Is Developer Sentiment Related to Software Bugs: An Exploratory Study on GitHub Commits

Syed Fatiul Huq, Ali Zafar Sadiq, Kazi Sakib
Institute of Information Technology
University of Dhaka
Dhaka, Bangladesh
bsse0732@iit.du.ac.bd, zafarsadiq120@gmail.com, sakib@iit.du.ac.bd

***Abstract*—The outcome of software products primarily depends on the developers, including their emotion or sentiment in a software development environment. Developer emotions have been observed to be correlated to several patterns, for instance, task resolution time, developer turnover, etc. by conducting sentiment analysis on software collaborative artifacts like Commits. This study aims to quantify the impact of those patterns by finding a relation between developer sentiment and software bugs. To do so, Fix-Inducing Changes — changes that introduce bugs to the system — are detected, along with changes that precede or fix those bugs. Sentiment of these changes are determined from their Commit messages using Senti4SD. It is statistically observed that Commits that introduce, precede or fix bugs are significantly more negative than regular Commits, with a higher proportion of emotional (non-neutral) messages. It is also found that a distinction between buggy and correct fixes exists based on the message's neutrality.**

***Index Terms*—Human Factors in Software Engineering, Sentiment Analysis, Fix Inducing Change**

I. Introduction

Sentiment analysis – the extraction of human emotions from written text [1] – is an important approach for understanding the behavioural pattern of software developers and its effect on software systems. Software developers post their textual input in online collaborative artifacts, e.g., Commits, Issue reports, discussions etc. Commits store historical data on the code changed by developers, along with their personal message. This message can be used to extract the developer's sentiment regarding the Commit's task. Commits, therefore, work as an important tool in establishing a direct relation between developer performance and emotion.

Developer performance can be derived from software Commits based on the type of code written in that Commit. For instance, buggy code can be extracted from Commits by finding Fix-Inducing Changes (FIC). FICs are Commits that induce fixes i.e., cause errors in the system [2]. Messages in FICs can contain different emotional expression than other Commits, to indicate complex assignments or imperfect work. Commits related to FICs, for instance, ones that fix or precede bugs, can similarly provide insight based on the messages' emotion. Therefore, this study observes the emotional patterns of four types of Commits to understand the relation between developer emotion and software bugs. With this aim, the following four Research Questions (RQs) are answered:

RQ1: How does developer sentiment relate to Fix-inducing Changes? FICs are Commits where bugs originate. FICs have been analyzed to correlate the introduction of bugs with other project metrics like day of Commit [2], code smell refactoring [3], change coupling [4] etc. Similarly, a link between FICs and the sentiment of their message is a direct correlation between bugs and emotion.

RQ2: How does developer sentiment relate to the parent of Fix-inducing Changes? The parent of a Commit is the Commit that immediately precedes it. The parent contains the latest code on which the current Commit was written and a message to describe the context of that code. The parent of an FIC (pFIC) is important for sentiment analysis as the coder of FIC refers to its message and can be influenced by it.

RQ3: How does developer sentiment relate to Fixing Changes? Fixing Changes (FCs) are the Commits that fix bugs introduced by FICs. These Commits are designated tasks for removing bugs. Sentiment in their messages can provide insight into the developer response to such tasks.

RQ4: How does developer sentiment relate to Fix-inducing Fixes? When FCs themselves create new bugs, those Commits are regarded as Fix-inducing Fixes (FIFs) [5]. It is observed whether FIFs can be differentiated from FCs using the emotion of their messages.

Previous studies on sentiment analysis in software engineering have related sentiment with different patterns and project properties. GitHub Commits have been mined [6] [7] to observe days with negative Commits, and how change size and personnel diversity can affect sentiment. Comments in GitHub [8] and Jira [9] [10] Issues have been analyzed to relate developer sentiment with Issue resolution activities. Others studied various effects of sentiment like performance degradation [11] and contributor turnover [12]. However, these studies do not relate the emotional patterns with software bugs, which would provide a quantifiable impact of those patterns.

In order to correlate sentiment with bug-related Commits, this study first collects Commits of GitHub repositories and categorizes these based on the RQs into four different types: FIC, pFIC, FC and FIF. Sentiment values for the Commit messages are extracted using Senti4SD [13], a sentiment polarity classifier specialized for the software engineering domain. Lastly, statistical analysis is conducted on the resulting sets of sentiment values with that of regular Commits.

