

A Longitudinal Study on the Maintainers’ Sentiment of a Large Scale Open Source Ecosystem

Isabella Ferreira
Polytechnique Montréal
isabella.ferreira@polymtl.ca

Kate Stewart
Linux Foundation
kstewart@linuxfoundation.org

Daniel German
University of Victoria
dmg@uvic.ca

Bram Adams
Polytechnique Montréal
bram.adams@polymtl.ca

Abstract—Software development is a collaborative activity in which feelings and emotions can affect the developer’s productivity, creativity, and contribution satisfaction. For example, the Linux Kernel Mailing List (LKML), which is used by subsystem maintainers to review patches sent by contributors, is known for its direct communication style, which is sometimes blamed as having a negative impact on contributors. In September 28, 2018, the kernel’s lead maintainer, Linus Torvalds, announced that he would take a temporary break from the community, which led numerous members of the kernel community and observers from other communities to wonder to what extent this unexpected event could raise awareness about respectful interactions between community members. This paper performs an exploratory study in which we use an off-the-shelf sentiment mining tool to assess whether the maintainers’ sentiment changed after Linus Torvalds’ temporary break from his maintainer role. Based on the data available thus far, we did not find any high-level changes in maintainer sentiment. In future work, we will perform more fine-grained sentiment analysis.

I. INTRODUCTION

Software development is a collaborative activity where developers use mailing lists, software code repositories, and issue tracking tools to manage their work [1]. Thus, like any collaborative activity, software development involves sentiment or emotions that can affect the developer’s productivity, creativity, and contribution satisfaction [1]. The discussions can get excited as different people from distinct cultures and interests are part of it. In this sense, the risk of offensive behaviors increases more and more [2]. As a consequence, modern open source communities establish a code of conduct, which are a selection of ground rules for communication between participants, with the goal to make everyone comfortable in contributing to the open source project [2].

In the Linux kernel, all development activities are organized across a large number of mailing lists centered around the Linux Kernel Mailing List (LKML). This mailing list acts as a central place for discussions about technical and organizational aspects of the kernel development [3]. Contributors submit patches to subsystem maintainers, who approve the submissions and initiate the process of integrating the patch into an upcoming release [4]. Linus Torvalds, the founder of the Linux operating system, has the final decision as a project maintainer.

On September 16, 2018, Linus announced in LKML that he would temporarily take a break from Linux maintainership: *“My flippant attacks in emails have been both unprofessional*

and uncalled for. Especially at times when I made it personal. In my quest for a better patch, this made sense to me. I know now this was not OK and I am truly sorry. I am going to take time off and get some assistance on how to understand people’s emotions and respond appropriately” [5]. After this, many members of the kernel community as well as observers from other communities are wondering to what extent this unexpected event could raise awareness about respectful interactions between community members. The topics of politeness, positive interactions, and openness to opinions have become more openly discussed.

Thus, this paper aims to study whether this debate has changed the way in which kernel maintainers interact with developers and with each other. For this, we empirically study the sentiment in two years of kernel review discussions. In particular, we analyzed 15,166 e-mail threads that contain at least one subsystem maintainer from January 2017 to January 2019 to answer the following research questions:

RQ1. How does the e-mail thread sentiment evolve in the kernel review discussions?

We could not find any significant difference in the sentiment across releases, months, and weeks. However, there is a slight difference in sentiment when comparing weekdays. Sunday, Tuesday, Wednesday, and Thursday are the days with more positive e-mail threads compared to negative e-mail threads.

RQ2. Did the maintainers’ sentiment change after Linus Torvalds’ temporary break from the community?

There is no significant difference in the sentiment before and after Linus took a temporary break when comparing releases. However, there is a slight difference when comparing months, and weeks. The sentiment is more positive around August and September, and in November it becomes more negative.

II. THE LINUX KERNEL REVIEW PROCESS

The Linux kernel still uses low-tech reviewing environments based on mailing list discussions [6], [7]. The patch submission process for the kernel is managed by Linus Torvalds, who makes the final decision, yet is guided by an army of “maintainers” in charge of vetting patches to one or more kernel subsystems. Figure 1 presents the life cycle of a typical kernel patch.

