list" initiates requests to list the PRs. The "–state all" parameter ensures that PRs in all states (open, closed, and merged) are fetched, providing a comprehensive set of PRs for potential comment analysis, even though subsequent productivity metrics such as LoC focus on the merged PRs. The "–json number" option specifies that the output should be in the JSON format and include only a number field for each PR, as this field is the unique identifier required for the next stage of data collection. The output is then piped to "–jq," which is a command-line JSON processor. The "–jq '.[].number" filter processes the JSON array, extracting only the numerical values of the number fields for each PR object. We set "limit 999999999" high in order to retrieve all available PR numbers for a given repository without encountering pagination limits. Finally, the ">" operator redirects the resulting stream of PR numbers, saving them in a file called "pr-num-list.txt."

This pr-num-list.txt file contains a simple newline-separated list of unique PR identifiers for the repository, as exemplified in Listing 2.

Listing 2. Example of collected PR number list.

```
$ cat pr-num-list.txt | head -n 10
197657
197653
```

```
197652
197651
197649
197645
197644
197641
197640
197633
```

The generated pr-num-list.txt file is crucial, as it serves as the direct input for the shell script detailed in Section 3.3, which iterates through each PR number in this list to fetch comprehensive data for each individual PR.

3.3. Collection of Detailed PR Information

Based on the PR number list obtained in Listings 1 and 2, detailed information on each PR was collected using the following script detailed in Listing 3.

Listing 3. Collecting detailed information based on PR number list.

```
#!/bin/bash

while read p; do
        gh pr view $p --json additions,assignees,author,
            autoMergeRequest,baseRefName,body,changedFiles,
            closed,closedAt,comments,commits,createdAt,deletions
            ,files,headRefName,headRefOid,headRepository,
            headRepositoryOwner,id,isCrossRepository,isDraft,
            labels, latestReviews,maintainerCanModify,
            mergeCommit,mergeStateStatus,mergeable,mergedAt,
            mergedBy,milestone,number,potentialMergeCommit,
            projectCards,projectItems,reactionGroups,
            reviewDecision,reviewRequests,reviews,state,
            statusCheckRollup,title,updatedAt,url > $p.json;
        sleep 0.8s;
done <pr-num-list.txt
```

Executing the script in Listing 3 sequentially processes each PR number from the list, collecting detailed information. This information is then output into individual JSON files named after the PR numbers. Listing 4 shows a subset of the collected information for each PR.

Listing 4. Subset of detailed information collected by Listing 3.

```
{
    "additions": 1,
    "assignees": [],
    "author": {
        "is_bot": true,
        "login": "app/"
    },
    "autoMergeRequest": null,
    "baseRefName": "master",
    "body": "TESTTTT",
```

```
"changedFiles": 1,
"closed": true,
"closedAt": "2019-03-15T21:49:12Z",
"comments": [],
"commits": [
    {
        "authoredDate": "2019-03-15T21:48:32Z",
        "authors": [
            {
                "email": "48438775+peter11232@users.noreply.
                    github.com",
                "id": "",
                "login": "",
                "name": "peter11232"
            }
        ],
        "committedDate": "2019-03-15T21:48:32Z",
        "messageBody": "TESTTTT",
        "messageHeadline": "Create Peter",
        "oid": "38e2ca9269fd897e987343f6ed452c5cb621c08d"
    }
],
"createdAt": "2019-03-15T21:48:54Z",
"deletions": 0,
"files": [
    {
        "path": "Peter",
        "additions": 1,
        "deletions": 0
    }
],
"headRefName": "patch-1",
"headRefOid": "38e2ca9269fd897e987343f6ed452c5cb621c08d",
"headRepository": null,
"headRepositoryOwner": {
    "login": ""
},
"id": "MDExOlB1bGxSZXF1ZXN0MjYxNzA4ODIw",
"isCrossRepository": true,
"isDraft": false,
"labels": [],
"latestReviews": [],
"maintainerCanModify": false,
"mergeCommit": null,
"mergeStateStatus": "DIRTY",
"mergeable": "CONFLICTING",
"mergedAt": null,
"mergedBy": null,
"milestone": null,
"number": 70593,
```

```
    "potentialMergeCommit": null,
    "projectCards": [],
    "projectItems": [],
    "reactionGroups": [],
    "reviewDecision": "",
    "reviewRequests": [],
    "reviews": [],
    "state": "CLOSED",
    "statusCheckRollup": [],
    "title": "Create Peter",
    "updatedAt": "2020-03-27T02:14:21Z",
    "url": "https://github.com/microsoft/vscode/pull/70593"
}
```


3.4. Database Loading

The collected detailed PR information files were parsed and loaded into a database to facilitate processing for research purposes. The parsing and database loading software was developed in Kotlin 1.6 [60]. Kotlin was selected for this task because of several advantageous features that benefit data processing applications. In particular, its concise syntax and expressiveness enhance code readability and maintainability. Kotlin's strong null-safety system proved especially valuable for robustly handling potentially missing or null fields encountered when parsing JSON data from the GitHub API, which helped to reduce the likelihood of runtime errors. Furthermore, its seamless interoperability with Java enabled us to use mature Java libraries for tasks such as JSON parsing and database connectivity via Java Database Connectivity (JDBC) [61]. Availability of features such as data classes simplified the modeling of structured PR information. These aspects contributed to an efficient development process for the data loading utility. Because of the frequent one-to-many relationships (e.g., PR to comments, PR to commits, and PR to assignees), the data were loaded into separate tables linked by keys.

3.5. Sentiment Analysis

Comment data were extracted from the database and sentiment analysis software was used to score the sentiment state of each comment, thereby quantitatively estimating the internal state of the system participants. SentiStrength-SE [62] was used for this purpose.

SentiStrength-SE is based on the widely used sentiment analysis tool SentiStrength [63], but provides enhanced accuracy by incorporating a specialized SE domain, making it suitable for quantitatively measuring sentiment states in OSS projects. SentiStrength-SE primarily uses a lexicon-based approach to analyze individual words and sentences, classifying them as positive, negative, or neutral based on sentiment scores. It adjusts scores based on booster words and negations (e.g., "not," "isn't," "hasn't"). It also utilizes sentence structure and contextual hints for more accurate analysis.

Although the original SentiStrength employs similar algorithms, its lexicon was trained for texts that are not always compatible with SE, leading to potential inaccuracies. It improved after using a word lexicon tailored to SE, accurately reflecting technical terminology. For example, words such as "dead," "block," or "error" are often negative in general contexts, but may not be so in SE contexts. In addition to adding to the lexicon, SentiStrength-SE incorporates refined contextual rules for more sophisticated analysis. Thanks to these adjustments, Islam and Zibran [62] were able to conduct experiments using a benchmark dataset of JIRA issue comments. In [64], SentiStrength-SE demonstrated higher precision and recall than the original SentiStrength as well as other sentiment analysis tools such as

NLTK and Stanford NLP. Previous research has utilized SentiStrength-SE [65,66]. While SentiStrength is often used in general social media analysis, SentiStrength-SE is more frequently employed in the software domain [66,67].

