Monitoring Sentiment in Open Source Mailing Lists – Exploratory Study on the Apache Ecosystem

Parastou Tourani   Yujuan Jiang   Bram Adams
Polytechnique Montréal, Canada
{parastou.tourani, yujuan.jiang, bram.adams}@polymtl.ca

Abstract

Large software projects, both open and closed source, are constructed and maintained collaboratively by teams of developers and testers, who are typically geographically dispersed. This dispersion creates a distance between team members, hiding feelings of distress or (un)happiness from their manager, which prevents him or her from using remediation techniques for those feelings. This paper evaluates the usage of automatic sentiment analysis to identify distress or happiness in a development team. Since mailing lists are one of the most popular media for discussion in distributed software projects, we extracted sentiment values of the user and developer mailing lists of two of the most successful and mature projects of the Apache software foundation. The results show that (1) user and developer mailing lists carry both positive and negative sentiment and have a slightly different focus, while (2) work is needed to customize automatic sentiment analysis techniques to the domain of software engineering, since they lack precision when facing technical terms

Keywords—Empirical Software Engineering, Sentiment Analysis, Mining Software Repositories, Mailing List Data

I. Introduction

Research in psychology, economic and organizational behaviour shows the importance of happiness and job satisfaction at work. Various books and papers [1], [2], [3] emphasize that happy people are more creative, learn more and achieve greater success at their work. Work in administrative science [4] highlights the direct linear relationship between positive sentiment (i.e., feelings like happiness, joy, excitement or, contentment) and creativity in organizations. Researchers also have found evidence of the economic impact of happiness and attitude of employees. For example, a recent study [5], saw 10-12% greater productivity for happier individuals, concluding that social scientists may need to pay more attention to emotional well-being as a causal force for productivity at work. Finally, organizational behaviour research [6], showed that affective factors are closely tied to the feelings of employees about their work and company.

Since software development and maintenance are a collaborative activity as well [7], it seems intuitive that the sentiment of software project members also plays a pivotal role in the success or failure of a software product, however software managers have a hard time keeping track of their people's feelings. Of course, the best way to know how people feel is by talking and working with them in person. Surprisingly, this becomes more and more difficult, not just in open source development, but also in traditional companies. For example, larger companies are distributed across a different campus or even country. Recently, companies like Mozilla even started promoting remote work. For example, in 2012 the release engineering team (12 people), which is the backbone responsible for bringing new features to customers through official releases, was spread across 4 different (non-contiguous) time zones to offer around-the-clock service and improve quality of life. Their manager noted "out-of-sight, out-of-mind is a real concern". It is even harder to know how the user community feels about a project, since this group is several orders of magnitude larger and spread all across the globe.

As such, in many cases electronic communication in the form of email (1-to-1 or mailing lists), chat rooms, video conferencing or phone calls have become the de facto means of communication. While one could expect people to be more reserved or careful when using those media since almost all of them record conversations (in contrast to face-to-face discussions), Bacchelli et al. [8] noted that people have become so accustomed to these communication channels that most of them freely express their actual feelings in their communications. Such feelings can range from positive expressions like "Great work so far!" to "who are the stupid people who manage this group?". In other words, for many organizations, emails and chat messages are one of the primary means of conveying and picking up signals and indications of good or bad feelings of colleagues and employees [9], [10], [8].

To help organizations and open source projects in picking up signals of good or bad sentiment easier and more accurately, this paper explores, as the first step, whether automatic sentiment analysis tools are able to identify periods of extremely positive or negative feelings in the developer and user community of two major Apache projects. Automatic sentiment analysis is an emerging area that blends natural language analysis and psychology to obtain cues about an individual's opinion or feelings towards a product. Our work, if successful, opens the door towards a new research area of customizing sentiment analysis and other techniques for software development and software maintenance. In particular, we address the following research questions:

RQ1) *How accurate is existing sentiment analysis on software engineering data sources like mailing lists?*

Since sentiment analysis (and the corresponding tools) primarily have been used in the context of psychology, finance or organizational behaviour, we first evaluate how well they perform on software engineering data sources, which typically contain technical terms. We find that the precision of SentiStrength for positive periods is 29.56%, and for negative periods 13.18%. These low values are due to ambiguities in technical terms as well as the difficulty of SentiStrength to distinguish extremely positive/negative documents from neutral ones. For Ant, some correlations between the number of closed bugs and appearance of sentiments in the user mailing list have been observed.

RQ2) *What types of sentiment can be observed in software engineering mailing lists?*

To understand the role of sentiment in software projects, we manually studied a representative sample of developer and user emails. 19.77% of the emails contain positive sentiment, compared to 11.27% for negative sentiment. We could distinguish 6 categories of positive sentiment, and 4 categories of negative sentiment.

RQ3) *Do developers and users show different sentiment?*

Finally, we studied whether users and developers of a software project show different sentiment in mailing list communication. The user mailing list and developer mailing list of each project show only little similarity in their sentiment trends. For emails with positive sentiment, user mailing lists contain substantially more "Curiosity", but less "Announcement" and "Socializing", while for emails with negative sentiment, user mailing lists contain more "Sadness" and less "Aggression" than developer mailing lists.

In the remainder of this paper, we first describe the background notions for sentiment analysis (Section II). Next, we describe the experimental setup (Section III). We then address the three research questions (Section III-D) and discuss our findings. After threats to validity (Section V) and related work (Section VI), we finish with conclusions (Section VII).

II. Background

This section provides background about sentiment, sentiment analysis, and the SentiStrength tool used in this paper.

A. Sentiment Analysis

The term "sentiment" (also "affect") refers to people's opinion, attitude, appraisal or emotions toward entities, events and their attributes [11]. Sentiment differs from "emotion", which is a state of feeling and distinguishes human feelings at a finer-grained level into categories like sadness, happiness, shame, and anger. Sentiment and emotion occur automatically when people communicate, since it helps people convey their message or understand other people's reactions. This implicit behaviour is not just limited to communications in real world, but even when people interact through computer aided communications [12], like comments or feedback that people make on community fora or chat rooms, or in more conventional electronic media like emails. These sentiments are universal, in that they occur as much in politics as in business contexts [13]. Even in software engineering [14], identification of sentiments and emotions in software artifacts can provide an indication of someone's opinions towards certain project decisions or other people.