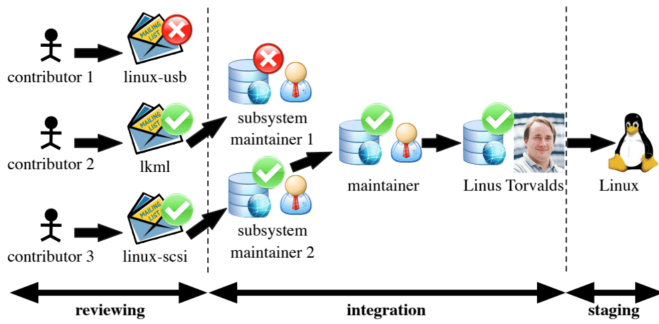


Fig. 1. Linux kernel development process (Extracted from [8]).

First, a developer sends a Request for Comments (RFC) to a kernel subsystem’s mailing list to get ideas for new features or bug fixes. Based on the received feedback, the developer implements and submits the resulting patch to the Linux Kernel Mailing List (LKML) or to any other subsystem’s mailing list. Afterwards, the subsystem maintainer reviews the patch, and either rejects the patch, requests a new version with modifications, or approves it. When the final version of the patch is approved, the subsystem maintainer commits the final patch to her Git repository. These patches then propagate to the repository to higher-level maintainers until (hopefully) it manages to reach the repository of Linus Torvalds. If he pulls in a patch, it becomes part of an official Linux kernel release. In this paper, we analyze the sentiment on the maintainers’ discussions made during the review process.

III. METHODOLOGY

This section presents the methodology used to answer our research questions.

A. Extracting the mailing list data

In order to quantitatively assess the sentiment polarity of the discussions made on LKML, we mined the Linux Kernel Mailing List¹. Since the mailing list is stored in different git repositories², we used git commands to extract its content in each repository. In this study, we only consider e-mails from January 2017 to January 2019. Since the discussion to review a patch is spread across a multitude of e-mail replies, we then group individual e-mails by e-mail threads. For that, we used the Mailboxminer tool [9].

After grouping individual e-mails by e-mail threads, we filtered the threads by retaining those containing at least two e-mails in their discussion, excluding e-mails that have patches. In other words, we only analyze threads with e-mails making up the actual discussions among the participants. Additionally, we only consider e-mail threads in which at least one maintainer is part of it. To identify whether an e-mail in a thread was sent by a maintainer or not, we parsed the

MAINTAINERS file³, which is a file available in the Linux Git repository describing the list of maintainers and files of each Linux subsystem.

After detecting the release cycle in which the first e-mail of a thread occurs, we extract all information necessary to perform our analysis, such as the e-mail body, the date of the first e-mail of the thread, and the release number of each thread.

B. Pre-processing the e-mail body

Before being able to identify the sentiment polarity in the mailing list data, we clean up the e-mail body of each thread. For that, we group the body of all e-mails per thread, then convert the text to lowercase. We remove lines prefixed by >, e-mails, URLs, and greetings such as “kind regards”, “best regards”, “hi”, and “hello”.

C. Identifying the sentiment polarity

To automatically detect the sentiment polarity expressed in e-mail threads, we applied the state-of-the-art Senti4SD tool [10] over the pre-processed e-mail threads’ content. Senti4SD represents word meaning based on distributional semantics via a neural network, using an approach called word embedding [10]. The authors used word2vec to build the Distributional Semantic Model (DSM), in which words are represented as vectors. The DSM has been trained over 20 million questions, answers, and comments from Stack Overflow. Thus, Senti4SD is trained in a software engineering domain, which avoids negative bias from sentiments in other sources.

Senti4SD returns the polarity of a certain text without mentioning the probability scores (sentiment strength). As such, Senti4SD only returns whether the text is positive, negative, or neutral. In a manual analysis, the authors defined as positive polarity texts showing joy and love. Conversely, negative polarity was used to describe anger, sadness or fear, and the neutral polarity indicates the absence of emotion [10].

