# Developers' Sentiment and Issue Reopening

Jonathan Cheruvelil
*Amazon Inc.*
Seattle, WA, USA
cheruvj@amazon.com

Bruno C. da Silva
*Comp Sci. and Software Eng. Department*
*California Polytechnic State University*
San Luis Obispo, CA, USA
bcdasilv@calpoly.edu

***Abstract*—Since software engineering is done primarily by humans, the sentiment of the developers is crucial when trying to achieve the best possible results. This study takes a look at a specific instance of this: developer sentiment in relation to issue reopening. In general, issue reopening is something to be avoided as it indicates that something went wrong in the original issue fixing, which means extra work will have to be done to fix it entirely. In this study, we were able to apply a sentiment analysis tool to issue tracking comments and observe how the scores varied across issues with no reopenings, one reopening, and many reopenings. We found evidence that suggests that negative sentiment correlates with issue reopening, although the effect size seems to be rather small.**

***Index Terms*—Sentiment Analysis, Issue Reopening**

## I. Introduction

Every step in the software engineering process – from deciding on architecture to fixing bugs – involves humans relying on and communicating with each other. As with anything else that involves human interaction, emotions can significantly affect the quality of the work [1].

Research has already been done on emotions and software development and has produced some interesting findings (as we mention in Section II. However, there are still some areas that desire more exploration as well as improvements that can be made regarding the effectiveness of the sentiment scoring techniques (many off-the-shelf sentiment analysis tools are not well-tailored for work in the software engineering domain [2]). For this project, we decided to explore the role that developer sentiment has on issue reopening as well as apply a recently proposed sentiment analysis tool that has been adapted to work better with software engineering texts.

Issue tracking is a cornerstone of the software engineering process. As code gets deployed, it is inevitable that parts of it may not work the way they are designed to. Tracking these issues in an orderly, procedural fashion guarantees that the software under development can maintain high standards, keeping both the developers and the clients satisfied. Reopening an issue can be problematic, however. It signals that an issue's original solution may not have been properly thought out or implemented [3]. It means that additional work – possible duplicated work – will have to be completed. It is something to avoid, and can even be frustrating for developers. Therefore, it can be valuable to explore how developers' sentiment relates to issue reopening, as it may lead to better practices and a smoother development process. This thought led us to form the following research questions:

**RQ1:** Are comments with negative sentiments more likely to appear in issues that have been reopened? We hypothesize that negative sentiment in issue comments may indicate that a problem regarding the issue may not be fully resolved or that the issue is complex enough that it is likely to be reopened in the future.

**RQ2:** Does a larger comment size correlate with more extreme sentiment scores? We hypothesize that more words in a comment lead to higher positive scores and lower negative scores since there will be more words available for analysis.

**RQ3:** Do different projects have different proportions in regards to sentiment scores and issue reopening status? We hypothesize that different projects will show varying proportions because projects can vary in complexity, size, team turnover, communication channels, and more. Also, projects are worked on by different developers with their own set of personalities and ways of utilizing issue tracking systems.

## II. Related Work

We do not intend to cover in this section the entire literature regarding the association between software engineering and developer sentiment. Rather, we cite here some of the work that influenced us. Islam and Zibran [4] developed a sentiment analysis tool that better suits the software engineering domain. The resulting tool, SentiStrengthSE, was utilized in this project to get sentiment scores from the issue comments. In a previous work [5], we completed a study that investigated how developer sentiment affects the status of Travis CI builds. We found evidence suggesting that negative sentiment affects the result of the build process, though the effect size is rather small most likely due to inaccuracies from the sentiment analysis tool. Destefanis et al [6] showed that politeness between developers has a positive correlation with a decreased time required to fix issues. Ortu et al [7] ran a similar study, observing the relationship between a variety of different sentiments with the time to fix issues and found that happier sentiments (JOY, LOVE, etc.) in comments led to a shorter fixing time while negative sentiments (SADNESS, etc.) led to a longer fixing time. We did not find any published study that applied SentiStrenghSE on issue comments in order to correlate with issue reopening, which is our main focus in this work.

## III. Research Settings: Tools and Repositories

To obtain a representation of the sentiment in issue comments, we used the tool SentiStrengthSE, which is an extension of the original SentiStrength. SentiStrength may not be the best tool to use in the software engineering domain, according to research that shows that its scores are rather inconsistent when parsing software-related inputs [4]. SentiStrengthSE builds off of the original tool by making it more usable for texts having to do with software engineering.