In order to automatically measure sentiment from recorded (typically textual) transcripts of communication, semantic measures have been proposed as a measure of subjectivity and opinion in text. Those measures usually measure the polarity (positive or negative) and "strength" (also "degree") of a document. Strength shows to which degree a word, phrase, sentence or document is positive or negative towards a subject topic, person, or idea [15]. For example, a positive opinion with strength 1 is much more positive than a positive opinion with strength 0.1. Some examples of approaches to extract the above measures are sentiment analysis [13], opinion mining [13], and affect mining. This paper focuses on sentiment analysis.

Sentiment analysis algorithms mainly use two approaches: machine learning or a lexical approach. With machine learning, typically text documents are taken as input and a classifier is produced as output [16]. According Aue et al. [17], classifiers perform very well when they are applied in the domain on which they were trained, otherwise their performance decreases significantly. Lexical approaches use language information in the form of a list of known sentiment-related words, their polarities and the grammatical structure of the language, then uses those to score the sentiment of the text. Word lists or dictionaries for lexical approaches can be created manually, or could be expanded by using seed words [18].

B. SentiStrength

SentiStength uses a lexical approach [19] to estimate sentiment score for an informal English text. SentiStrength divides the given text into sections, then based on the words or phrases within each section and other language information like grammatical structure knowledge, it assigns both a positive and negative value to the section since according to psychological research a human can experience both negative and positive feelings for the same piece of text [12]. These values for positive values range from 1 to 5, and for negative from -5 to -1. To calculate the sentiment of each word or phrase, SentiStrength looks up the word or phrase in its lexicon and (if found) uses the associated sentiments (positive and negative), otherwise a value of zero (no sentiment). Since each word or phrase might have different score, and a section consists of multiple phrases, the score of the whole section should be derived from the individual phrases' scores, taking into account the structure of the sentence. Structure of the sentence, for example, refers to modifier word like "very" and "extremely", which act as boosters and alter the score, or to symbols, punctuations like "!" and smilies.

We decided to use a lexical sentiment analysis tool rather than a machine learning-based one as the former algorithms are simpler and it have been used successfully in several research projects [19], [18], [20]. In addition, compared to many existing commercially-oriented opinion mining tools, SentiStrength considers sentiments related to expressing friendship or showing social support [19].

III. Experimental Setup

This section explains the methodology used to address the research questions of the introduction.

A. Selection of Subject Systems

Mailing lists are the core means of project communication in open source communities like Apache, where developer and user mailing lists are used during software development and maintenance to discuss technical issues, propose changes, report bugs, or ask how-to questions about configuration or any other parts of the product.

For this reason, this study investigates the mailing lists of two major projects of the Apache Software Foundation, i.e., Tomcat and Ant. Tomcat is an open source web server and servlet container first released in 1999[1], while Ant is a software tool for automating software build processes[2] with initial release in July 2000. Both of them are mature and widely deployed, successful projects.

Of both projects, we downloaded the developer and user mailing list data in the textual mbox format from the official Apache archive pages[3].

B. Pre-processing of Data

Before being able to analyze the email data, we first had to filter out emails that did not contain actual human content. A major category of emails to filter out were automatic confirmation emails. Such emails are sent automatically by various servers like Bugzilla issue tracking systems or version control systems. Obviously since these emails are not sent by humans, they are fully objective (neutral) and hence are not the subject of our study. They only contain source code patches or reports originally submitted elsewhere (not on the mailing list). To filter out automated emails for each project, we manually identified the different patterns that they might have. Some of these patterns have special subjects, while others have a specific sender of email. Using regular expressions, we reduced the email data from 635,906 emails, down to a total of 595,673 emails. Table I shows how this number is distributed across the four studied mailing lists. To avoid duplication of contents of emails, we also removed the quoted parts of email threads as they have considered in their original emails.

Table I: Number of emails per mailing list

Ant Developer	Ant User	Tomcat Developer	Tomcat User	Total
20,292	169,329	360,733	45,319	595,673

After recovering all non-automated emails, the next step is to filter out any non-natural language text inside these emails. The unstructured and noisy nature of the emails related to the development of a software system causes many emails to contain technical information about design, implementation (e.g., source code or excerpts related to reported bugs) and defect-related information like stack traces. Bacchelli et al. [8] founded that the content of development emails can be classified into five categories: natural language, source code, patch, stack trace and junk (i.e., textual information like the signatures or spam status of authors).

To find sentiment values of emails, only the natural language category of email content should be taken into attention. For this, Bacchelli et al. [10] found that lightweight methods based on regular expressions were the most effective. For this reason, we used a combination of regular expressions and searching for lines with special characters and keywords to filter out uninteresting email content like source code or stack traces.

C. Sentiment Score Computation

In order to automatically detect the sentiment expressed in emails, we applied the SentiStrength tool and ran it over the pre-processed emails of developers and users. SentiStrength scores each line of email with a value from -5 to +5, however we need one sentiment score for each email as a whole. To find the best way to aggregate the SentiStrength scores of all lines of an email into one value [21], we ran an experiment on a random sample of 100 emails from the Tomcat project. Given the large number of emails, we sampled enough emails to have a confidence level of 95% and confidence interval of 5%. Two of the authors manually scored the sentiment of each email. Then, we compared the manual score to the following aggregation methods across SentiStrength's line-level scores: minimum, 1st-quartile, average, median, 3rd-quartile and maximum. Note that the "maximum" method corresponds to finding the most extreme value, be it negative or positive.

Table II shows that the Mean, Median and Max Value methods are the most accurate aggregation methods. However, Mean and Median only worked well for neutral sentiment, whereas the Max method also had an accuracy of 36% for positive sentiment and 21% for negative sentiment. For this reason, we chose the Max Value for our purposes. This seems reasonable, since an email usually consists of a small number of sentences and the sentence with the maximum value of sentiment likely dominates the overall sentiment of an email.

Table II: Accuracy of different aggregation methods for SentiStrength.

	Min	1st Qu.	Median	Mean	3rd Qu.	Max
Positive	0%	0%	0%	0%	100%	36%
Negative	100%	100%	0%	0%	0%	21%
Total	14%	14%	64%	64%	19%	39%

D. Analysis of the Sentiment Values

The specific analysis used for each research question is presented in the next section with the corresponding findings. When studying the evolution of sentiment, we abstract up from the sentiment of individual emails to the average Max sentiment of all emails sent in one month. A period of one month in open source development strikes a nice balance between being too short (nothing significant happening) and being too long (multiple releases happening).