IV. CASE STUDY RESULTS

This section presents the results of our research questions. We present below the motivation, the approach, and the discussion of our findings for each research question.

A. How does the e-mail thread sentiment evolve in the kernel review discussions?

a) *Motivation:* Since maintainers play an essential role in the kernel development, we want to assess the sentiment behind their review discussions. For that, we analyze the overall sentiment polarity of e-mail thread, across releases, month, weeks, and weekdays.

b) *Approach:* We compute the sentiment polarity of all e-mail threads from January 2017 to January 2019 in which at least one kernel subsystem maintainer participated.

¹<https://lore.kernel.org/patchwork/project/lkml/list/>

²https://lore.kernel.org/lkml/_text/help/

³<https://github.com/torvalds/linux/blob/master/MAINTAINERS>

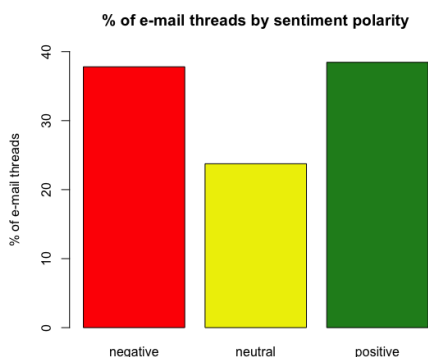


Fig. 2. Percentage of e-mail threads by sentiment polarity.



Fig. 3. Percentage of e-mail threads by sentiment and by releases.

c) **Findings:** Overall, 38% of the e-mail threads are positive, 38% are negative, and 24% are neutral. Figure 2 presents these percentages by sentiment polarity. Since these percentages aggregate sentiment across the entire studied period, we now focus on how this sentiment evolves across time.

No consistent trends in sentiment polarity are found at the release level. When analyzing the sentiment by releases (Figure 3), we can see that despite the fact that the number of positive and negative e-mail threads varies across releases, the number of positive and negative e-mail threads in each release remains relatively stable. As explored further in RQ2, we do notice that the releases right before and after Torvalds' temporary break (vertical line) saw a slightly larger proportion of positive e-mail threads compared to negative threads, yet this difference in proportion does not extend beyond.

Similarly, no consistent differences in sentiment polarity can be found at the month level, nor at the week level. After analyzing whether the sentiment changed across releases or

not, we want to assess if there is any variation across months. Thus, Figure 4 presents the percentage of emails by sentiment polarity and by month. As presented in the graphs, some months have more negative threads than others (e.g., 02/2017 and 03/2017) and vice versa (e.g., 09/2018 and 10/2018). Figure 5 shows similar results at the week level.

The overall distribution of positive, negative and neutral sentiment at the month level remains stable across time. To assess how the relative proportions in the values of positive, negative, and neutral sentiment evolve across months, we computed the Shannon Entropy [11] (third graph in Figure 4). An entropy of 1 indicates that the three sentiment values are purely uniformly distributed, while an entropy of 0 indicates that one of the three would be 100% while the other ones would be 0%. Our results show that the entropy values are around 0.5 across all months, with only slight changes across months.

Wednesdays and Thursdays see more positive e-mail threads compared to negative threads, while Saturday sees more negative e-mail threads. Figure 6 presents the percentage of e-mail threads by sentiment and weekday. There is not a huge difference in positive and negative sentiment across weekdays. However, the relative proportion of negative e-mail threads on Saturday is remarkable.

Finding for RQ1. There is no significant, nor consistent difference in the sentiment across releases, months, and weeks. Yet, we did find a larger proportion of negative e-mail threads on Saturdays.