SentiStrengthSE assigns positive and negative scores to segments of texts. SentiStrengthSE works by checking the input for positive and negative words, which are supplied in a provided dictionary. The tool can assign two different scores to a word: either positive scores or negative scores. The positive scores can range from 5 (extremely positive) to 1 (not positive) and the negative scores can range from -5 (extremely negative) to -1 (not negative). Once the scores are assigned to the words, SentiStrengthSE takes the maximum positive score and the minimum negative score and assigns them as the overall scores for the text. For example, the text "Thanks for the work. The logic is great, but the code style is ugly" will get assigned the scores 2 [Thanks], 3 [great], and -3 [ugly], leading to an overall sentiment score represented by the tuple (3, -3).

We also used Jira, a well-known issue tracking software, wich provides functionality to manage the different stages of an issue (open, closed, being fixed, etc.). Moreover, Jira provides an API for Java that allows you to obtain and post information to their servers from within a Java project. Using their API, we were able to get the relevant information for different issues (including transition history) and run analysis on a large collection of issues. Jira was also used by the SentiStrengthSE authors to empirically validate improvements implemented over the original SentiStrength.

Also, we explored the issues from Apache projects. Apache is a non-profit that coordinates many popular open-source projects, and also uses Jira for its projects and has all their issues available to the public. Since their projects usually involve a variety of different developers and cover a wide spectrum of topics, we believed that using data from Apache's issue tracking system would be appropriate for our analysis.

## IV. Research Settings: Data Collection

To begin our investigation, we had to obtain a sample of issues from Jira and compute the scores for their comments. Jira has a large number of different projects, but we wanted to get data from the projects that had both a large number of committers and a large number of issues. The rationale behind this is that more committers and issues would supposedly lead to a larger variety of issue commenters, which would reduce the chance that the data would be swayed by the commenting style of only a handful of specific developers. Apache has some statistics[1] for projects hosted on Jira, so we selected eight projects that were on both rankings of highest number of committers and highest number of issues by the time we did this research (see Table I). For each of the eight projects, we collected about 3000 issues using the Jira Java API.

One specific factor that we took into consideration was the date that an individual issue had been created. We wanted the issue to be available long enough so that it would have the opportunity to be closed and perhaps even reopened again. We were unable to find any statistics as to how long an appropriate time would be, so we decided to have a simple cutoff of a year. Any issues that had been created less than a year ago were eliminated from the data set. Then, for each issue that we decided to include in our set, we would keep the following information from each: Issue name, date created, transition history (a string containing all of issue status changes), and all comments (the issue description was not considered a part of the comments). From this data, we also obtained the two sentiment scores (a tuple representing the most negative and most positive) and kept track of the number of words being analyzed. The results were then put into a csv file. Our scripts are available online so our procedures can be replicated[2].

## V. Results

To investigate the research questions, we created graphs that would show the relationship between the resulting sentiment scores and the reopening status of the issues. We started by taking a look all the data at once. The issues were divided up by their highest positive and lowest negative sentiment scores (each issue would have an entry both of its scores). Then, a bar graph was created for each category, with each bar showing the percentage of issues that had no reopenings, one reopening, or many reopenings. The graphs for the entire data set are illustrated on Figures 1 and 2. On the top of each bar, there is the number of issues classified with that specific score.

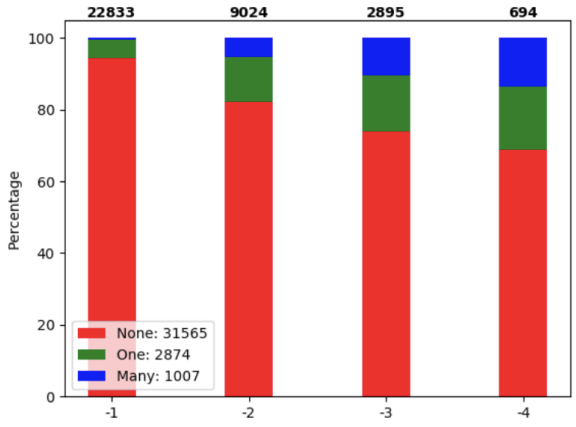


Fig. 1. Distribution of negative scores for all issues.