[1]http://tomcat.apache.org/
[2]http://ant.apache.org/
[3]http://mail-archives.apache.org/mod_mbox/

IV. Experimental Results

RQ1. How accurate is existing sentiment analysis on software engineering data sources like mailing lists?

Motivation. Detecting sentiment of software team members is essential in modern software development and maintenance, where members are mostly geographically distributed and therefore physical face-to-face meetings are impossible or scarce. In such environments, automatic monitoring of sentiment, either of individuals or of a whole community, can play an important role in managing software projects and identifying potential risks that might threaten the sustainability of a project.

Our first research question explores how precise a modern sentiment analysis tool is in detecting the sentiments of software-related mailing lists. In particular, we use a popular sentiment analysis tool to identify positive and negative peaks of sentiment across time, with the aim of enabling managers or team leads to identify good or bad trends in the sentiment of project stakeholders. Depending on the outcome of the analysis, existing techniques could (a) be used as is to monitor sentiment in a software engineering context, (b) might need customization, or (c) might need to be reconsidered for the purpose of identifying the sentiment in a software project.

Approach. Given the large number of emails, we sampled 800 emails, out of a total of 595,673 existing emails, which is an appropriate number to obtain a confidence level of 95% and confidence interval of 5%, i.e., 400 in the user mailing lists and 400 in the developer mailing lists. To equally cover the lifetime of each of the four mailing lists, each mailing list had 100 emails sampled from its most positive months and 100 emails from its most negative months. These 100 positive (resp. negative) emails of a project were picked from the mailing list's 4 months with the most positive resp. most negative sentiment (8 critical months). To avoid being stuck with 4 consecutive top months (limiting our evaluation to a too narrow period), we choose at most one most positive (resp. negative) month per year, (starting with the year having the highest/lowest peaks). In other words, for a given mailing list, we end up selecting the 25 emails with the highest sentiment score in each of the 4 top months, and the 25 emails with the lowest sentiment score in each of the 4 most negative months.

After sampling, two separate raters read the emails and manually scored them with a positive, negative or neutral value. They ignored the amplitude of the SentiStrength scores, just focusing on the sign (positive/negative/neutral), since by definition these emails correspond to the most extreme (positive/negative) emails and our goal was to validate whether this was correct. We then used the manual validation to calculate a precision value for SentiStrength. Since the two raters obtained an agreement of 76.62%, our precision values are relative to the emails for which both raters agreed.

Findings. Sentiment evolves over time, with a lot of variation in the form of upward and downward trends. Figure 1 shows the evolution of the average SentiStrength score per month for the 4 mailing lists. We can see how average sentiment is bounded between -0.15 and 0.2 for the Tomcat mailing lists, with a peak up to 0.3 for the Tomcat developer mailing list. The Ant mailing lists go from -0.2 to 0.2, with

Table III: Confusion matrices of SentiStrength

Actual by raters for developer		Computed by SentiStrength -1	Computed by SentiStrength 1
	-1	26	10
	0	113	89
	1	19	50

Actual by raters for user		Computed by SentiStrength -1	Computed by SentiStrength 1
	-1	15	18
	0	125	95
	1	13	50

Actual by raters for both		Computed by SentiStrength -1	Computed by SentiStrength 1
	-1	41	28
	0	238	184
	1	32	39

Table IV: Precision and empirical recall of SentiStrength.

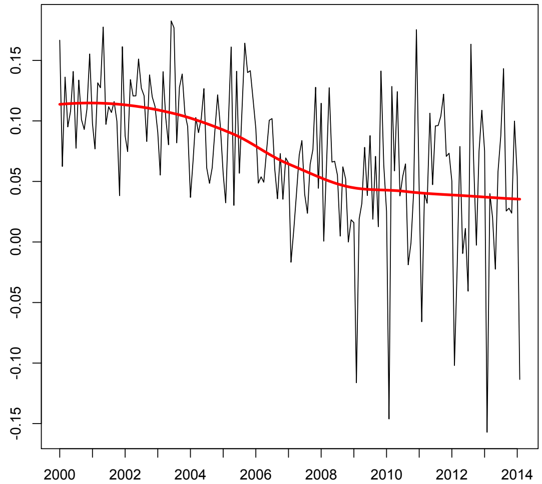
	Positive Precision	Negative Precision	Total Precision	Positive Recall	Negative Recall	Total Recall
Developer	33.55%	16.45%	24.75%	72.46%	72.22%	72.38%
User	25.65%	9.80%	17.70%	75.00%	45.45%	63.52%
Total	29.56%	13.18%	21.24%	73.55%	59.42%	68.42%

peaks over -0.4 and 0.4 for the user mailing list, and even 0.6 for the developer mailing list.

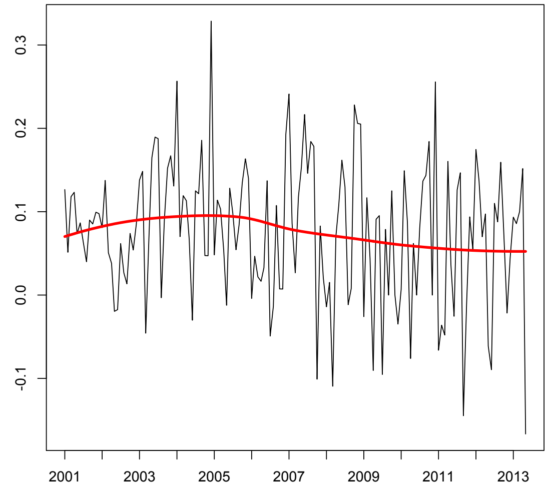
Given the jagged nature of the plots, a lot of noise is present. For practical applications, one should either filter the noise (i.e., putting average values below a certain threshold to zero) or focus only on the most extreme peaks, since those are indicators of major problems or opportunities in the project that could be worth investigating. As an example of the first kind of filtering, we added loess local regression lines [22], which are smoothed regression lines based on a running average. They are ideal to identify the predominant trend in a noisy curve. We can see how the Ant user and Tomcat developer mailing lists have a more or less constant trend, while the Tomcat user mailing list sees a clearly downward trend and the Ant developer mailing list an upward trend. It is important to note that the average values of the trend line remain slightly positive, even for the downward trends. This indicates that, overall, both projects have a healthy, i.e., positive, community.