SentiStrength-SE outputs a sentiment score for each comment text, ranging from −4 (very negative) to +4 (very positive). To illustrate this classification, representative (anonymized) examples for each sentiment category are provided below along with their illustrative SentiStrength-SE scores:

- **Positive Example:** "Done. Briefly considered not signing you all up for installer change notifications but then did so. Fortunately:)" (Illustrative SentiStrength-SE Score: +3)
- **Neutral Example:** "Can you please resolve merge conflicts and make sure all builds are green?" (Illustrative SentiStrength-SE Score: 0)
- **Negative Example:** "IMO, I don't think it is worth the hassle to backport this. Cry out if you disagree." (Illustrative SentiStrength-SE Score: −3)

We calculated the average sentiment score for each participant (comment author) by summing the scores for all comments they wrote during the study period and dividing them by the total number of comments. To represent each contributor's overall sentiment tendency, we used the average sentiment score from all comments within the repository's observation periods. This metric was chosen for its conventional and comprehensive summary of the capabilities of the expressed sentiments. The participants were then classified into three groups based on their average sentiment scores:

- **Positive Group:** Average sentiment score > 0
- **Neutral Group:** Average sentiment score = 0
- **Negative Group:** Average sentiment score < 0.

These thresholds were derived directly from the SentiStrength-SE output scale, in which a score of 0 indicates neutral sentiment, positive scores reflect a positive sentiment, and negative scores reflect negative sentiment.

3.6. Data Processing

After collecting data, including sentiment scores, we processed them for statistical analysis in the following sequence:

1. We collected the comments and average sentiment score for each individual.
2. From the PR-related tables, we accessed the PR count, total LoC, and LoC per PR for each individual. Only merged PRs were considered for these metrics, as only the merged PRs were ultimately reflected in project outcomes. LoC was used because it is the most accessible and intuitive metric for measuring productivity in OSS, allowing for the collection of large amounts of data. This metric overcomes the difficulty of using other productivity measurement techniques such as in-depth observation or function points, which are challenging to use in OSS development contexts owing to the diverse affiliations (or lack thereof) of participants who are not necessarily part of the same organization.
3. We merged the two datasets (comments and PR data) based on individual participant identifiers within each repository. Participants were included in the analysis if they contributed to at least one merged PR and authored at least one comment within the respective repository. Individuals with only PR data (no comments) or only comments (no PRs) for a given repository were excluded from the dataset.
4. We derived the final data containing PR count, total LoC, LoC per PR, average comment sentiment score, and comment count for each individual.

5. We classified each individual based on their average comment sentiment score into Positive (>0), Neutral (0), or Negative (<0).

3.7. Statistical Analysis

Statistical analyses were performed using the processed data. Normality tests (Shapiro–Wilk and Anderson–Darling) indicated that the distributions of the key variables were non-normal; thus, we selected non-parametric statistical methods. These non-parametric tests are appropriate for data that do not meet normality assumptions, and are inherently more robust than their parametric counterparts because they reduce the influence of outliers. After dividing participants into the three sentiment groups (Positive, Neutral, Negative), the following statistical analyses were conducted for each group concerning PR count, total LoC, LoC per PR, and comment count. Two different correlation analysis methods were used to examine the trends more closely under non-parametric assumptions.

- Kruskal–Wallis H test [68] (comparison among three groups)
- Mann–Whitney U test [69] (pairwise comparison between groups)
- Spearman's ρ [70] and Kendall's tau [71] correlation analyses

Correlation analysis was performed to determine if correlations exist between individual metrics, and if so, their strengths. This analysis was performed for the pairs: PR count vs. LoC, PR count vs. PR comments, and LoC vs. PR comments. It was conducted using Python 3.11 [72]. The numerical calculations used pandas 2.1.1 [73], and statistical tests used scipy 1.11.3 [74].

4. Results

The analysis suggests that participants with positive sentiment are generally associated with higher observed productivity and more active communication. Interestingly, participants expressing negative sentiment also appeared to be associated with higher productivity compared with that of the neutral group, although their communication activity was generally lower than that of the positive group but higher than that of the neutral group.

4.1. Observed Differences in Productivity and Communication Across Sentiment Groups

To address **RQ1** and **RQ2**, we analyzed the number of PRs, total LoC, LoC/PR, and number of comments by participants classified as positive, neutral, or negative sentiment groups.

Aggregated results for all twenty projects as well as for the top ten Star and top 10 Fork repositories are listed in Tables 2–4. These tables show the number of contributors (unique participants) and the median and mean values for each metric categorized according to sentiment group. Before delving into each descriptive statistics table (Tables 2–4), we must highlight the consistent overarching pattern we observed: participants exhibiting positive sentiments generally demonstrated the highest median and mean values for productivity indicators (number of PRs, total LoC, LoC/PR) and communication volume (comments). Notably, participants with negative sentiments tended to outperform the neutral group in these areas, with the neutral group consistently ranking the lowest.

The interpretations of the metrics in these tables are as follows:

- For "No. of PRs," the "Median" indicates the central value when participants within a sentiment group are ranked by the number of merged PRs they created: 50% of the participants created many PRs or fewer and 50% created many PRs or more. The "Mean" is the average number of merged PRs per participant in that group.

- For "Total LoC," the "Median" and "Mean" refer to the respective central and average values of the total LoC (from merged PRs only) written by each participant within each sentiment group.
- For "LoC/PR," each participant has an average number of LoC per their own merged PR(s). The "Median" and "Mean" values in the tables represent the central and average values of these individual participant-level averages within each sentiment group.
- For "Comments," as exemplified by the reviewer's query, a "Median" of 2.0 for a group indicates that 50% of the participants in that group wrote two or fewer comments in PR discussions and that 50% wrote two or more. A "Mean" of 6.45 would represent the average number of comments written per participant in that group.

Table 2 first presents these descriptive statistics aggregated across all twenty repositories.

Table 2. Median and mean values for each sentiment group across all twenty repositories.

Category		Number of Contributors	Median	Mean
total		24,607	-	-
Positive	No. of PRs	12,262	2.0	17.37
	Total LoC		86.0	9865.64
	LoC/PR		39.12	332.31
	Comments		6.0	91.65
Neutral	No. of PRs	7647	1.0	3.22
	Total LoC		17.0	620.69
	LoC/PR		12.0	135.34
	Comments		2.0	6.45
Negative	No. of PRs	4698	2.0	10.04
	Total LoC		51.0	7661.10
	LoC/PR		28.0	262.59
	Comments		5.0	46.23

Table 3 details the same metrics for the top ten Star repositories.

Table 3. Median and mean values for each sentiment group in the top ten Star repositories.

Category		Number of Contributors	Median	Mean
total		8321	-	-
Positive	No. of PRs	4232	1.0	16.65
	Total LoC		62.5	5832.69
	LoC/PR		34.4	201.42
	Comments		4.0	55.96
Neutral	No. of PRs	2632	1.0	5.14
	Total LoC		17.0	336.69
	LoC/PR		13.0	81.47
	Comments		1.0	8.13
Negative	No. of PRs	1457	1.0	11.85
	Total LoC		42.0	3372.37
	LoC/PR		25.0	151.09
	Comments		4.0	30.18

Similarly, Table 4 provides descriptive statistics for the top ten Fork repositories.

Table 4. Median and mean values for each sentiment group in the top ten Fork repositories.

Category		Number of Contributors	Median	Mean
total		16,286	-	-
Positive	No. of PRs	8030	2.0	17.75
	Total LoC		104.0	11,991.10
	LoC/PR		43.05	401.29
	Comments		7.0	110.47
Neutral	No. of PRs	5015	1.0	2.21
	Total LoC		17.0	769.74
	LoC/PR		12.0	163.62
	Comments		2.0	5.58
Negative	No. of PRs	3241	2.0	9.22
	Total LoC		58.0	9589.11
	LoC/PR		29.0	312.71
	Comments		6.0	53.45

The positive sentiment group was consistently observed across all analysis groups (all twenty, top ten Star, top ten Fork) to have higher median and mean values than the other two groups for all measured productivity indicators and communication volume. These observations offer initial insights into **RQ1** and **RQ2**, suggesting that participants with a positive attitude tend to show higher productivity in code contribution and more active communication.