978-1-7281-5143-4/20/$31.00 © 2020 IEEE

The findings of the statistical test show a significant relation between bug-related changes and sentiment. All four Commit types are, on average, more negative than regular Commits. Additionally, neutrality is observed to be more predominant (6%) in regular Commits. In terms of polarity, pFIC, FC and FIFs have 12% more negative Commits than positive. Lastly, FCs that have more negative messages tend to become FIFs.

II. Related Work

With textual data exponentially increasing through online mediums, new and useful information are being derived by analyzing these data. Among different text analysis techniques, sentiment analysis is used for understanding trends in social media [14] and customer reception of business products [15]. Sentiment has also been studied as an important factor in organizations, finding relationship between employee emotion and their performance [16]. This indicates the effect of emotions in the workplace, which includes software development.

To understand the effect of sentiment in software development, different collaborative artifacts like Commit messages, bug reports etc. have been analyzed. Guzman et al. [11] have shown that real life behavior of developers is positively correlated with sentiments in the mentioned artifacts. Garcia et al. [12] showed that developer turnover causes negative sentiment in the project. Issues in open source projects have shown to be rich with sentiment [8], which has been correlated with Issue resolution activities [9] [10].

Commits have also been analyzed, as these are an integral aspect of modern software development processes. Sentiment in Commit messages has been observed to be related with various properties of the developers and project [6] [7]. These properties include time and day of Commit, number of changed files, project language, geographical distribution of the team and project rating. The obtained relation indicate that emotions of developer vary depending on environmental and project characteristics. For better understanding of the emotional impact, these should be correlated to performance metric of a developer.

An important metric for analyzing developer performance is Fix-Inducing Changes (FIC). FICs are Commits that introduce a bug in the software and these are later on fixed [2]. Various researches have analyzed FICs to derive useful coding information, for example, how fixes themselves created new bugs [17] and its relation with code coupling [4]. Other than analyzing programmatic cause and effects, introduction of FICs in projects can also be analyzed for behavioral aspects of developers. No study in Software Engineering was found to link sentiments with FIC as performance metric.

III. Methodology

This study aims to incorporate sentiment analysis with the introduction and patching of bugs. To do so, Commits are extracted from open source repositories and analyzed to detect the four intended types. Sentiment analysis on the Commit messages are then conducted to extract developer emotion. The processes are described in the following subsections.

A. Commit Categorization

The four Research Questions (RQs) involve four types of Commits: Fix-Inducing Changes (FIC), Parents of FICs (pFIC), Fixing Changes (FC) and Fix-Inducing Fixes (FIF). This categorization is based on the changes' involvement with bugs. FICs contain code that is or causes causes bugs. This buggy code is posted right after pFIC, the closest parent to FIC and the code on which the bug was written. FICs also prompt FCs, code that removes the buggy parts introduced by FICs, as a designated task to patch a reported bug. Lastly, changes, that are assigned with removing a bug but in doing so creates their own, lie in the category of FIFs.

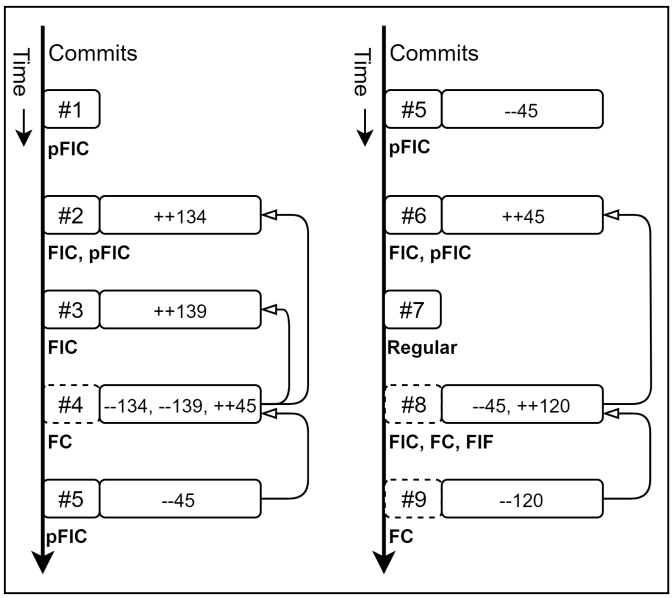

Fig. 1. Detecting and Categorizing Commits

To automatically detect and categorize all four Commits accordingly, the following steps are applied.