### A. RQ1: Are comments with negative sentiments more likely to appear in issues that have been reopened?

The graphs show that for more extreme scores (high positive and low negative scores), the proportion of issues that have

[1]Apache Projects Statistics, available at https://projects.apache.org/statistics.html

[2]https://github.com/bcdasilv/sentiment-analysis-on-issues

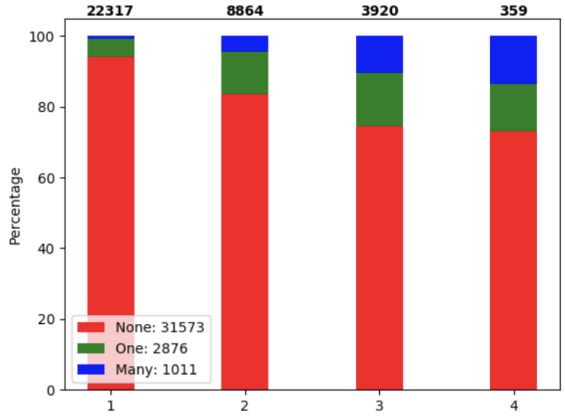


Fig. 2. Distribution of positive scores for all issues

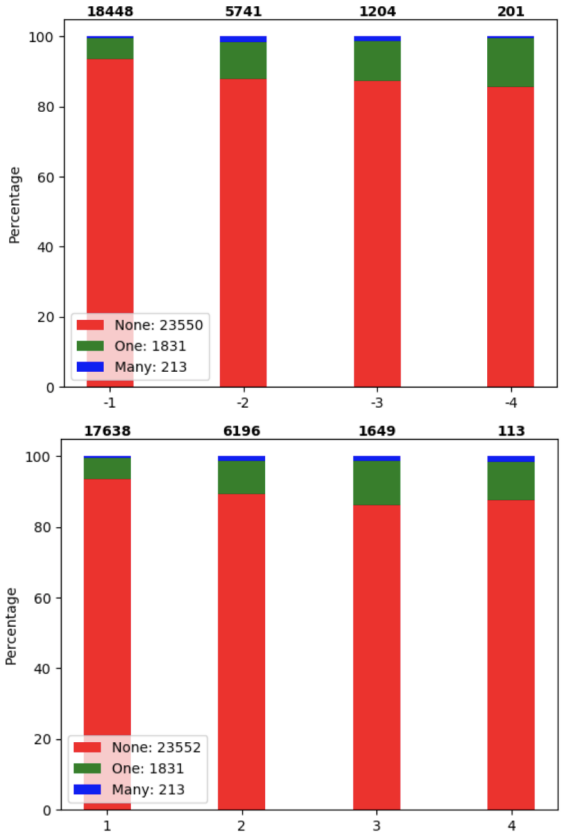


Fig. 3. Distribution of negative and positive scores for comments <500 words

been opened at least once is larger. To further investigate the association, we used the chi-squared test for independence to better understand the relationship between issue reopening status and sentiment. Running the test on all of the data points gave us a chi-squared value of 0.0, which indicates that the difference in proportions is statistically significant. Furthermore, we calculated the Cramer's V value for that data to understand the effect size. The value for the positive sentiments was 0.176, while the value for negative sentiments was 0.187. For this particular set of data, both of these values would be considered a small to medium effect size.

### *B. RQ2: Does a larger comment size correlate with more extreme sentiment scores?*

To investigate this, we made graphs similar to the first two, but the data set was divided up into three categories: comments with under 500 words, comments with between 500 and 1000 words, and comments with over 10000 words. The proportions were calculated in the same fashion, and the resulting graphs are displayed in Figures 3-5.

From those graphs, it seems apparent that as we get larger and larger comment sizes, we are expected to get more extreme sentiment scores on both ends (negative and positive). This is as we hypothesized, and the rationale behind it makes sense. As mentioned before, SentiStrengthSE works by obtaining the lowest negative score and the highest positive score in a given input. Therefore, if the input is longer, then there are more words available to be analyzed and a larger chance that well see a word or phrase that gets an extreme score.