A detailed examination of the descriptive statistics for the aggregates of all twenty repositories (Table 2) revealed clear trends and established the primary observed pattern. Participants in the positive sentiment group exhibited the highest central tendency and average values across all metrics: a median of 2.0 PRs (mean 17.37), 86.0 total LoC (mean 9865.64), 39.12 LoC/PR (mean 332.31), and 6.0 comments (mean 91.65). Conversely, the neutral sentiment group consistently presented the lowest values: median 1.0 PRs (mean 3.22), 17.0 total LoC (mean 620.69), 12.0 LoC/PR (mean 135.34), and 2.0 comments (mean 6.45). The negative sentiment group occupied an intermediate position, with values generally higher than that of the neutral group but lower than that of the positive group: median 2.0 PRs (mean 10.04), 51.0 total LoC (mean 7661.10), 28.0 LoC/PR (mean 262.59), and 5.0 comments (mean 46.23).

This primary Positive > Negative > Neutral ranking observed in the combined dataset was largely mirrored when analyzing the subsets of the top ten Star repositories (Table 3) and top ten Fork repositories (Table 4). In both of these distinct subsets, positive sentiment consistently registered the highest values for key metrics (for Star repositories, mean total LoC of 5832.69; for Fork repositories, mean total LoC of 11,991.10), while the neutral group typically showed the lowest values (for Star repositories, mean total LoC of 336.69; for Fork repositories, mean total LoC of 769.74). The negative sentiment group maintains its intermediate position in these subsets. This consistency across different repository groupings reinforces the robustness of the observed Positive > Negative > Neutral hierarchy for the productivity and interaction metrics based on these descriptive statistics.

These descriptive findings highlight a consistent pattern across different repository groupings, suggesting that positive sentiment is associated with higher productivity and communication, whereas neutral sentiment is associated with the lowest. Although less productive than the positive group, the negative sentiment group still surpassed the productivity of the neutral group. These observations set the stage for subsequent statistical hypothesis testing.

Prior to examining the outcomes of the Kruskal–Wallis H and Mann–Whitney U tests presented in Tables 5–7, we must note that these statistical analyses broadly corroborated

the patterns identified in the descriptive statistics. These tables collectively demonstrate that the observed differences in the productivity and communication metrics across the three sentiment groups were mostly statistically significant. A recurring finding is that both the positive and negative sentiment groups tended to show higher activity levels than those of the neutral sentiment group.

The outcomes of the Kruskal–Wallis H and Mann–Whitney U tests for all twenty repositories are shown in Table 5.

Table 5. Test results for all twenty repositories.

Category	Kruskal–Wallis H Test			Mann–Whitney U Test								
				Positive-Neutral			Neutral-Negative			Positive-Negative		
	Stat.	ϵ^2	*p*-Value	Stat.	r_{rb}	*p*-Value	Stat.	r_{rb}	*p*-Value	Stat.	r_{rb}	*p*-Value
No. of PRs	1586.55	0.064	<0.001	60,981,076.0	0.300	<0.001	13,562,406.0	−0.244	<0.001	26,904,913.0	−0.065	<0.001
Total LoC	1790.90	0.072	<0.001	63,403,497.0	0.352	<0.001	13,324,718.0	−0.258	<0.001	25,763,927.5	−0.105	<0.001
LoC/PR	1172.00	0.047	<0.001	60,285,774.5	0.285	<0.001	14,415,289.5	−0.197	<0.001	26,014,063.0	−0.096	<0.001
Comments	3971.18	0.161	<0.001	70,109,454.5	0.495	<0.001	9,059,851.5	−0.495	<0.001	27,658,384.5	−0.039	<0.001

Table 6 presents the corresponding test results for the top ten Star repositories.

Table 6. Test results for top ten Star repositories.

Category	Kruskal–Wallis H Test			Mann–Whitney U Test								
				Positive-Neutral			Neutral-Negative			Positive-Negative		
	Stat.	ϵ^2	*p*-Value	Stat.	r_{rb}	*p*-Value	Stat.	r_{rb}	*p*-Value	Stat.	r_{rb}	*p*-Value
No. of PRs	394.82	0.047	<0.001	6,973,654.0	0.252	<0.001	1,548,296.5	−0.192	<0.001	2,894,748.0	−0.061	<0.01
Total LoC	460.53	0.055	<0.001	7,270,662.5	0.305	<0.001	1,501,109.0	−0.217	<0.001	2,789,273.5	−0.095	<0.001
LoC/PR	304.90	0.036	<0.001	6,954,815.0	0.248	<0.001	1,596,388.0	−0.167	<0.001	2,811,140.5	−0.088	<0.001
Comments	1300.30	0.156	<0.001	8,260,316.5	0.483	<0.001	995,769.0	−0.480	<0.001	2,969,978.5	−0.036	0.035 *

* Not significant because it was less than the usual *p*-value criterion of 0.05 but more than the criterion based on the Bonferroni correction [75,76]. The significance level for the Mann–Whitney U tests was set at 0.017 using the Bonferroni correction.

The test results for the top ten Fork repositories are presented in Table 7.

Table 7. Test results for top ten Fork repositories.

Category	Kruskal–Wallis H Test			Mann–Whitney U Test								
				Positive-Neutral			Neutral-Negative			Positive-Negative		
	Stat.	ϵ^2	*p*-Value	Stat.	r_{rb}	*p*-Value	Stat.	r_{rb}	*p*-Value	Stat.	r_{rb}	*p*-Value
No. of PRs	1211.81	0.074	<0.001	26,721,644.0	0.327	<0.001	5,952,711.0	−0.267	<0.001	12,032,923.5	−0.075	<0.001
Total LoC	1345.71	0.082	<0.001	27,723,557.0	0.376	<0.001	5,891,048.0	−0.275	<0.001	11,501,190.5	−0.116	<0.001
LoC/PR	873.12	0.053	<0.001	26,267,469.5	0.304	<0.001	6,420,072.5	−0.210	<0.001	11,654,302.5	−0.104	<0.001
Comments	2713.92	0.166	<0.001	30,344,676.5	0.507	<0.001	4,051,938.5	−0.501	<0.001	12,328,809.5	−0.052	<0.001

The Kruskal–Wallis H tests consistently indicated statistically significant overall differences ($p < 0.001$) among the three sentiment groups across all repository sets and metrics (Tables 5–7). The associated ϵ^2 values generally suggest moderate [77] overall effects of sentiment group on productivity metrics, with ϵ^2 values typically ranging from 0.04 to 0.08. The overall effect of sentiment group is consistently larger for Comments, with ϵ^2 values indicating moderate to relatively strong effects.