1) Remote repositories are locally cloned and their Commit lists are extracted. Only Commits from the default branch are collected since the contributions from different developers are reviewed and merged here.
2) Commit messages are analyzed to understand the purpose or contribution of the Commit. The message text is lemmatized and terms like "bug", "fix" and "patch" are searched. The presence of these words indicate that the Commit is related to a bug fixing activity [18].
3) Next the modifications in these Commits are analyzed to observe the exact changes. Each line in this Commit that is changed from its parent Commit is listed. It is checked whether only non-code changes like documentations and comments occur. Since these cannot introduce bugs, no bugs can be removed by modifying these. The type of change is also analyzed, of which there are three: Insert, Delete and Replace. Only Delete and Replace types are considered because these indicate the removal of buggy code, proving the existence of a previous Commit

where that was introduced. After these filtering steps, the remaining Commits are categorized as **FCs**.

4) For all the modified lines in the Fixing Changes, excluding non-code and Insert changes, their origin Commits are tracked. These Commits signify the introduction of the buggy codes that FCs patch. The Commits found from this step are regarded as **FICs**.
5) For each FIC, its immediate parent Commit is extracted and labeled as **pFIC**.
6) Commits that are labeled as both FC and FIC are categorized as **FIFs**.

Application of this process is displayed in Fig. 1 where a series of Commits on a single file is analyzed. The dashed Commits – #4, #8 and #9 – are FCs, detected from Step 3. #4 modifies lines (134, 139) introduced in #2 and #3, hence these two Commits are labeled as FICs. Their parents, #1 and #2 respectively, are pFICs. Now, although #5 removes a code (45) of #4, since #5 is not an FC, #4 is not labeled as an FIC. Next, the FC #8 modifies line 45, which is introduced in both #4 and #6. However, only the latest update is considered, hence #6 is the FIC. Lastly, #8 also adds code that is removed by another FC #9, making #8 an FIF.

B. Quantifying Sentiment

After categorizing the Commits, their sentiment values are quantified. The sentiment extracted from the messages provides two insights: subjectivity (neutral vs emotional) and polarity (negative vs positive). This study deals with both these aspects along with the sentiment's numeric value. When quantifying sentiment, two factors affect its correctness: the context of the words and their intrinsic emotional value. To understand the context, the meaning of the text and its parts needs to be approximated. Hence the text is not treated as a bag of words, rather the inter-relation of the words, clauses and sentences are interpreted. Furthermore, while for most words, their inherent sentiment value is universal, some words can have different meaning based on the domain it is used in. In the software engineering domain, words and terms like "bug", "errors" etc do not have the traditional negative connotations. These exceptions must be handled to extract the correct sentiment values of the Commit messages.

In this study, sentiment values for the Commit messages are extracted using the Senti4SD tool [13]. It is an emotion polarity classifier specializing in the software engineering domain. The tool uses lexicon-based and keyword-based features along with semantics to better contextualize the words. Additionally, it has been trained on a gold standard of 4423 StackOverflow posts, which incorporates software engineering domain-specific terminologies. The tool has been observed to perform best among similar sentiment analysis tools when conducted on collaborative artifacts in GitHub [19] [20].

Senti4SD takes in texts as input and, after analysis, assigns a sentiment polarity label – negative, neutral or positive – to the text. Before this analysis, each Commit message is parsed to remove code component and markdown text, and compressed into a single line. After generating the result, the polarity labels are transformed to respective numeric representation – -1, 0 and +1. The numeric transformation helps in statistically analyzing the data. Table I shows sentiment values after applying this process. The Commit messages are extracted from Google's Guava project. The table shows the existence of the three sentiment values in Commit messages.

TABLE I
EXAMPLES OF SENTIMENT VALUES OF COMMIT MESSAGES FROM THE GUAVA PROJECT

Sha	Commit message	Senti4SD result	Value
bbab2ce	ARGHGH, guess I was in the wrong directory when submitting... amateurs...	Negative	-1
f5ad01f	Some fixes to java5-compatible compilation	Neutral	0
2346903	Almost got it right, but luckily Colin was here to help me.	Positive	1

IV. THE EXPERIMENT

The methodology described is conducted on thirteen different repositories from GitHub. The repositories, relevant data and the process of statistically analyzing these data are described in the following subsections.