B. Did the maintainers' sentiment change after Linus Torvalds' temporary break from the community?

a) *Motivation:* Four months after Linus Torvalds took a temporary break from the Linux kernel community, we want to analyze whether the maintainers' sentiment has changed. As a result of this research question, we will be able to assess whether this event has affected maintainer sentiment in the short term.

b) *Approach:* Similar to RQ1 (Section IV-A), we compute the sentiment polarity of e-mail threads sent by at least one subsystem maintainer. Particularly, we compare the period from January 2017 to September 2018 with the period from October 2018 to January 2019.

c) **Findings:** Right before and after Torvalds' temporary break, the same percentages of positive (38%), negative (38%) and neutral (24%) e-mail threads are found. First, we compared whether the percentage of e-mail threads by sentiment polarity changed before and after Linus took a temporary break. We could not find any difference in the percentage of positive, negative, and neutral e-mail threads. Furthermore, the resulting percentages were identical to those found in RQ1 (see Figure 2).

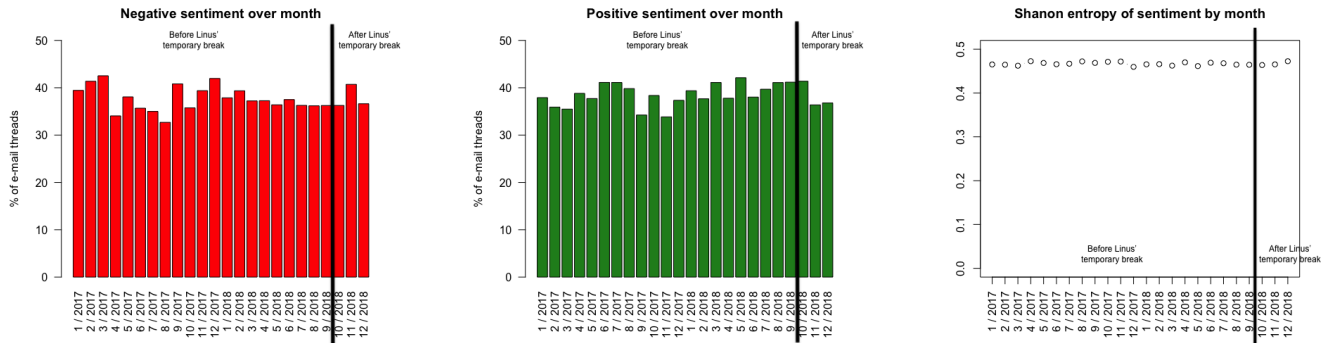


Fig. 4. Percentage of e-mail threads by sentiment and by month.

Overall, there is no difference in sentiment across releases before and after Torvalds’ break. By analyzing the changes in sentiment by releases (Figure 3), we can see that releases 4.18 and 4.19, which surround September 2018, had more positive e-mail threads compared to the next release (4.20). This release had a slight increase in the number of negative e-mail threads (622 e-mail threads) compared to positive threads (582 e-mail threads).

We identified minor increases in positive sentiment surrounding September 2018.

By looking at Figure 4, we can see that from July 2018 the sentiment started to become more positive (39.68% e-mail threads) compared to the negative sentiment for the same month (36.27% e-mail threads). Afterwards, in August 2018, 41% of the e-mail threads were positive, against 36.18% of negative e-mail threads. The same behavior remained in September and October 2018. However, in November 2018, the number of negative e-mail threads increased again to 40.71% compared to 36.37% positive e-mail threads. The next month, in December 2018, the positive and negative e-mail threads were similar. As a conclusion, one month after Linus Torvalds’ return to the Linux community, the initial sentiment changes had vanished again.

The sentiment slightly changed in the weeks before and after Linus took a temporary break. When analyzing the sentiment across week numbers (Figure 5), we can see that in week 38 (from September 17, 2018 to September 23, 2018), right before Linus’ temporary break, the number of positive e-mail threads increased compared to the number of negative e-mail threads in the same week (88 positive e-mail threads compared to 57 positive e-mail threads). Furthermore, after Linus’ come-back, in weeks 43 and 44 (from October 22, 2018, to November 04, 2018) the number of negative e-mail threads increased (68 and 78 e-mail threads, respectively) compared to positive e-mail threads (46 and 57 e-mail threads, respectively). This result aligns with the one of the monthly analysis, where November had more negative e-mail threads than the previous months.