The Mann–Whitney U test further elucidates these differences. The positive group consistently ranks significantly higher than the neutral group across all metrics and repository sets, with r_{rb} mostly in the small-to-large range [78]. Similarly, the negative group

consistently ranks significantly higher than the neutral group, with r_{rb} values indicating small-to-large effects and notably large effects for Comments. Direct comparisons between the positive and negative groups generally yield weak and often moderate effect sizes; for instance, $r_{rb} = -0.065$ for the number of PRs across all twenty repositories, suggesting that the negative group ranks slightly higher, although the effect is negligible. This pattern of the positive and negative groups outperforming the neutral group is largely consistent, with more nuanced differences between results for the positive and negative groups. An exception is noted for Comments in the top ten Star repositories, in which the difference between positive and negative groups is not significant after the Bonferroni correction.

Overall, these statistical tests and effect sizes support the exploration of **RQ1** and **RQ2**. Significant differences and associations of practical note exist in productivity and communication metrics based on participant sentiment. Both the positive and negative sentiment groups generally show higher activity levels than the neutral group, with the distinctions particularly pronounced for communication volume. While descriptive statistics suggest a Positive > Negative > Neutral trend, rank-based effect sizes indicate more comparable performance between the positive and negative groups in direct comparisons, with both clearly outperforming the neutral group.

4.2. Observed Correlation Patterns and Potential for Feedback Loops

To address **RQ3**, we investigated the observed correlation patterns between participant sentiment, productivity indicators, and communication volume. We also explored the potential effects of feedback loops. In addition, we performed Spearman's ρ rank and Kendall's tau correlation analyses, examining correlations within each sentiment group between number of PRs vs. Total LoC, number of PRs vs. Comments, and Total LoC vs. Comments. Before detailing the correlation coefficients in Tables 8–10, we must call attention to a general pattern that emerges from these analyses. These tables consistently indicate statistically significant positive correlations between various productivity indicators and communication volume within each sentiment group. Furthermore, the strength of these positive correlations tends to be most pronounced for participants in the positive sentiment group, followed by the negative sentiment group. The correlation is weakest for the neutral sentiment group. This finding suggests varying degrees of interconnectedness between these two activities based on sentiment.

Table 8 displays the Spearman's ρ and Kendall's tau correlation coefficients for the dataset comprising all twenty repositories.

Table 8. Correlation analysis results for all twenty repositories.

Category		Spearman's ρ		Kendall's Tau	
		Correlation Coefficient	***p*-Value**	**Correlation Coefficient**	***p*-Value**
Positive	No. of PRs vs. Total LoC	0.6944	<0.001	0.5549	<0.001
	No. of PRs vs. Comments	0.6793	<0.001	0.5541	<0.001
	Total LoC vs. Comments	0.6025	<0.001	0.4465	<0.001
Neutral	No. of PRs vs. Total LoC	0.4585	<0.001	0.3722	<0.001
	No. of PRs vs. Comments	0.3626	<0.001	0.3156	<0.001
	Total LoC vs. Comments	0.3427	<0.001	0.2614	<0.001
Negative	No. of PRs vs. Total LoC	0.6471	<0.001	0.5157	<0.001
	No. of PRs vs. Comments	0.5908	<0.001	0.4779	<0.001
	Total LoC vs. Comments	0.5294	<0.001	0.3875	<0.001

The correlation results for the top ten Star repositories are presented in Table 9.

Table 9. Correlation analysis results for top ten Star repositories.

Category		Spearman's ρ		Kendall's Tau	
		Correlation Coefficient	*p*-Value	Correlation Coefficient	*p*-Value
Positive	No. of PRs vs. Total LoC	0.6610	<0.001	0.5335	<0.001
	No. of PRs vs. Comments	0.6076	<0.001	0.4992	<0.001
	Total LoC vs. Comments	0.5687	<0.001	0.4247	<0.001
Neutral	No. of PRs vs. Total LoC	0.4259	<0.001	0.3458	<0.001
	No. of PRs vs. Comments	0.3176	<0.001	0.2808	<0.001
	Total LoC vs. Comments	0.3066	<0.001	0.2360	<0.001
Negative	No. of PRs vs. Total LoC	0.6187	<0.001	0.4999	<0.001
	No. of PRs vs. Comments	0.5344	<0.001	0.4388	<0.001
	Total LoC vs. Comments	0.5190	<0.001	0.3841	<0.001

Table 10 shows the correlation coefficients for the top ten Fork repositories.

Table 10. Correlation analysis results for top ten Fork repositories.

Category		Spearman's ρ		Kendall's Tau	
		Correlation Coefficient	*p*-Value	Correlation Coefficient	*p*-Value
Positive	No. of PRs vs. Total LoC	0.7077	<0.001	0.5623	<0.001
	No. of PRs vs. Comments	0.7113	<0.001	0.5799	<0.001
	Total LoC vs. Comments	0.6159	<0.001	0.4550	<0.001
Neutral	No. of PRs vs. Total LoC	0.4740	<0.001	0.3847	<0.001
	No. of PRs vs. Comments	0.3840	<0.001	0.3319	<0.001
	Total LoC vs. Comments	0.3617	<0.001	0.2752	<0.001
Negative	No. of PRs vs. Total LoC	0.6552	<0.001	0.5192	<0.001
	No. of PRs vs. Comments	0.6097	<0.001	0.4912	<0.001
	Total LoC vs. Comments	0.5292	<0.001	0.3864	<0.001

The correlation analyses reveal statistically significant positive correlations ($p < 0.001$ for all reported coefficients) between all examined pairs of metrics across all sentiment groups and repository types. Specifically, for **RQ3**, the strength of these observed correlations appears to vary by sentiment group. In all cases, the positive sentiment group shows the strongest correlation. For instance, in the "All twenty repositories" group (Table 8), Spearman's ρ for the positive group ranges from 0.6025 to 0.6944. These coefficients suggest moderate-to-strong positive monotonic associations. Practically, a ρ value such as 0.6944 (e.g., between No. of PRs and Total LoC) indicates that within the positive sentiment group, individuals who rank higher in terms of the number of PRs they submit also quite consistently rank higher in terms of total contribution to LoC. This finding indicates a notable and fairly dependable pattern in which these productive activities are strongly coupled for individuals in this group.

The negative group also shows moderate correlations (Spearman's ρ from 0.5294 to 0.6471). For example, a ρ of 0.6471 (No. of PRs vs. Total LoC) indicates a relatively strong tendency for these metrics to align positively. This tendency suggests that even among participants with negative sentiments, higher engagement in one productive activity is often mirrored by

higher engagement in others, although perhaps with slightly less consistency or strength than in the positive group. In contrast, the neutral group exhibits consistently weaker correlations (Spearman's ρ from 0.3427 to 0.4585). While statistically significant, these values indicate low-to-moderate positive monotonic relationships. In other words, the neutral group's metrics tend to increase together, but the relationship is less consistent; an individual's rank on one metric is a less reliable indicator of their rank on another compared to the positive or negative sentiment groups. This pattern of Positive > Negative > Neutral in terms of correlation strength is generally consistent across both correlation methods (Spearman's ρ and Kendall's tau) and across the different repository subsets.

These observations related to **RQ3** suggest clear correlation patterns: individuals who submit more PRs are often observed to contribute more to LoC and engage in more communication. Reflecting how consistently these behaviors co-occur, this interconnectedness appears most pronounced and dependable within the positive sentiment group. Stronger correlations within the positive group could be an indication of the speculative idea of positive feedback loops; the possibility invites further exploration. Higher productivity may be associated with more positive interactions, which in turn could be related to positive sentiment and further associated with contributions and communication. Conversely, the weaker correlations result in less consistent co-occurrence of these behaviors in the neutral group, which could suggest that such reinforcing cycles are less active or prominent for these participants.