The second kind of filtering, i.e., only focusing on the most extreme peaks, yields for each mailing list a small number of very large values. For Ant, the peaks get more extreme towards the right, as can be seen in Figure 1. This seems to be linked to a decreasing number of emails being sent to the mailing lists (the volume dropped from an average of 1758 emails in 2000-2007 to an average of 365 in 2008-2014). Tomcat sees many more extreme values over time, often in bursts. This is why our manual analysis considered maximum one positive/negative peak per year, since otherwise our analysis would only consider a very narrow period of time.

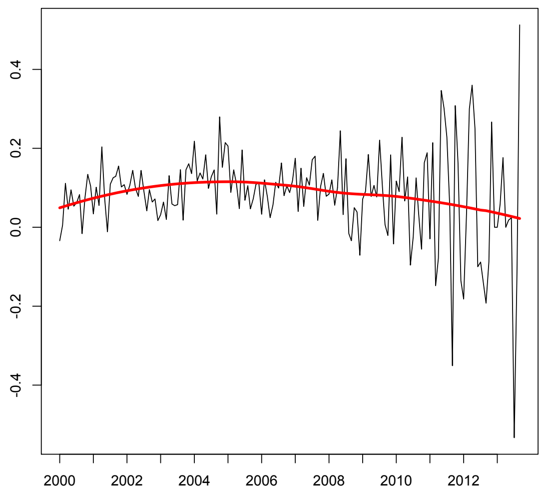
The precision of SentiStrength for positive months is 29.56%, while for negative months it is 13.18%. Table IV shows the evaluation results of SentiStrength by the two raters, while Table III shows the confusion matrix of each group. We see how for positive emails, SentiStrength obtained a precision of 29.56%, while for negative emails a precision of 13.18%. For reference, the SentiStrength documentation mentions a 60.7% precision for positive texts and 64.3% for negative texts on documents on the social web, which is much higher than



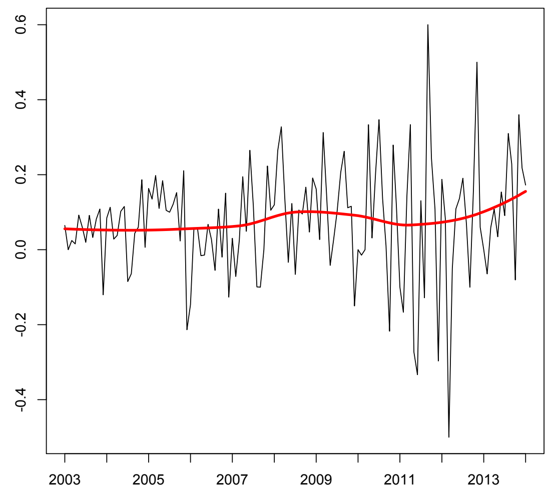
(a) Tomcat User



(b) Tomcat Developer



(c) Ant User



(d) Ant Developer

Figure 1: Average monthly sentiment value across time for the four mailing lists. (For readability, the plots have different Y axes.)

the numbers that we obtained for the emails.

One possible explanation for the relatively low precision is that most of the emails in the top positive and negative months turned out to be neutral. This is confirmed by calculating the empirical recall, i.e., recall where we use the set of all emails identified as positive (resp. negative) in our sample of 800 as oracle (basically ignoring the neutral emails). Table IV shows that positive empirical recall is higher than 72.46%, which indicates that positive sentiments primarily are found in the set of 400 positive emails, not the negative emails. In other words, precision is low because the sets of 400 emails contained too many neutral emails rather than emails of the opposite sentiment (e.g., negative emails in positive sample). Only for the user mailing lists we see a relatively low empirical recall, which means that only half of the negative emails showed up in the negative data set.

Investigating the result of the SentiStrength, another possible explanation can be found in the nature of software development and maintenance emails, where people mostly write about problems or solutions in a very technical manner. Many technical keywords are used that are (a) unknown to SentiStrength's pre-compiled list of phrases or (b) sometimes already known to have a positive or negative sentiment when used in a non-technical context. For example, "Safe", "Security", "Value", "Support" and "Dynamic" are existing English terms with a known positive sentiment, while "Kill", "Defect", "Error", "Disabled", "Failure" and "Default" are known to be negative, while neither of these are interpreted as positive/negative in software development or maintenance area: they are just technical terms used in a different (technical) meaning. Table V shows examples of sentences identified to be incorrect due to technical jargon.

Positive months have a larger variation in precision than negative months. Table VI shows the distribution of precision per month. Especially for negative months, precision is concentrated in a narrow band [0.09,0.19], while for positive

Table V: Examples of Sentences with Incorrect SentiStrength Score.

Incorrect Positive Samples	Incorrect Negative Samples
that dynamically add JARs to lookup classes (I share Connor's concerns about this)	And I can only (try to) fix errors which occur ...
trunk with support for types and tasks only.,At least for starters.	can even enforce dependencies of the code compiled by the different
It seems eclipse has a security manager enabled. That means tasks that have	We thought this was an error on WinRARs side, so an user contacted the
security checks will perform them and if they are loaded by ant class	1) It might be good to have EXIT_ON_INIT_FAILURE=3Dtrue by default in TC8.
on the objects currently declared on that role and their respective XML	duplicate. While the original problem was indeed related to using EL in
was to provide something like ProcessHelper class that can be stored as a ref	Currently Tomcat HTTP 1.1 Connector disables the use of chunked encoding if
	The problem is with those locales for which CharsetMapper.getCharset(locale)
	returns null.,There is an error in ResponseBase.setLocale() that it will set
	land, so that we could e.g. log an error if we encounter invalid data

Table VI: Distribution of precision per month.

	Min	1st Qu.	Median	Mean	3rd Qu.	Max
Positive	0.04167	0.17800	0.27530	0.31050	0.40530	0.68750
Negative	0.00000	0.09348	0.15790	0.14050	0.18900	0.25000

months we see more variation [0.18,0.41]. In other words, although the negative peaks have a low precision, this low precision seems evenly distributed across each month and hence the general trend of negative sentiment (as in, for example, Figure 1) remains correct. For positive months, this is less the case, yet those months tend to have a higher precision. Hence, although the low precision values on the one hand mean bad news, the sentiment results can still be interpreted.