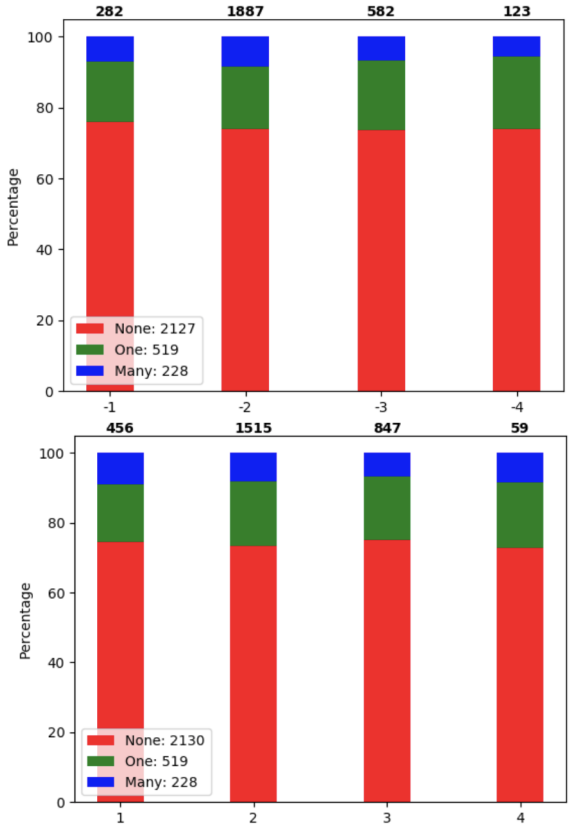


Fig. 4. Distribution of negative and positive scores for comments with between 500 and 1000 words

### *C. RQ3: Do different projects have different proportions in regards to sentiment scores and issue reopening status?*

To investigate RQ3, we created the same kind of graphs for each individual project. The same kind of statistical tests were run and the results have been summarized in the Table I. As shown in the table, the proportions vary from project to project.

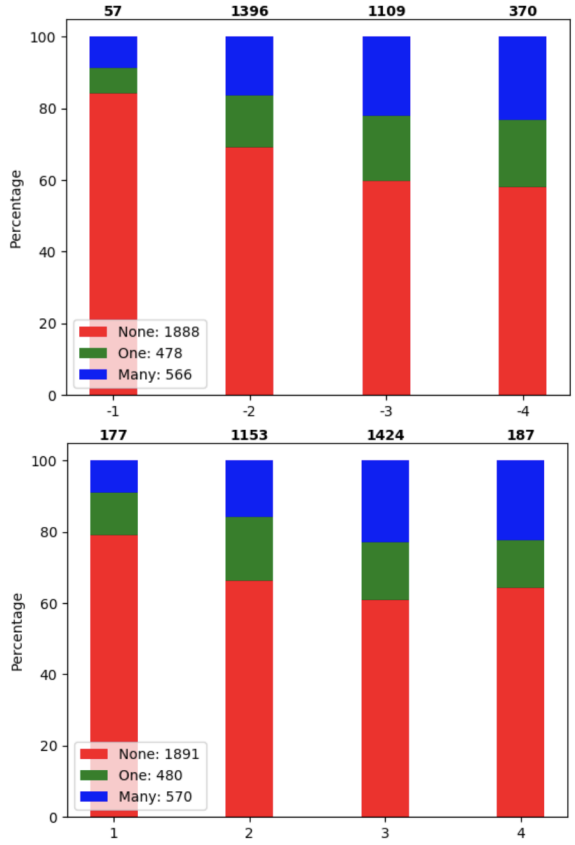


Fig. 5. Distribution of negative and positive scores for comments with over 1000 words