In summary, the results suggest a significant association between participant sentiment and both productivity and communication levels in OSS projects. Positive sentiment is observed to align with higher outputs and stronger interconnections between different contribution activities, potentially indicating a more vibrant and possibly self-reinforcing engagement pattern.

5. Discussion

This study approached OSS development projects as an STS, empirically analyzing the relationship between participant attitude (reflecting sentiment) as a a key element of the system and key outcome indicators such as productivity and intra-system interactions. The results offer several insights into the dynamics of OSS systems.

As detailed in Section 4, our empirical findings consistently indicate a significant association between collaborator sentiment and both productivity and interaction metrics. Most notably, a clear hierarchical pattern emerged: participants with positive sentiment exhibited the highest performance, followed by those with negative sentiments, whereas the neutral group consistently showed the lowest levels of contribution and engagement. This Positive > Negative > Neutral ranking warrants careful consideration. This pattern suggests that participants with positive emotions tend to contribute most actively, while those with negative emotions exhibit higher productivity than those in the neutral group.

This counterintuitive result is particularly noteworthy. It implies that expressing emotions, even negative ones, may be associated with higher engagement compared to neutrality. This result also suggests that negative sentiment does not necessarily translate to lower productivity. Perhaps participants who express any sentiment are relatively more engaged contributors than those who remain neutral. For instance, a participant who takes the time to articulate a strong negative opinion on a particular issue or proposed change, even if the tone is critical, demonstrates engagement with the project's specifics. Such critical feedback, even though expressed negatively, can stem from genuine concern for the project's quality or direction. Moreover, it can trigger important discussions, highlight overlooked problems, or lead to necessary revisions. This form of active albeit negatively-valenced participation can ultimately be more constructive than passive neutrality, which

may signify lack of interest or superficial contributions that do not elicit strong emotional responses. Participants in the neutral group may include those who make minor routine contributions with minimal interaction or those who are primarily observing without investing emotionally or intellectually in the ongoing development dialogues. Therefore, the act of expressing sentiments, regardless of polarity, could be a proxy for a certain threshold of engagement that surpasses that of truly neutral or disengaged participants. However, distinguishing constructive negative feedback from overtly toxic or disruptive behavior is crucial, as the latter is likely detrimental to the system.

Based on these results, we observed a correlation between positive attitudes and productivity in OSS project collaboration and participation. This finding may suggest that fostering an environment that promotes positive emotions among developers could be linked to enhanced productivity-related outcomes. Furthermore, the stronger correlations observed within the positive sentiment group (see Section 4) are consistent with the speculative idea of positive feedback loops, inviting further exploration. Our framing of OSS development as an STS (see Section 2.1) allows us to consider feedback mechanisms that potentially shape system behavior. For instance, Dwyer [18] noted the importance of feedback loops as a component of STS theory for understanding system adaptation. Building on Dwyer's work, Sterman altered the perspective of system dynamics; he described how complex systems can be governed by feedback, including reinforcing (positive) loops in which initial changes may be amplified, potentially contributing to virtuous cycles [79]. Thus, we can tentatively hypothesize from the stronger interconnections in our positive sentiment group that positive sentiment contributes to fostering more effective collaboration and higher productivity. These positive outcomes could in turn conceivably generate positive experiences (e.g., successful PR merges and community recognition), which may then play a role in potentially reinforcing positive sentiments and continued engagement. Such a speculative line of reasoning regarding potential reinforcing cycles could resonate with the general principles of feedback in STS, as noted by Dwyer [18], as well as with the dynamics of reinforcing loops in complex systems as explained by Sterman [79]. It may also tentatively complement our existing reference to feedback by Storey et al. [17], and could be hypothetically linked to theories such as broaden-and-build theory [8], which describes how positive emotions might create upward spirals. Conversely, the weaker correlations found in the neutral group suggest that such reinforcing cycles are less active or prominent for these participants. However, we must reiterate that while our correlational findings are suggestive of such possibilities, they do not establish causality. Further research, possibly employing longitudinal or simulation modeling approaches, is needed to explore more definitively the mechanics and impact of these potential loops.

Previous research has relied on indirect methods such as sentiment analysis of email threads [15], examining the relationship between PR sentiment and approval time/likelihood [42], and studying the connection between emotions during bug fixing and resolution time/frequency [13,14]. These indirect approaches were partially necessitated by the challenges of quantifying aspects of software development, knowledge-intensive activities, and the statistical difficulties posed by metrics such as sentiment scores and productivity measures. However, our study demonstrates that using direct indicators such as sentiment scores and LoC, applying nonparametric statistics, and establishing comparison groups can empirically elucidate differences and correlations between these factors directly. Thus, these approaches offer a valuable methodological insight for future research in this area.

Moreover, active engagement in OSS activities can foster open innovation [4] and contribute to sustainability goals [6,7]. These implications extend beyond companies; open source-appropriate technology based on OSS can contribute significantly to global sustainability [80]. A United Nations Development Programme (UNDP) report [81] sug-

gested that OSS has the potential to lower information and communications technology entry barriers for developing countries worldwide, ultimately contributing to sustainable human development.

Furthermore, applying affective events theory [82], which posits that workplace events trigger emotional reactions that influence attitudes and behaviors, allows us to view OSS development as distributed collaboration within a virtual organization. This perspective opens avenues for interdisciplinary research between organizational behavior and SE. By recognizing the correlation between developer emotions, attitudes, and productivity, companies can potentially improve collaborative environments and cultures to boost OSS project productivity and community vitality. Such efforts could not only advance corporate goals of open innovation and sustainability but also yield benefits on a global scale.

5.1. Limitations

Although this study has provided valuable insights into the relationship between collaborator sentiment and productivity in OSS projects, several limitations should be considered when interpreting the findings:

- The quantification of human emotion using SentiStrength-SE may involve some distortion. Similarly, the chosen productivity metrics (LoC and PR) simplify complex technical contributions, potentially obscuring the nuances of work and its full impacts.
- This study was correlational and did not establish causation. Whether sentiment influences productivity, productivity influences sentiment, or unobserved factors affect both is unclear. Establishing causality would require longitudinal or experimental designs.
- The consistently lowest performance of the neutral group across all metrics lacks a definitive explanation. This result might be linked to a higher prevalence of one-off contributors less inclined to express sentiment; however, this requires further investigation.
- The focus on top-tier GitHub projects was owing to the impossibility of analyzing all repositories, and may limit the generalizability of our findings to the broader and more diverse OSS ecosystem.
- This study did not control for potential confounding variables such as project size, application domain, or individual contributor experience. These factors can interact with sentiment and productivity in complex ways (e.g., different communication norms in larger projects and varying sentiment/productivity patterns based on experience). Future studies should address this issue by incorporating such variables.
- Systematic unavailability of participants' demographic data (e.g., age, sex, location, professional background) or detailed contextual information on GitHub limited our ability to explore these factors.
- Average sentiment scores were computed as the arithmetic mean of all comments per participant without normalizing for the total volume of comments; consequently, a participant with one strongly valenced comment might be grouped with another having many mildly valenced comments if their averages were similar.
- All analyses treated contributor–repository instances as distinct data points. This approach implies that an individual who contributed to multiple projects within the sample was represented by multiple observations in this specific pooled dataset. Such non-independent observations could potentially impact the reliability of these specific results.

Addressing these limitations offers promising directions for future research aiming for a more comprehensive understanding of socio-technical dynamics in OSS development.