Automated lexical sentiment analysis obtains a precision of around 30% and 13% for positive and negative months, respectively, especially because of neutral emails and ambiguous terms being misinterpreted.

RQ2. What types of sentiment can be observed in software engineering mailing lists?

Motivation. As mentioned earlier, mailing lists play an important role in communication among team members involved in software development of open source software. Different individuals including developers and users talk about numerous issues in various stages of the project via emails: developers discuss about the problems that they encountered in design or implementation, announce some essential news about the project like a product release or some critical changes that they have made, acknowledge other people's help, while users ask questions about using the product, suggest solutions to improve the product and so on [23]. During these discussions, both developers and users also convey their opinions or sentiment. Our second research question seeks to understand the different kinds of such sentiment in OSS mailing lists.

Approach. We investigated the same emails sampled in RQ1 to address this research question. The same two raters categorized the emails based on the emails' sentiment value. Even if an email was classified incorrectly by the algorithms, we still analyzed its true sentiment(positive/negative), such that each of the 800 emails was tagged with a category. Our categorization is orthogonal to that of Bacchelli et al. [23], since they focused on the content of the emails while we focus on whether or not the person writing the email feels positively or negatively. We started the categorization on one mailing list, then applied (and enhanced) it on the other mailing lists. As a third step, any discrepancies between the two raters' categories were discussed and resolved.

Findings. Table VIII gives an overview of the categories of the sentiments identified as well as their relative proportions, while Table VII shows representative examples of each category. 34.21% of sample emails expresses sentiments in the developer mailing lists, compared to 27.87% for the user mailing lists. The percentages for individual categories should be interpreted like this: 34.71% of the 19.77% developer emails with positive sentiment correspond to "Satisfactory Opinion". In the following, we define each category and discuss the results across all analyzed emails ("Total" in Table VIII), while RQ3 compares categories between user and developer mailing lists.

Positive Sentiment.

- *Satisfactory Opinion:* In its simplest form, satisfactory opinions are feelings of sympathy or positive impressions that people have towards a software system, new release, new feature or code change. Depending on the specific mailing list (developer or user), these subjects vary from users talking about a special feature or a product as a whole to developers dealing with new changes or parts of the code. According to Table VIII, this group of emails is the most popular among emails with positive sentiment.
- *Friendly Interaction:* Since software development is a collaborative activity, constructive communication amongst the people involved might lead to higher productivity [14]. Well-mannered interactions with a positive undertone are a good start towards this. In response, if these emails are answered and guided with respect and a positive attitude, the interaction continues in a friendly and constructive manner. This group of email usually contains expressions of appreciation and support, such as "Hope this helps", "Thanks and really appreciate it". Since the goal of development mailing lists is to ask and answer questions of practitioners and users during software development and maintenance, being able to measure the amount of *Friendly Interaction* can help us to identify the heartbeat and soul of a community. Together with Satisfactory Opinion, *Friendly Interaction* is the most popular positive sentiment in the analysed mailing lists.
- *Explicit Signals:* Independent of the content type, authors often directly write indications about their good mood, for example in the form of emoticons

Table VII: Examples of categorization of email sentiments.

Category	Example
Satisfactory Opinion	Thank you very much Gianluca!,Great work so far.,And I know it will... We (UF, I cannot claim to represent the tomcat devs) are happy with a,simple round robin distribution for new requests.
Friendly Interaction	Thanks and really appreciate your assistance. butwith a good community effort we should be able to be done within a reasonable timeframe and enjoy a successful 2.0 version!
Explicit Signals	Fixing leaks is good :) Oh wait, there'sWindow$, so I guess there are takers ;)
Announcement	I am pleased to announce that I have a version of IvyDE ready to be released. With 8 +1 votes and no 0 or -1 votes, the vote is successful and Charles Duffy is now a committer.
Socializing	If anyone is interested in getting together for some drinks or exploring the city (I've never been to Vancouver) on Thursday, email me privately [. . .] Beer is always acceptable, though sometimes tough to ship.
Curiosity	I am keen on having two web applications be able to share sessions. I can't reproduce the scenario that causes the deletion but the"autoDeploy" attribute has piqued my curiosity.
Unsatisfied Opinion	Isn't this a bit premature, junit 4 isn't even "out" yet. And no, Tomcat is not reliable at all, it's more kinda toy for boreddevelopers such as me.
Aggression	Anyone volunteering to buy me a second pair of glasses? Or does gump drink too much ? who are the stupid people who manages this group.
Uncomfortable Situation	I'm absolutely furious that Tomcat did not say (almost) anything in its logs. I have not been into ivy(ide) (yet) and currently heavily constrained ontime (new job)
Sadness	Oh geez... really?,We're going to have a top-post vs bottom-post flame-war?? I was very sloppy and changed the wrong one.

(e.g., smilies), which reinforce the positive sentiment.

- *Announcement:* Many emails are written by someone to announce good news from the author's perspective, such as a new release that mitigates severe problems, or advertisement for a person during a vote for new members to join the developer or committer team. This group of emails expresses valuable interactions and milestones in a project.
- *Socializing:* Emails are used to socialize between community members by sending and discussing invitations to visit each other or attend community events or meetings. Such emails obviously contribute to a positive sentiment since they might strengthen good relationships among people.
- *Curiosity:* In the large amount of emails containing questions and answers, we found that there are other indicators for positive sentiments too. One important one is when an email shows signals of curiosity when asking a question, or encourages and reveals signs of hope while answering a question. Such emails have positive sentiments since they provide a hint of participative and aspiring behaviour of a community. Hence, despite their relatively small market share in Table VIII, this group of emails is highly valuable for a community.

Negative Sentiment.