A. Dataset Description

Thirteen GitHub repositories are chosen based on popularity[1] and inclusion in the GHTorrent dataset[2]. As seen in Table II, the repositories are all mature with an average 9.7 years of project life and 148 releases. Furthermore, with an average of 170 contributors, the projects contain the collaboration necessary to analyze inter-developer communication. Due to being open source, the projects enforce communication via Commit messages as their teams are geographically dispersed.

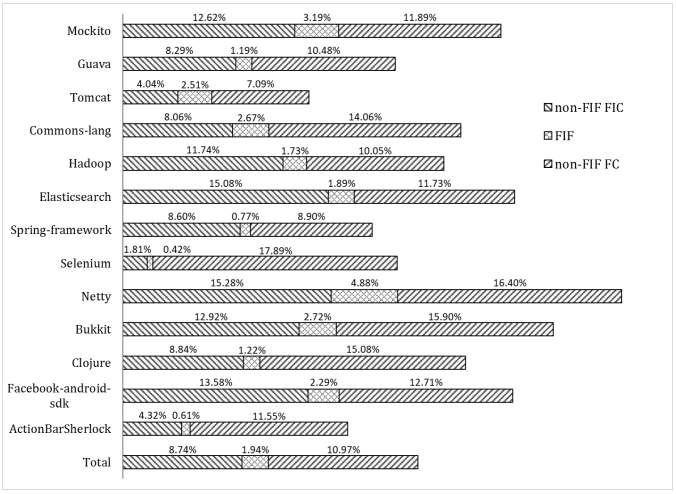

Fig. 2. Ratio of FIC, FIF and FCs in Projects

[1]https://medium.com/issuehunt/top-11-popular-java-projects-on-github-48aaad5b4e0a

[2]http://ghtorrent.org/msr14.html

TABLE II
REPOSITORY DESCRIPTION

Project Name	https://github.com/	Lifetime (Years)	Releases	Contributors	Commit Number
Guava	google/guava	9	88	214	5134
Mockito	mockito/mockito	12	470	170	5190
Tomcat	apache/tomcat	5	210	31	21570
Commons-lang	apache/commons-lang	16	87	127	5594
Hadoop	apache/hadoop	5	326	239	23271
Elasticsearch	elastic/elasticsearch	9	262	1312	49416
Spring-Framework	spring-projects/spring-framework	16	165	416	20017
Selenium	SeleniumHQ/selenium	8	119	466	24548
Netty	netty/netty	10	210	414	9630
Bukkit	Bukkit/Bukkit	9	67	106	1509
Clojure	clojure/clojure	11	148	144	3290
Facebook-Android-Sdk	facebook/facebook-android-sdk	7	106	76	1322
Actionbarsherlock	JakeWharton/ActionBarSherlock	9	32	53	1480
Average		9.7	148	170	13228.54

Extracting the Commits from the thirteen repositories yields the primary dataset. Categorizing the data according to the *Commit Categorization* section produces four sets of Commits. As shown in Fig. 2, 10% of the Commits on average are FICs, Commits that introduced bugs. 12% Commits are changes that are related to fixing those bugs. The intersection of these two sets are the FIFs: an average of 2% Commits.

B. Statistical Analysis

Wilcoxon rank sum test is conducted on the resulting sentiment data. The test determines whether the difference of means between two ordinal or interval non-parametric distributions is significant. It generates a p-value which, if < 0.05, rejects the null hypothesis: the means are not significantly different. In Table III, the reported p-values are modified with Bonferroni corrections to eliminate family-wise error rate.

Additionally, to understand whether the proportion of different sentiment has any relation with the four types of Commits, a Chi-square independence test is conducted. The test shows whether observed frequencies of the two categories of variables have a significant association. In Table IV, the italicized values represent significance based on this test.

V. RESULT ANALYSIS

The individual sentiment data based on the Commit categories are statistically analyzed to infer relation. Findings of the four Research Questions (RQs) are described as follows.

RQ1: How does developer sentiment relate to Fix-inducing Changes?

Observation: The first RQ observes the relation between developer sentiment and Fix-inducing Changes (FICs). From Table III it can be seen that, when all emotions are considered, there is a significant difference of sentiment values between FICs and regular Commits. Based on the statistical evidence, it can be said that FICs are more negative than other Commits.