6. Conclusions

This study treated the OSS development environment as a socio-technical system. To overcome the limitations of indirect productivity metrics and associated uncertainties, we analyzed the relationship between sentiment and productivity using direct indicators (LoC, sentiment scores) and nonparametric statistical techniques. We demonstrated that this direct approach can effectively reveal correlations. Our analysis empirically confirmed that participants with positive sentiment exhibited higher code contribution volumes and community participation levels on average. Interestingly, participants with negative sentiment also showed higher productivity than the neutral group, although generally less than the positive group.

These findings highlight a significant correlation between developer sentiment and productivity within corporate OSS projects. The observation that developers with positive sentiment tend to show high productivity and community contributions suggests that companies might benefit from considering measures to help developers maintain positive sentiment. Because of the possible link to productivity improvements, strategies such as providing constructive feedback, recognition, and performance-based rewards may motivate developers to participate in OSS more positively. Activating OSS engagement in this way can contribute to open innovation and enhanced sustainability at the corporate level, and can also potentially extend positive impacts to national or global sustainability efforts.

Additionally, while emotion-related research has traditionally been the domain of organizational behavior, the availability of large-scale text data from platforms such as GitHub enables quantitative analysis of the emotion–productivity link in OSS development using real-world data. This advance offers value for future interdisciplinary research.

Although this study has provided notable findings, it is important to also acknowledge the limitations discussed in Section 5.1. These limitations, particularly those concerning causality, generalizability, and the measurement of complex constructs, underscore valuable opportunities for future research based on expanded scope or refined methodologies. Key directions for future work include the following:

- Collecting data on individual activity duration and contribution patterns to verify whether neutral participants primarily consist of one-off contributors or exhibit significantly shorter engagement periods.
- Examining whether individuals exhibit consistent or varying sentiment across the different OSS projects to which they contribute and how this variation relates to their productivity in each project.
- Developing or applying robust methodologies to classify participants into distinct roles within OSS projects (e.g., core developers, peripheral contributors, maintainers, and issue reporters) and analyze how sentiment expression and its impact on productivity and project dynamics vary across these different roles. This classification could inform more tailored community management strategies.
- Investigating the impact of comment volume on average sentiment scores and its relationship with productivity, potentially by comparing sentiment profiles of participants with high versus low comment activity or by applying weighting schemes to sentiment aggregation.
- Investigating the stability or fluctuation of an individual's sentiment across different contributions or over time. Understanding these temporal dynamics of sentiment is a promising direction that could provide deeper insights.

Such investigations would further extend the findings of the current study and provide valuable insight into the socio-technical dynamics of OSS development.

In conclusion, participant sentiment is a factor that influences both the vitality of the OSS development system and overall performance within it. Based on this systemic

understanding, developing and applying effective management strategies can contribute not only to the success of individual projects but also to the advance of open innovation and a sustainable technological ecosystem.

Author Contributions: Conceptualization, J.L. and K.C.; methodology, J.L.; software, J.L.; validation, J.L.; formal analysis, J.L.; writing—original draft preparation, J.L.; writing—review and editing, J.L. and K.C.; supervision, K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly available datasets were analyzed in this study. The data can be found at: https://www.github.com.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Fuggetta, A. Open Source Software—An Evaluation. *J. Syst. Softw.* **2003**, *66*, 77–90. [CrossRef]
2. Ducheneaut, N. Socialization in an Open Source Software Community: A Socio-Technical Analysis. *Comput. Support. Coop. Work. (CSCW)* **2005**, *14*, 323–368. [CrossRef]
3. Baxter, G.; Sommerville, I. Socio-technical systems: From design methods to systems engineering. *Interact. Comput.* **2011**, *23*, 4–17. [CrossRef]
4. West, J.; Gallagher, S. Challenges of Open Innovation: The Paradox of Firm Investment in Open-Source Software. *R&D Manag.* **2006**, *36*, 319–331. [CrossRef]
5. Guterres, A. *Roadmap for Digital Cooperation*; United Nations: New York, NY, USA, 2020.
6. World Benchmarking Alliance. *Digital Inclusion Benchmark 2023 Insights Report*; World Benchmarking Alliance: Amsterdam, The Netherlands, 2023.
7. World Benchmarking Alliance. *Digital Inclusion Benchmark 2021 Scoring Guidelines*; World Benchmarking Alliance: Amsterdam, The Netherlands, 2021.
8. Fredrickson, B.L. The role of positive emotions in positive psychology: The broaden-and-build theory of positive emotions. *Am. Psychol.* **2001**, *56*, 218. [CrossRef] [PubMed]
9. Miller, C.; Cohen, S.; Klug, D.; Vasilescu, B.; Kästner, C. "Did you miss my comment or what?": Understanding toxicity in open source discussions. In Proceedings of the 44th International Conference on Software Engineering, Pittsburgh, PA, USA, 21–29 May 2022; pp. 710–722. [CrossRef]
10. Cohen, S. Contextualizing toxicity in open source: A qualitative study. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, 23–28 August 2021; pp. 1669–1671. [CrossRef]
11. Girardi, D.; Lanubile, F.; Novielli, N.; Serebrenik, A. Emotions and Perceived Productivity of Software Developers at the Workplace. *IEEE Trans. Softw. Eng.* **2022**, *48*, 3326–3341. [CrossRef]
12. Tulili, T.R.; Rastogi, A.; Capiluppi, A. Exploring turnover, retention and growth in an OSS Ecosystem. *arXiv* **2025**, arXiv:2504.16483.
13. Ortu, M.; Adams, B.; Destefanis, G.; Tourani, P.; Marchesi, M.; Tonelli, R. Are Bullies More Productive? Empirical Study of Affectiveness vs. Issue Fixing Time. In Proceedings of the 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories, Florence, Italy, 16–17 May 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 303–313. [CrossRef]
14. Carige Junior, R.; Carneiro, G. Impact of Developers Sentiments on Practices and Artifacts in Open Source Software Projects: A Systematic Literature Review. In Proceedings of the 22nd International Conference on Enterprise Information Systems. SCITEPRESS—Science and Technology Publications, Prague, Czech Republic, 5–7 May 2020; pp. 31–42. [CrossRef]
15. Ferreira, I.; Stewart, K.; German, D.; Adams, B. A Longitudinal Study on the Maintainers' Sentiment of a Large Scale Open Source Ecosystem. In Proceedings of the 2019 IEEE/ACM 4th International Workshop on Emotion Awareness in Software Engineering (SEmotion), Montreal, QC, Canada, 28 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 17–22. [CrossRef]
16. Luthans, F. The need for and meaning of positive organizational behavior. *J. Organ. Behav. Int. J. Ind. Occup. Organ. Psychol. Behav.* **2002**, *23*, 695–706. [CrossRef]
17. Storey, M.A.; Singer, L.; Cleary, B.; Figueira Filho, F.; Zagalsky, A. The (R) Evolution of social media in software engineering. In Proceedings of the Future of Software Engineering Proceedings, Hyderabad India, 31 May–7 June 2014; pp. 100–116. [CrossRef]
18. Dwyer, C. Socio-technical Systems Theory and Environmental Sustainability. *Sprouts* **2011**, *3–5*.