***Finding for RQ₂.** While the sentiment before and after Linus break does not differ significantly, we did observe a slight increase in positive sentiment at the level of months and weeks after his break.*

V. DISCUSSION AND CHALLENGES

While we did not find any major trends in maintainer sentiment within e-mail threads, our results thus far are still exploratory in nature, since we only consider sentiment at a relatively coarse-grained level in both time (weekly level at best) and space (e-mail threads). Both kinds of granularity imply that we are aggregating sentiment across hundreds of different conversations, involving hundreds of different participants. While this aggregation does give a higher-level view of sentiment, it might cancel out the effects of particularly strong sentiments, both positive and negative ones.

The sentiment mining tool used, Senti4SD, is a state-of-the-art sentiment mining tool trained on a software engineering corpus. However, it only returns the polarity of a text snippet, not its strength. As such, we are unable to distinguish between for example an e-mail thread that is 100% negative compared to one that is 50% negative. Furthermore, the presence of sarcasm, mixing of natural language and source code, typos, emojis, etc. makes sentiment mining a noisy process, potentially missing or incorrectly classifying text according to a certain sentiment. Furthermore, apart from sentiment mining, techniques exist to mine other forms of human affect like emotions and politeness [12], [13].

Finally, since Linus break is a relatively recent event (September 2018), our data set does not yet allow to observe medium- to long-term effects of Linus Torvalds’ temporary break. We are continuing to gather data to replicate our study on a larger time scale.

VI. THREATS TO VALIDITY

We discuss the main threats to the study validity [14], with the respective minimizations, as follows.

Internal Validity. We assume a causal relationship between a developers’ sentiments and what they write in emails, based on empirical evidence conducted in different domains [15].

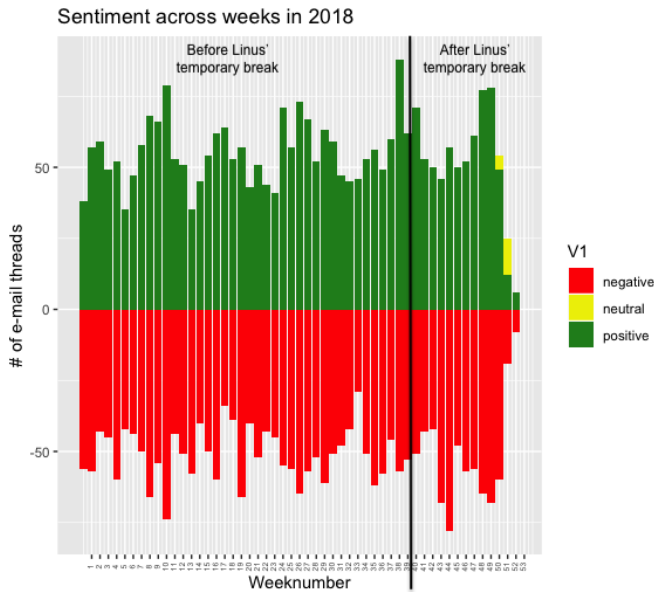
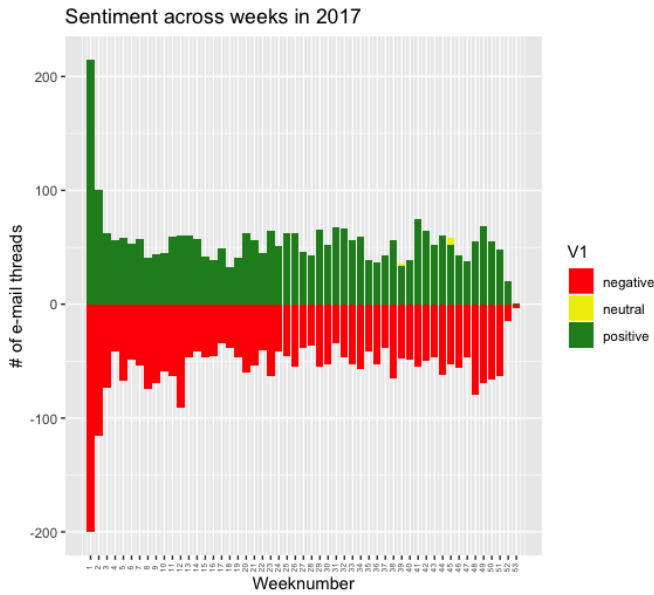


Fig. 5. Percentage of e-mail threads by sentiment and by week number.