Furthermore. from Table IV, it can be observed that both FICs and regular Commits have similar ratios of negative and positive emotions. Here negative sentiment outpopulates positive ones by at least 40%. However, FICs have more emotional messages than regular ones. FICs contain 6% more non-neutral messages.

TABLE III
COMMIT RELATION WITH SENTIMENT

		All Emotion		Polar Emotion	
RQ	Commit Type	Mean	p-value	Mean	p-value
1	FIC	-0.085	$4.72e^{-16}$	-0.403	0.99
	Regular	-0.064		-0.402	
2	pFIC	-0.082	$3.27e^{-12}$	-0.436	$4.26e^{-07}$
	Regular	-0.063		-0.394	
3	FC	-0.119	$< 8.8e^{-16}$	-0.595	$< 8.8e^{-16}$
	Regular	-0.058		-0.366	
4	FIF	-0.016	$< 8.8e^{-16}$	-0.580	$< 8.8e^{-16}$
	Regular	-0.076		-0.397	
5	FIF	-0.137	$2.08e^{-05}$	-0.576	0.8565
	non-FIF FC	-0.106		-0.581	

TABLE IV
SENTIMENT PROPORTIONS

		Polarity		Subjectivity	
RQ	Commit Type	Negative	Positive	Emotion	Neutral
1	FIC	70.13%	29.87%	*21.01%*	*78.99%*
	Regular	70.12%	29.88%	*15.82%*	*84.18%*
2	pFIC	*71.78%*	*28.22%*	*18.92%*	*81.08%*
	Regular	*69.72%*	*30.28%*	*16.00%*	*84.00%*
3	FC	*79.77%*	*20.23%*	*20.03%*	*79.97%*
	Regular	*68.31%*	*31.69%*	*15.84%*	*84.16%*
4	FIF	*78.98%*	*21.02%*	*2.84%*	*97.16%*
	Regular	*69.86%*	*30.14%*	*1.76%*	*98.24%*
5	FIF	78.79%	21.21%	*23.83%*	*76.17%*
	non-FIF FC	79.07%	20.93%	*18.29%*	*81.71%*

Inference: Commits that introduce bugs are more negative and contain less neutral message than regular Commits. This information can be used for precautionary measures with sensitive changes. For instance, Commits with negative messages can be given better emphasis during reviews. More priority should be given on monitoring and maintaining the sentiment of developers, as that relates to problematic performance.

RQ2: How does developer sentiment relate to the parent of Fix-inducing Changes?

Observation: From Table III, it is observed that, parent of FICs are more negative compared to regular Commits, when considering both polar and neutral emotions. Table IV shows that pFICs contain 2% more negative messages than regular Commits, while negative sentiment outnumbers positive ones

by at least 39%. It is also observed that pFICs have 3% more emotional messages than regular Commits.

Inference: Commits prior to introducing bugs are found to be more negative and less neutral compared to regular Commits. This information depicts the working situation prior to an error, as well as the state of the message left by the previous developer. Neutral messages should be posted so that it does not influence the next developer to make a mistake.

RQ3: How does developer sentiment relate to Fixing Changes?

Observation: Results from Table III shows that considering all and polar emotions, FCs and regular Commits have a significant difference of sentiment values. Statistically, FCs are more negative than regular Commits. Furthermore. from Table IV, it can be observed that FCs have more negative (by 10%) and emotional (by 5%) than other Commits.

Inference: Commits that fix bugs are more negative and less neutral than regular Commits. This represents the emotional state of developers when tasked with solving bugs, proving the affect of complex assignments on developer emotion.

RQ4: How does developer sentiment relate to Fix-inducing Fixes?

Observation: This RQ is observed in two ways. First, FIFs are compared with regular Commits, showing a statistical significance in their negativity, as seen in Table III. Table IV shows that there are 9% more negative messages in FIFs than in regular Commits.

In the second observation, comparing incorrect (FIF) and correct fixes (non-FIF FC), Commits that correctly fix bugs are significantly less negative, with 5% more neutral messages.

Inference: Incorrect bug fixes can be differentiated from regular Commits and correct fixes using developer sentiment. The finding indicates that developers assigned with bug fixes will mistakenly add a new bug with a negative emotional state. Additionally, FCs that show emotional messages should be revised for possible bugs.

These findings provide helpful insight into the developers' emotional patterns regarding software bugs and how sentiment can be used to anticipate bugs being introduced.