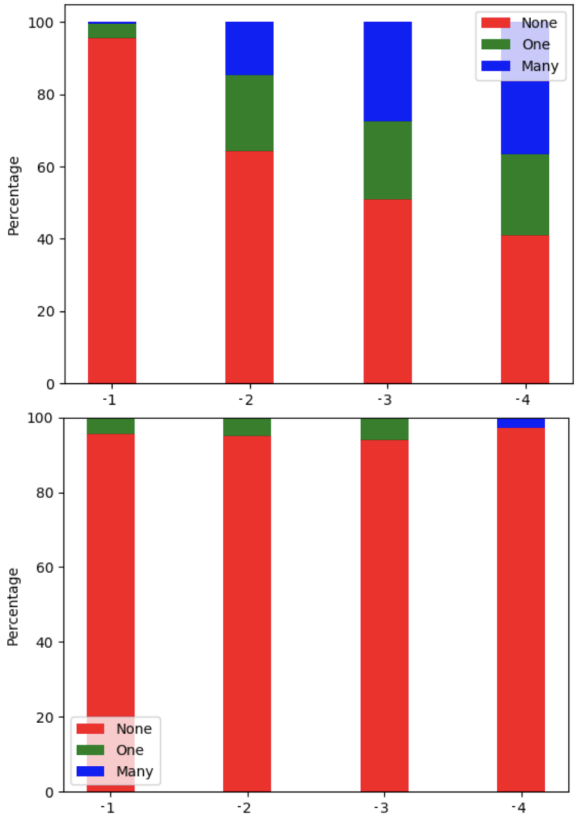


Fig. 6. The distribution of negative scores for ZOOKEEPER (top) and CLOUDSTACK (bottom)

In some, the difference in proportions by score is very slight (i.e. CLOUDSTACK, Figure 6 bottom), while others have a very large difference (i.e. ZOOKEEPER, Figure 6 top).

TABLE I
A SUMMARY OF THE STATISTICS FOR EACH OF THE PROJECTS

| Project Title | Percent of Issues Selected | Cramer's V Score (Negative) | Cramer's V Score (Positive) |
|---|---|---|---|
| ZOOKEEPER | 88.4% | .303 | .288 |
| MNG | 77.5% | .164 | .152 |
| CLOUDSTACK | 46.6% | .034 | .029 |
| FELIX | 94.0% | .108 | .122 |
| QPID | 60.6% | .112 | .109 |
| ZEPPELIN | 51.2% | .072 | .065 |
| GROOVY | 69.5% | .101 | .107 |
| HADOOP | 30.2% | .224 | .202 |

As made apparent by the table and the graphs, the differences between projects can be very dramatic. This should not come as a total surprise, as different developers use issue tracking systems in different ways. Some teams may choose to take in-depth discussions offline, while others may fully use the issue comment system. Some developers may use more emotional language while commenting than others. So much of these scores are based on human decisions, and the difference in humans is very apparent here.

## VI. Conclusion and Future Work

Throughout all of the data we analyzed, we found evidence that negative sentiment in issue comments has some correlation with issue reopening. Although there is evidence that there is a relationship, we have observed the effect size to be rather small. It is also worth noting that we did not find anything to suggest that the opposite effect could exist. More extreme scores almost always had an equal or higher proportion of reopened issues, even though the difference may have been small. We never observed more extreme scores having a lower proportion of reopened issues, suggesting that if a relationship exists, it exists in the way we expect it to.

Research on this topic can be furthered. We would recommend a larger scale study to be completed on a larger number of projects and issues, and comparing multiple tools. In addition, we have already started to apply IBM tools on developer text in order to analyze multiple facets of the developers sentiment and emotions such as agreeableness, openness, extraversion, excitement, among others.

## References

[1] T. DeMarco and T. Lister, *Peopleware: Productive Projects and Teams (3rd Edition)*, 3rd ed. Addison-Wesley Professional, 2013.

[2] R. Jongeling, P. Sarkar, S. Datta, and A. Serebrenik, "On negative results when using sentiment analysis tools for software engineering research," *Empirical Softw. Engg.*, vol. 22, no. 5, pp. 2543–2584, Oct. 2017.

[3] T. Zimmermann, N. Nagappan, P. J. Guo, and B. Murphy, "Characterizing and predicting which bugs get reopened," in *2012 34th International Conference on Software Engineering (ICSE)*, June 2012, pp. 1074–1083.
[4] M. R. Islam and M. F. Zibran, "Leveraging automated sentiment analysis in software engineering," in *Proceedings of the 14th International Conference on Mining Software Repositories*, ser. MSR '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 203–214.
[5] R. Souza and B. Silva, "Sentiment analysis of travis ci builds," in *Proceedings of the 14th International Conference on Mining Software Repositories*, ser. MSR '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 459–462.
[6] G. Destefanis, M. Ortu, S. Counsell, S. Swift, M. Marchesi, and R. Tonelli, "Software development: do good manners matter?" *PeerJ Computer Science*, vol. 2, p. e73, 2016.
[7] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli, "Are bullies more productive?: Empirical study of affectiveness vs. issue fixing time," in *Proceedings of the 12th Working Conference on Mining Software Repositories*, ser. MSR '15. Piscataway, NJ, USA: IEEE Press, 2015, pp. 303–313.