- *Unsatisfied Opinion:* In contrast to *Satisfactory Opinion*, emails with negative sentiment might contain unpleasant or even offensive opinions towards various issues that people complain about. Similar to the positive counterpart, these issues can cover topics ranging from the software system as a whole to individual contributions or characteristics of a project like a code change, or even a new feature proposed by a person. *Unsatisfied Opinion* is one of the most important kinds of emails with negative sentiment.
- *Aggression:* This category covers emails with signs of poor and destructive communications, like flamewars, or people attacking or insulting each other. A second group of emails in this category consists of less extreme emails that ask their question or report a problem while complaining or while answering to an email in an angry way. As shown in Table VIII, *Aggression* is one of the most frequently occurring negative sentiments, together with *Unsatisfied Opinion*.
- *Uncomfortable Situation:* Some emails have indicators that reveal the author of the email to be in an uncomfortable situation, such as suffering from a problem for ages, or being confused about unexplainable behaviour of the software system, or worrying about risks and fears. Some emails even reveal their authors to be under severe pressure like time constraints that might overwhelm them. This category is as common as the *Aggression* category. Independent of their specific rationale, these emails refer to negative symptoms reflecting poor quality of the software or parts of it (in the eyes of the unhappy author of the email), or simply to disagreement with management of the project.
- *Sadness:* Finally, there are also emails in which authors explicitly apologize or express feelings of sadness towards a problem. Although not aggressive, such emails also are carriers of unpleasant news or events, which is why we grouped them under negative sentiment. Fortunately, in many cases other people follow up comforting the sad author, giving rise to *Friendly Interaction*.

Neutral Sentiment. Emails with neutral sentiment refer to emails that show no sentiment, for example the author just describes a solution or a problem in a (possibly detailed) technical way, without showing specific emotions or other subjective signs. Another example of this group are the typical howto emails seen a lot in developing mailing lists in which guidelines or steps for doing a task are explained.

Based on Table VIII, emails in which authors give their opinion towards an issue or towards the answer that they have got, can have a significant impact on the sentiment of emails. Similarly, emails in which the authors express situations such

Table VIII: Categorization of email sentiment.

		Total		Developer		User	
Positive Sentiment	Satisfactory Opinion	19.77%	34.71%	22.48%	33.33%	17.05%	36.54%
	Friendly Interaction		33.88%		33.33%		34.62%
	Explicit Signals		13.47%		14.49%		12.12%
	Announcement		6.61%		8.70%		3.85%
	Socializing		5.79%		7.25%		3.85%
	Curiosity		6.61%		2.90%		11.54%
Negative Sentiment	Unsatisfied Opinion	11.27%	18.84%	11.73%	19.44%	10.82%	18.18%
	Aggression		11.59%		16.66%		6.06%
	Uncomfortable Situation		52.17%		52.78%		51.51%
	Sadness		17.39 %		11.11%		24.24%
Neutral Sentiment		68.95%		65.80%		72.13%	

as the constraints or limitations that they have encountered, play an important role in negative sentiment emails. Finally, the quality of interactions among a community during the development of the project polarizes sentiment in emails substantially. By quality of interaction, we refer to how grateful, or supportive and helpful stakeholders are when interacting with each other or, conversely, in contrast how offensive they are. Apart from these factors, there are some minor factors that also can affect the sentiment of emails, like the amount of desire that people have towards a task, or friendship among people, as shown in the *socializing* category.

19.77% of the sampled emails were positive, 11.27% negative. We identified 6 positive sentiment categories, and 4 negative ones.

RQ3. Do developers and users show different sentiment?

Motivation. Now that we have categorized emails with different sentiment (positive, negative or neutral), we can analyze potential differences between developer and user mailing lists in terms of stakeholder sentiments. Intuitively, these two mailing lists have different purposes and different individuals subscribed to them. Typically, a developer mailing list is used for discussions about the actual development of the project, such as changes to the source code and related issues including bug fixes. On the other hand, configuration, how-to and support questions about the product are sent to the user mailing list. Despite these different purposes, one could expect sentiment in the developer list to be coupled to the user list, for example when complaints about a major bug trigger stressful discussions between developers to fixing this bug. Our third research question analyzes whether such coupling of sentiment does occur.

Approach. To answer this question, we compare the average monthly sentiment plots of Figure 1 between the developer and user mailing lists of both projects. We also use time series analysis to compute the cross-correlation between both pairs of user/developer mailing list to quantitatively measure a (possibly lagged) correlation between both lists [22]. Finally, we compare the popularity of sentiment categories in Table VIII between both types of lists.

Findings. The user mailing list and developer mailing list of each project do not necessarily follow a similar trend. First of all, comparing the plots of users and developers shows that Tomcat and Ant follow different trends. While the Tomcat mailing lists both feature a downward trend from sentiment values around 0.1 to 0.05, the Ant mailing lists see an opposite trend, with the developer mailing list suddenly seeing a surge in average monthly sentiment towards 0.15 and higher instead of a downward trend towards 0.05. However, even for Tomcat the trends are not that highly correlated: the highest cross-correlation between developer and user mailing list occurs for a lag of 4 months, but only reaches a correlation of 0.19. Ant has a slightly higher (but still low) correlation of 0.22 for a lag of 1 month. Interestingly, in both cases the lag is positive, which suggests that the sentiment of the developer mailing list tends to follow (in time) that of the user mailing list. A potential hypothesis is that bugs and new features typically are proposed by users, then trickle down to developers.

The developer plots show substantially more fluctuation in sentiment value than the user plots, with a very large variation between the lowest and highest sentiment values. For example, the Tomcat developer list has positive peaks reaching 0.3, and the Ant developer list even reaching 0.6. The user mailing lists, even though varying as well, seem more compact, except for the last couple of months. The latter is likely due to the lower mailing list volume in that period, as discussed earlier.

User mailing lists contain substantially more "Curiosity", but less "Announcement" and "Socializing". Considering positive sentiment emails, developer and user mailing lists are quite similar since both have the highest proportions for *Satisfactory Opinion* and *Friendly Interaction*, which in total comprise around two-thirds of all positive emails. For users, these proportions are even a little higher. Furthermore, users express more their *Curiosity* about different issues, which means that users also convey more desire in comparison with developers. This seems normal, as most of the time, there are more newcomers among the users that are in the process of becoming more familiar with the system. Hence, those users show more desire to learn and obtain answers such that they become able to use the system properly. On the other hand, we can see a higher percentage of developers in the *socializing* category, which means that among developers there is a more friendly and decontracted atmosphere. Considering that developers need to collaborate more closely, this observation indeed makes sense. The *announcement* category also takes up a bigger part in the developer mailing list. The reason for this is that among developers there are often announcements for a candidate during a vote in addition to regular announcements related to a new software release, while there are hardly any such announcements for users.