19. Trist, E.L.; Bamforth, K.W. Some Social and Psychological Consequences of the Longwall Method of Coal-Getting: An Examination of the Psychological Situation and Defences of a Work Group in Relation to the Social Structure and Technological Content of the Work System. *Hum. Relations* **1951**, *4*, 3–38. [CrossRef]
20. Scacchi, W. Understanding the requirements for developing open source software systems. *IEE Proc. Softw.* **2002**, *149*, 24. [CrossRef]
21. Padhye, R.; Mani, S.; Sinha, V.S. A Study of External Community Contribution to Open-Source Projects on GitHub. In Proceedings of the 11th Working Conference on Mining Software Repositories, Hyderabad, India, 31 May–1 June 2014; ACM: New York, NY, USA, 2014; pp. 332–335. [CrossRef]
22. Soares, D.M.; De Lima Júnior, M.L.; Murta, L.; Plastino, A. Acceptance Factors of Pull Requests in Open-Source Projects. In Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, 13–17 April 2015; ACM: New York, NY, USA, 2015; pp. 1541–1546. [CrossRef]
23. Gousios, G.; Pinzger, M.; Deursen, A.V. An Exploratory Study of the Pull-Based Software Development Model. In Proceedings of the 36th International Conference on Software Engineering, Hyderabad India, 31 May–7 June 2014; ACM: New York, NY, USA, 2014; pp. 345–355. [CrossRef]
24. Guo, Y.; Leitner, P. Studying the Impact of CI on Pull Request Delivery Time in Open Source Projects—A Conceptual Replication. *PeerJ Comput. Sci.* **2019**, *5*, e245. [CrossRef]
25. LINE. Enable to Receive Compressed Request from Client by Joonhaeng. Pull Request #3087. LINE/Armeria. Available online: https://github.com/line/armeria/pull/3087 (accessed on 30 May 2025).
26. Meyer, A.N.; Fritz, T.; Murphy, G.C.; Zimmermann, T. Software Developers' Perceptions of Productivity. In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, Hong Kong, China, 16–21 November 2014; ACM: New York, NY, USA, 2014; pp. 19–29. [CrossRef]
27. Zhou, M.; Mockus, A. Developer Fluency: Achieving True Mastery in Software Projects. In Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering, Santa Fe, NM, USA, 7–11 November 2010; ACM: New York, NY, USA, 2010; pp. 137–146. [CrossRef]
28. Kieburtz, R.; McKinney, L.; Bell, J.; Hook, J.; Kotov, A.; Lewis, J.; Oliva, D.; Sheard, T.; Smith, I.; Walton, L. A Software Engineering Experiment in Software Component Generation. In Proceedings of the IEEE 18th International Conference on Software Engineering, Berlin, Germany, 25–30 March 1996; pp. 542–552. [CrossRef]
29. Devanbu, P.; Karstu, S.; Melo, W.; Thomas, W. Analytical and Empirical Evaluation of Software Reuse Metrics. In Proceedings of the IEEE 18th International Conference on Software Engineering, Berlin, Germany, 25–30 March 1996; pp. 189–199. [CrossRef]
30. Blackburn, J.; Scudder, G.; Van Wassenhove, L. Improving Speed and Productivity of Software Development: A Global Survey of Software Developers. *IEEE Trans. Softw. Eng.* **1996**, *22*, 875–885. [CrossRef]
31. Delorey, D.P.; Knutson, C.D.; Chun, S. Do Programming Languages Affect Productivity? A Case Study Using Data from Open Source Projects. In Proceedings of the First International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS'07: ICSE Workshops 2007), Minneapolis, MN, USA, 20–26 May 2007; IEEE: Piscataway, NJ, USA, 2007; p. 8. [CrossRef]
32. Symons, C. Function Point Analysis: Difficulties and Improvements. *IEEE Trans. Softw. Eng.* **1988**, *14*, 2–11. [CrossRef]
33. Jiang, Q.; Lee, Y.C.; Davis, J.G.; Zomaya, A.Y. Diversity, Productivity, and Growth of Open Source Developer Communities. *arXiv* **2018**, arXiv:1809.03725.
34. Thurstone, L.L. The measurement of social attitudes. *J. Abnorm. Soc. Psychol.* **1931**, *26*, 249. [CrossRef]
35. Sarnoff, I. Psychoanalytic theory and social attitudes. *Public Opin. Q.* **1960**, *24*, 251–279. [CrossRef]
36. Greenwald, A.G. On Defining Attitude and Attitude Theory. In *Psychological Foundations of Attitudes*; Elsevier: Amsterdam, The Netherlands, 1968; pp. 361–388. [CrossRef]
37. Strzalkowski, T.; Harrison, T.; Sa, N.; Katsios, G.; Khoja, E. GitHub as a Social Network. In *Advances in Artificial Intelligence, Software and Systems Engineering*; Springer: Cham, Switzerland, 2018; pp. 379–390, ISSN 2194-5365. [CrossRef]
38. Ng, C.L.; Siew-Hoong, A.; Tong-Ming, L. A Study on the Element of Sentiment toward Knowledge Sharing among Knowledge Workers in a Virtual CoP. *Inf. Manag. Bus. Rev.* **2013**, *5*, 553–560. [CrossRef]
39. Dehbozorgi, N.; Maher, M.L.; Dorodchi, M. Emotion Mining from Speech in Collaborative Learning. *Adv. Sci. Technol. Eng. Syst. J.* **2021**, *6*, 90–100. [CrossRef]
40. Joshi, R.B. Reinforcement Learning for GitHub Pull Request Predictions: Analyzing Development Dynamics. Master's Thesis, Carleton University, Ottawa, ON, Canada, 2023. [CrossRef]
41. Guzman, E.; Azócar, D.; Li, Y. Sentiment Analysis of Commit Comments in GitHub: An Empirical Study. In Proceedings of the 11th Working Conference on Mining Software Repositories, Hyderabad, India, 31 May–1 June 2014; ACM: New York, NY, USA, 2014; pp. 352–355. [CrossRef]
42. Asri, I.E.; Kerzazi, N.; Uddin, G.; Khomh, F.; Janati Idrissi, M. An Empirical Study of Sentiments in Code Reviews. *Inf. Softw. Technol.* **2019**, *114*, 37–54. [CrossRef]