VI. Conclusion

This study finds the relation between developer sentiment and software bugs. Four types of Commits: Fix-Inducing Changes (FIC), parent of FICs (pFIC), Fixing Changes (FC) and Fix-inducing Fixes (FIF) are categorized and their messages are analyzed. Sentiment polarity values of the messages are extracted using the classifier Senti4SD. The Commits are observed to contain 6% less neutrality than regular Commits. Additionally, There are more negative messages in these Commits than positive. On average, bug related Commits are significantly more negative than regular Commits.

This result differentiates Commits that are related to bugs from regular ones based on developer sentiment. The finding strengthens the patterns observed in previous studies and prompts further investigation into the quantifiable impact of emotions on software systems.

Acknowledgment

This study is supported by the Fellowship from ICT Division, Government of Bangladesh – No.: 56.00.0000.028.33.093.19-427, date: 20.11.2019 – and Bangladesh Research and Education Network (BdREN).

References

[1] B. Pang, L. Lee, *et al.*, "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[2] J. Śliwerski, T. Zimmermann, and A. Zeller, "When do changes induce fixes?," in *ACM sigsoft software engineering notes*, vol. 30, pp. 1–5, ACM, 2005.

[3] G. Bavota, B. De Carluccio, A. De Lucia, M. Di Penta, R. Oliveto, and O. Strollo, "When does a refactoring induce bugs? an empirical study," in *2012 IEEE 12th International Working Conference on Source Code Analysis and Manipulation*, pp. 104–113, IEEE, 2012.

[4] A. Z. Sadiq, M. J. I. Mostafa, and K. Sakib, "On the evolutionary relationship between change coupling and fix-inducing changes," 2019.

[5] Z. Yin, D. Yuan, Y. Zhou, S. Pasupathy, and L. Bairavasundaram, "How do fixes become bugs?," in *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, pp. 26–36, ACM, 2011.

[6] E. Guzman, D. Azócar, and Y. Li, "Sentiment analysis of commit comments in github: an empirical study," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, pp. 352–355, ACM, 2014.

[7] V. Sinha, A. Lazar, and B. Sharif, "Analyzing developer sentiment in commit logs," in *Proceedings of the 13th International Conference on Mining Software Repositories*, pp. 520–523, ACM, 2016.

[8] F. Jurado and P. Rodriguez, "Sentiment analysis in monitoring software development processes: An exploratory case study on github's project issues," *Journal of Systems and Software*, vol. 104, pp. 82–89, 2015.

[9] 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*, pp. 303–313, IEEE Press, 2015.

[10] M. Mäntylä, B. Adams, G. Destefanis, D. Graziotin, and M. Ortu, "Mining valence, arousal, and dominance: possibilities for detecting burnout and productivity?," in *Proceedings of the 13th International Conference on Mining Software Repositories*, pp. 247–258, ACM, 2016.

[11] E. Guzman and B. Bruegge, "Towards emotional awareness in software development teams," in *Proceedings of the 2013 9th joint meeting on foundations of software engineering*, pp. 671–674, ACM, 2013.

[12] D. Garcia, M. S. Zanetti, and F. Schweitzer, "The role of emotions in contributors activity: A case study on the gentoo community," in *2013 International Conference on Cloud and Green Computing*, pp. 410–417, IEEE, 2013.

[13] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, "Sentiment polarity detection for software development," *Empirical Software Engineering*, vol. 23, no. 3, pp. 1352–1382, 2018.

[14] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pp. 30–38, 2011.

[15] H. Cui, V. Mittal, and M. Datar, "Comparative experiments on sentiment classification for online product reviews," in *AAAI*, vol. 6, p. 30, 2006.

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

[17] H. Yang, C. Wang, Q. Shi, Y. Feng, and Z. Chen, "Bug inducing analysis to prevent fault prone bug fixes.," in *SEKE*, pp. 620–625, 2014.

[18] S. Kim, E. J. Whitehead Jr, and Y. Zhang, "Classifying software changes: Clean or buggy?," *IEEE Transactions on Software Engineering*, vol. 34, no. 2, pp. 181–196, 2008.

[19] N. Novielli, D. Girardi, and F. Lanubile, "A benchmark study on sentiment analysis for software engineering research," in *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*, pp. 364–375, IEEE, 2018.

[20] I. El Asri, N. Kerzazi, G. Uddin, F. Khomh, and M. J. Idrissi, "An empirical study of sentiments in code reviews," *Information and Software Technology*, 2019.