User mailing lists contain substantially more "Sadness", while developer mailing lists contain a lot more "Aggression". Indeed, comparing emails with negative sentiment, we can detect two differences between developers and users.

Users adopt apologies and direct expressions for revealing their *Sadness* about two times more frequently than developers. This might be due to the fact that users are more likely to inadvertently make certain mistakes, after which they apologize or demonstrate similar expressions.

Finally, we have found that "Aggression" emails are a lot less common in the user mailing list, i.e., interactions among users rarely involves bad manners (6% vs. 19%). This means that developers more often state their negative opinions about bugs or new features. This might show that developers are very passionate about their work and the project as a whole, while (most of the) users are less negative than one would expect up front when complaining about their problems. The latter is a bit surprising, since in half of the negative user emails the user is unhappy because she is in a "Uncomfortable Situation" and definitely needs help.

Generally, we can say that developers and users show different proportions of sentiment categories during the construction and maintenance of the software project. This seems to confirm the different roles and perspectives of both groups of stakeholders towards a software system, and hence their attitude towards each other can be different. The developer mailing list comprises communication among colleagues, while the other list basically contains customer support communication. The main constant factor in both is the software product that is being discussed.

Sentiment on developer mailing lists chronologically seems to follow that on user mailing lists. User lists feature more "Curiosity" and "Sadness", but less "Aggression", "Announcement" and "Socializing".

V. Discussion and Threats to Validity

Although we have identified different categories of sentiment in developer and user mailing lists, and motivated our work based on the link between sentiment and productivity in other domains, thus far we have not analyzed the possible link between sentiment and productivity in software development teams. Although this is outside the scope of the exploratory study performed in this paper, to better understand our results we did perform a small, initial study checking possible correlation between sentiment and productivity in the form of bug fixing activity.

For this reason, we have extracted the number of closed bugs for each month in the Tomcat and Ant bug repositories as an indicator of the effectiveness of the developers in fixing bugs. Similar to RQ3, we then calculate the cross-correlation between the monthly number of closed bug reports and lagged versions of the average Max SentiStrength emotion score. We then check large correlation values (positive or negative), as well as the corresponding lag. Only for the two Ant mailing lists, we found significant correlations of -0.43 for the user mailing list and 0.65 for the developer list. In both cases, we obtained this correlation for a lag of 5 months, in the sense that 5 months after a month with a particular average Max sentiment score, a higher number of closed bugs is observed. For the Ant developer list, we also observed a negative correlation of -0.62 for negative lag of 7 months. For the Tomcat project, no such high correlations could be observed.

As an start for an explanation, we studied the Ant release history. We found that the mean time between successive releases is around 6.5 months and the median time is around 4.2 months. The correlations for a lag of 5 months might be related to this release cycle time. Even if this would be correct (more analysis is needed for this), it is still not clear why the user and developer mailing lists show opposite correlation signs, nor why the Ant developer list shows a second negative correlation. We plan on exploring this in future work.

Regarding the threats, internal validity threats concern factors that might mistify the obtained results. We assume a causal relationship between a developer's sentiments and what he or she writes in emails, based on empirical evidence conducted in different domains [24]. In addition, the emails used in this study were collected over an extended period from developers or users not aware of being monitored, hence we are confident that the sentiments that we found are genuine. Another internal threat to validity is whether one can deduce the correct sentiment based on emails in isolation, without considering earlier emails in the thread. In most of the analyzed emails, the individual emails were indeed clear. In a minority of cases, when suspecting irony or observing specific cases of jargon, the raters looked up the earlier emails in the thread.

Threats to construct validity focus on how accurately the observations describe the phenomena of interest. Different stages in preprocessing of data such as filtering out automatic emails or extracting the natural text from emails might introduce some inaccuracies. However, after each stage enough testing has been done to assure the correctness of the data. With respect to the SentiStrength tool's results, our sample in RQ2 and RQ3 is based on the most extreme (positive/negative) results of the tool. Even though the tool's results had a low precision, there is no reason to believe that the results (and hence our sample) are systematically biased, thus we do not believe this impacts our results in a major way. To determine the correct sentiment of each email, we relied on human raters. In earlier work [14], we performed a user study with a large group of raters in the context of emotion mining. This showed that human raters agree sufficiently on "joy" and "sadness", which roughly coincide with positive and negative sentiment. For this reason, we used only two raters for this study.

Threats to external validity correspond to the generalizability of our experimental results [24]. In this paper, we study emails from two popular open source projects. We chose the two successful mature projects as a representative sample of the universe of open source software projects, with different development teams and from different domains. We have no evidence to support the assertions that these results are generalizable even to other projects that have most similarities with the studied projects. Replication studies should confirm whether other similar open source projects confirm our study results or not.

VI. Related Work

Substantial work [25], [26], [27] has shown the influence of emotions on work results as well as on personnel effectivity in different workplace types. For example, De Choudhury et al.

describe how assessment of employees' feelings enables an organization to detect causes of joy, sadness and frustration among the employees, based on which plans can be made to improve general emotions, workgroup dynamics, employee collaboration and hence work effectiveness [28]. Positive feelings inside a community can be an indicator of the quality and value of the interactions between people, which is why it is vital to support managers to discover the emotions of their teams [14].

Previous studies mostly used sentiment analysis in the areas of marketing and financial markets but not in software engineering. For example, many online markets like mobile app stores or Amazon provide facilities for customers to assess their products and give their opinion. In such cases, sentiment analysis can be applied on the reviews of customers for products and services. Twitter and Facebook are also popular websites for sentiment analysis applications like monitoring the reputation of a specific brand [29].

Similarly, analysis of financial markets uses sentiment analysis on news items, articles, blogs and tweets about companies to drive automated trading systems like StockSonar [29]. Vivek Sehgal et al. [30] introduced a new approach for stock prediction based on sentiments of online messages, from which correlations between stock values and sentiments are learnt to enable prediction. Sanjiv R. Das et al. [30] designed an algorithm to train small investor sentiment classifier from stock message boards, which can be used to assess the impact of small investor behaviour on stock market activity.