43. Huq, S.F.; Sadiq, A.Z.; Sakib, K. Is Developer Sentiment Related to Software Bugs: An Exploratory Study on GitHub Commits. In Proceedings of the 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER), London, ON, Canada, 18–21 February 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 527–531. [CrossRef]
44. Li, L.; Cao, J.; Lo, D. Sentiment Analysis over Collaborative Relationships in Open Source Software Projects. In Proceedings of the International Conference on Software Engineering and Knowledge Engineering, Pittsburgh, PA, USA, 9–19 July 2020.
45. Steinmacher, I.; Conte, T.; Gerosa, M.A.; Redmiles, D. Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects. In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, Vancouver, BC, Canada, 14–18 March 2015; ACM: New York, NY, USA, 2015; pp. 1379–1392. [CrossRef]
46. Licorish, S.A.; MacDonell, S.G. Exploring the Links between Software Development Task Type, Team Attitudes and Task Completion Performance: Insights from the Jazz Repository. *Inf. Softw. Technol.* **2018**, *97*, 10–25. [CrossRef]
47. Wagner, S.; Ruhe, M. A Systematic Review of Productivity Factors in Software Development. In Proceedings of the 2nd International Workshop on Software Productivity Analysis and Cost Estimation (SPACE 2008), Beijing, China, 9 July 2008.
48. Meyer, A.N.; Zimmermann, T.; Fritz, T. Characterizing Software Developers by Perceptions of Productivity. In Proceedings of the 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Toronto, ON, Canada, 9–10 November 2017; pp. 105–110. [CrossRef]
49. Murphy-Hill, E.; Jaspan, C.; Sadowski, C.; Shepherd, D.; Phillips, M.; Winter, C.; Knight, A.; Smith, E.; Jorde, M. What Predicts Software Developers' Productivity? *IEEE Trans. Softw. Eng.* **2021**, *47*, 582–594. [CrossRef]
50. Satratzemi, M.; Xinogalos, S.; Tsompanoudi, D.; Karamitopoulos, L. Examining Student Performance and Attitudes on Distributed Pair Programming. *Sci. Program* **2018**, 1–8. [CrossRef]
51. Anany, M.; Hussien, H.; Aly, S.; Sakr, N. Influence of Emotions on Software Developer Productivity. In Proceedings of the 9th International Conference on Pervasive and Embedded Computing and Communication Systems, Vienna, Austria, 19–20 September 2019; pp. 75–82. [CrossRef]
52. Wachs, J.; Nitecki, M.; Schueller, W.; Polleres, A. The geography of open source software: Evidence from github. *Technol. Forecast. Soc. Chang.* **2022**, *176*, 121478. [CrossRef]
53. Schreiber, R.R. Organizational influencers in open-source software projects. *Int. J. Open Source Softw. Process. (IJOSSP)* **2023**, *14*, 1–20. [CrossRef]
54. Rajanen, M.; Iivari, N. Examining usability work and culture in OSS. In Proceedings of the Open Source Systems: Adoption and Impact: 11th IFIP WG 2.13 International Conference, OSS 2015, Florence, Italy, 16–17 May 2015, Proceedings 11; Springer: Berlin/Heidelberg, Germany, 2015; pp. 58–67.
55. Qiu, H.S.; Li, Y.L.; Padala, S.; Sarma, A.; Vasilescu, B. The signals that potential contributors look for when choosing open-source projects. *Proc. ACM Hum.-Comput. Interact.* **2019**, *3*, 1–29. [CrossRef]
56. Kalliamvakou, E.; Gousios, G.; Blincoe, K.; Singer, L.; German, D.M.; Damian, D. The Promises and Perils of Mining GitHub. In Proceedings of the 11th Working Conference on Mining Software Repositories, Hyderabad, India, 31 May–1 June 2014; ACM: New York, NY, USA, 2014; pp. 92–101. [CrossRef]
57. Kalliamvakou, E.; Gousios, G.; Blincoe, K.; Singer, L.; German, D.M.; Damian, D. An In-Depth Study of the Promises and Perils of Mining GitHub. *Empir. Softw. Eng.* **2016**, *21*, 2035–2071. [CrossRef]
58. Borges, H.; Hora, A.; Valente, M.T. Understanding the Factors That Impact the Popularity of GitHub Repositories. In Proceedings of the 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME), Raleigh, NC, USA, 2–7 October 2016; pp. 334–344. [CrossRef]
59. Jiang, J.; Lo, D.; He, J.; Xia, X.; Kochhar, P.S.; Zhang, L. Why and How Developers Fork What from Whom in GitHub. *Empir. Softw. Eng.* **2017**, *22*, 547–578. [CrossRef]
60. Jemerov, D.; Isakova, S. *Kotlin in Action*; Simon and Schuster: New York, NY, USA, 2017.
61. Reese, G. *Database Programming with JDBC and JAVA*; O'Reilly Media, Inc.: Newton, MA, USA, 2000.
62. Islam, M.R.; Zibran, M.F. SentiStrength-SE: Exploiting Domain Specificity for Improved Sentiment Analysis in Software Engineering Text. *J. Syst. Softw.* **2018**, *145*, 125–146. [CrossRef]
63. Thelwall, M.; Buckley, K.; Paltoglou, G. Sentiment Strength Detection for the Social Web. *J. Am. Soc. Inf. Sci. Technol.* **2012**, *63*, 163–173. [CrossRef]
64. Ortu, M.; Murgia, A.; Destefanis, G.; Tourani, P.; Tonelli, R.; Marchesi, M.; Adams, B. The emotional side of software developers in JIRA. In Proceedings of the 13th International Conference on Mining Software Repositories, Austin, TX, USA, 14–22 May 2016; pp. 480–483. [CrossRef]
65. Lin, B.; Zampetti, F.; Bavota, G.; Di Penta, M.; Lanza, M.; Oliveto, R. Sentiment analysis for software engineering: How far can we go? In Proceedings of the 40th International Conference on Software Engineering, Gothenburg Sweden, 27 May–3 June 2018; pp. 94–104. [CrossRef]

66. Islam, M.R.; Zibran, M.F. Leveraging Automated Sentiment Analysis in Software Engineering. In Proceedings of the 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), Buenos Aires, Argentina, 20–21 May 2017; pp. 203–214. [CrossRef]
67. Obaidi, M.; Nagel, L.; Specht, A.; Klünder, J. Sentiment Analysis Tools in Software Engineering: A Systematic Mapping Study. *Inf. Softw. Technol.* **2022**, *151*, 107018. [CrossRef]
68. Kruskal, W.H.; Wallis, W.A. Use of ranks in one-criterion variance analysis. *J. Am. Stat. Assoc.* **1952**, *47*, 583–621. [CrossRef]
69. Mann, H.B.; Whitney, D.R. On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.* **1947**, *18*, 50–60. [CrossRef]
70. Myers, J.; Well, A.; Lorch, R. *Research Design and Statistical Analysis*, 3rd ed.; Taylor & Francis: Oxfordshire, UK, 2013.
71. Kendall, M.G. A New Measure of Rank Correlation. *Biometrika* **1938**, *30*, 81–93. [CrossRef]
72. Python Software Foundation. Python 3.11.7 Documentation. Available online: https://docs.python.org/3.11/ (accessed on 30 May 2025).
73. McKinney, W. Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 28 June–3 July 2010; Walt, S.v.d., Millman, J., Eds.; 2010; pp. 56–61. [CrossRef]
74. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef]
75. Dunn, O.J. Multiple Comparisons Among Means. *J. Am. Stat. Assoc.* **1961**, *56*, 52–64. [CrossRef]
76. Dunnett, C.W. A Multiple Comparison Procedure for Comparing Several Treatments with a Control. *J. Am. Stat. Assoc.* **1955**, *50*, 1096–1121. [CrossRef]
77. Rea, L.M.; Parker, R.A. *Designing and Conducting Survey Research: A Comprehensive Guide*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
78. Vargha, A.; Delaney, H.D. A Critique and Improvement of the CL Common Language Effect Size Statistics of McGraw and Wong. *J. Educ. Behav. Stat.* **2000**, *25*, 101–132. [CrossRef]
79. Sterman, J. *System Dynamics: Systems Thinking and Modeling for a Complex World*; Working Paper; Accepted: 2016-06-01T12:52:00Z; Massachusetts Institute of Technology, Engineering Systems Division: Cambridge, MA, USA, 2002.
80. Pearce, J.M. The Case for Open Source Appropriate Technology. *Environ. Dev. Sustain.* **2012**, *14*, 425–431. [CrossRef]
81. Hoe, N.S. *Breaking Barriers: The Potential of Free and Open Source Software for Sustainable Human Development*; United Nations Development Programme: New York, NY, USA, 2007.
82. Weiss, H.M.; Cropanzano, R. Affective events theory. *Res. Organ. Behav.* **1996**, *18*, 1–74.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.