Construct Validity. The preprocessing of the data may introduce some inaccuracies and may not remove all the noise in the data. Concerning Senti4SD, a manual analysis is still necessary to ensure the reliability of the sentiment polarities reported by the tool in the Linux kernel community. Even though the tool has a good precision and recall, and it was trained on software engineering data, the results might be inaccurate in the context of the kernel.

External Validity. We study emails from a popular and large open source project. However, we do not have evidence to support that the results found in this study are generalizable to projects with similar characteristics.

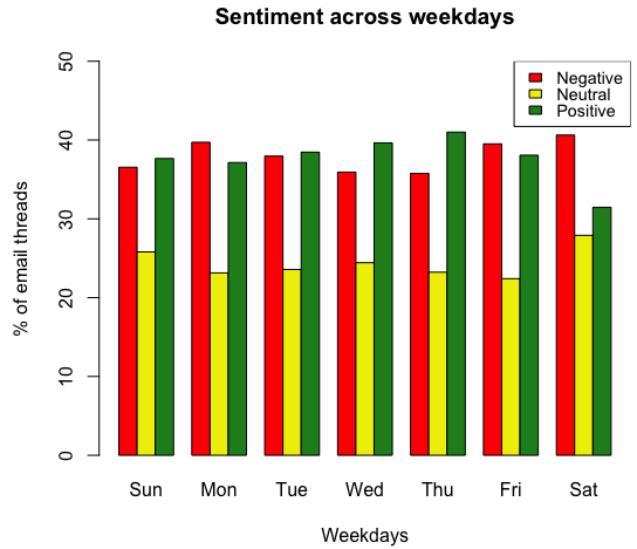


Fig. 6. Percentage of e-mail threads by sentiment polarity and by weekday.

VII. RELATED WORK

Tourani, Jiang, and Adams [16] have studied the presence and evolution of positive and negative sentiment in individual e-mails in mailing lists of two major Apache projects. As a result, the authors found that the mailing lists contain sentiment. Additionally, the authors found six categories of positive sentiment in individual e-mails and four categories of negative sentiment. Furthermore, they found that user mailing lists have more curiosity and sadness, and developer mailing lists have more aggression, announcement, and socializing.

Novielli et al [17] have studied the performance and reliability of four sentiment analysis tools customized for software engineering (SentiStrength, SentiStrengthSE, SentiCSR, and Senti4SD). Furthermore, Jongeling, Robbert, et al. [18] have studied whether the sentiment analysis tools agree with the sentiment recognized by human evaluators. Their results suggest a need for sentiment analysis tools specifically targeting the software engineering domain. Murgia et al. [19] have evaluated the feasibility of a tool for automatic emotion mining. They found that issue comments do express emotions regarding design choices, and maintenance activity. Regarding agreement among human raters, the authors found that emotions like love, sadness, and joy are easier to agree on.

Finally, Sinha, Vinayak, Alina Lazar, and Bonita Sharif [20] have analyzed the sentiment in commits within a seven-year time frame. The authors found that a majority of the sentiment in GitHub projects is categorized as neutral. Overall, more negative sentiment was detected on Tuesday, however, for the top five projects with the most commits, Wednesdays and Thursdays were the most negative.

VIII. CONCLUSION

As more people join collaborative software engineering activities, their discussions can get more excited and feature a variety of tones. A negative tone may impact the developer's productivity, creativity, and contribution. In this paper, we empirically assessed the maintainers' sentiment in the Linux Kernel Mailing List. We studied 15,166 e-mail threads from January 2017 to January 2019 and analyzed whether changes in sentiment could be observed surrounding Linus Torvalds' temporary break from the community.

Our results show that (i) most of the discussions are equally positive or negative, (ii) e-mail threads on Saturdays are slightly more negative, (iii) the proportion of positive and negative e-mail threads varies over time, with (iv) slightly higher percentages of positive threads around September 2018, when Linus Torvalds made a temporary break from the kernel community. We discussed several ways in these initial results need to be refined in order to better grasp the ability of sentiment to measure interactions in a developer community.

ACKNOWLEDGMENTS

The authors would like to thank the Linux Foundation for their support.

REFERENCES

- [1] 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*. ACM, 2014, pp. 352–355.
- [2] P. Tourani, B. Adams, and A. Serebrenik, "Code of conduct in open source projects," in *Software Analysis, Evolution and Reengineering (SANER), 2017 IEEE 24th International Conference on*. IEEE, 2017, pp. 24–33.
- [3] G. Hertel, S. Niedner, and S. Herrmann, "Motivation of software developers in open source projects: an internet-based survey of contributors to the linux kernel," *Research policy*, vol. 32, no. 7, pp. 1159–1177, 2003.
- [4] Y. Tian, J. Lawall, and D. Lo, "Identifying linux bug fixing patches," in *Proceedings of the 34th International Conference on Software Engineering*. IEEE Press, 2012, pp. 386–396.
- [5] "Linux founder to take some time off: I need to change some of my behavior," <http://fortune.com/2018/09/17/linux-git-linux-torvalds-bullying-abuse-time-off/>, accessed: 2018-12-26.
- [6] Y. Jiang, B. Adams, F. Khomh, and D. M. German, "Tracing back the history of commits in low-tech reviewing environments: a case study of the linux kernel," in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, 2014, p. 51.
- [7] F. Armstrong, F. Khomh, and B. Adams, "Broadcast vs. unicast review technology: Does it matter?" in *Software Testing, Verification and Validation (ICST), 2017 IEEE International Conference on*. IEEE, 2017, pp. 219–229.
- [8] Y. Jiang, B. Adams, and D. M. German, "Will my patch make it? and how fast?: Case study on the linux kernel," in *Proceedings of the 10th Working Conference on Mining Software Repositories*. IEEE Press, 2013, pp. 101–110.
- [9] N. Bettenburg, E. Shihab, and A. E. Hassan, "An empirical study on the risks of using off-the-shelf techniques for processing mailing list data," in *2009 IEEE International Conference on Software Maintenance*. IEEE, 2009, pp. 539–542.
- [10] 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.
- [11] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.
- [12] 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*. IEEE Press, 2015, pp. 303–313.
- [13] N. Imtiaz, J. Middleton, P. Girouard, and E. Murphy-Hill, "Sentiment and politeness analysis tools on developer discussions are unreliable, but so are people," in *2018 IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. IEEE, 2018, pp. 55–61.
- [14] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.
- [15] T. D. Cook, D. T. Campbell, and W. Shadish, *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin Boston, 2002.
- [16] P. Tourani, Y. Jiang, and B. Adams, "Monitoring sentiment in open source mailing lists: exploratory study on the apache ecosystem," in *Proceedings of 24th annual international conference on computer science and software engineering*. IBM Corp., 2014, pp. 34–44.
- [17] 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)*. IEEE, 2018, pp. 364–375.
- [18] R. Jongeling, P. Sarkar, S. Datta, and A. Serebrenik, "On negative results when using sentiment analysis tools for software engineering research," *Empirical Software Engineering*, vol. 22, no. 5, pp. 2543–2584, 2017.
- [19] A. Murgia, M. Ortu, P. Tourani, B. Adams, and S. Demeyer, "An exploratory qualitative and quantitative analysis of emotions in issue report comments of open source systems," *Empirical Software Engineering*, vol. 23, no. 1, pp. 521–564, 2018.
- [20] V. Sinha, A. Lazar, and B. Sharif, "Analyzing developer sentiment in commit logs," in *Proceedings of the 13th International Conference on Mining Software Repositories*. ACM, 2016, pp. 520–523.