Despite extensive work on sentiment analysis for product reviews, marketing and financial markets, few research has studied the role of sentiment or emotion analysis in software engineering. Recently, Marta N. Gómez et al. [31] examined whether the personality factors of team members and team climate factors are related to the quality of the developed software by the team. Analysis of student projects showed that software quality has a significant correlation with personality traits of team members like extroversion and team climate factors such as participation. Finally, they derived guidelines for software project managers with respect to team formation. Peter C. Rigby et al. [32] also used LIWC, a psychometrically-based linguistic analysis tool, to study the Apache httpd developer mailing list. In their study, they assessed the personality of four top developers, and two top developers that have left the project. They also examined the word usage on the mailing lists near releases to find the general attitude of developers in these periods. Blerina Bazelli et. al. [33] studied the personality traits of authors of questions on StackOverFlow.com, which is one of the most popular Question and Answers website used by all kinds of programmers. As a replication of Rigby et al.'s work, they applied LIWC (this time on SO questions), then categorized the extracted personalities based on the online reputations of the analyzed authors. They found that top reputed authors are more extrovert and issue less negative emotions. Against these studies, which are about intrinsic personality of developers, our paper looks at instantaneous sentiments to obtain the general trend of community sentiment.

Munmun et al. explored various emotional expressions of employees at 500 large software corporation by characterizing the emotional expression of the employees in a fine-grained continuous manner via posts on an internal Twitter-like microblogging tool [28]. They empirically show that affective expression in the enterprise can be the result of various workplace factors. These factors can be exogenous and endogenous workplace factors, geography of organization or the organizational hierarchy. This analysis extracted sentiment of employees over time, by analysing textual content of microblog posts using LIWC. They concluded that affective expression in the workplace can provide an efficient tool for assessing key factors and performance relevant outcomes.

Guzman et al. [21] used latent Dirichlet allocation to find the topics discussed in collaboration artefacts like messages from mailing lists and web discussions in university projects. They then used lexical sentimental analysis on the topics to obtain an average emotion score for each of the topics. They evaluated their approach by interviewing the project leaders, which revealed the need for more details in the generated topic summaries.

Zhang et al. [34], using natural language processing and sentiment analysis techniques applied on online forums, investigated how to extract problematic API features, i.e., features that cause difficulties for API users and often are discussed in a forum. Their study was conducted by means of a preliminary manual analysis, followed by an empirical evaluation. In particular, they extracted phrases from online threads and realized that meaningful problematic features mostly appeared in the phrases that contain negative sentences or the neighbours of negative sentences. By meaningful features they referred to features that help an API support team to find out what kinds of problems their users have and that also help them to find out how they can improve their API more effectively.

Regarding the use of development mailing lists as source of valuable information related to software development, comprehension, and maintenance, Bacchelli et al. classified email content at the line level [8]. By combining parsing techniques and machine learning, they partitioned the content of development emails in five categories, i.e., natural language, source code, patch, stack trace, and junk. Later, Bacchelli et al. [23] also conducted research to better understand mailing list communication. They analysed OSS mailing lists both quantitatively and qualitatively, showing the wide range of topics discussed in email threads apart from source code, such as project status and social interactions. Our paper analyzes sentiment in the natural language category of email content.

VII. Conclusion

Instead of algorithms or techniques to improve technical software development or maintenance issues, this paper focused on the human aspects involved with these activities. In particular, we studied the presence and evolution of positive and negative sentiment in the email communication of users and developers of two large open source projects.

On the one hand, we found that a state-of-the-art automatic sentiment analysis tool obtains only a modest precision due to the presence of ambiguous technical terms and the difficulty of distinguishing neutral (technical) emails from positive or negative ones. Hence, substantial work is needed to customize off-the-shelf sentiment analysis tools to the domain of software engineering. Still, the relatively uniform precision across each month already allowed to observe certain trends in the data.

On the other hand, we observed that developer and user mailing lists do contain sentiment (resp. 19.77% and 11.27% of the emails). We identified 6 categories of positive sentiment in emails, and 4 categories of negative sentiment. Furthermore, the two types of mailing lists have their own focus, with user mailing lists having more curiosity and sadness, and developer mailing lists more aggression, announcements and socializing. Furthermore, we found weak correlations that suggest sentiment in the developer mailing list to chronologically follow that of the user mailing list.

This paper only scratched the surface of sentiment analysis in mailing lists, hence a lot more work is needed on other systems and other tools. Ultimately, the goal of this field is to warn managers and other leading stakeholders of extremely positive or negative sentiment in a project, such that they can choose which of the widely known team building or other activities are necessary to improve the stakeholders' sentiment, morale and productivity.

ACKNOWLEDGMENTS

We would like to thank Emily Coffey for her valuable insights on the perception of emotions.

REFERENCES

[1] I. Robertson and C. Cooper, *Wellbeing: Productivity and Happiness at Work*. Palgrave Macmillan, 2011.

[2] S. Rao, *Happiness At Work*. McGraw-Hill Education (India) Pvt Limited, 2010.

[3] C. Barron and A. Barron, *The Creativity Cure: How to Build Happiness with Your Own Two Hands*. Scribner, 2013.

[4] T. M. Amabile, S. G. Barsade, J. S. Mueller, and B. M. Staw, "Affect and Creativity at Work," *Administrative Science Quarterly*, vol. 50, no. 3, pp. 367–403, 2005.

[5] A. J. Oswald, E. Proto, and D. Sgroi, "Happiness and productivity," Bonn, IZA discussion papers 4645, 2009.

[6] "Predicting Turnover of Employees from Measured Job Attitudes." *Organizational Behavior & Human Performance*, vol. 13, no. 2, pp. 233–243, Apr. 1975.

[7] I. Mistrìk, J. Grundy, A. Hoek, and J. Whitehead, Eds., *Collaborative Software Engineering*. Springer, 2010.

[8] A. Bacchelli, T. D. Sasso, M. D'Ambros, and M. Lanza, "Content classification of development emails," in *Proc. of the 34th Intl. Conf. on Software Engineering (ICSE)*, 2012, pp. 375–385.

[9] P. C. Rigby, D. M. German, and M.-A. Storey, "Open source software peer review practices: A case study of the apache server," in *Proceedings of the 30th International Conference on Software Engineering*, ser. ICSE '08. New York, NY, USA: ACM, 2008, pp. 541–550.

[10] A. Bacchelli, M. Lanza, and R. Robbes, "Linking e-mails and source code artifacts," in *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 1*, ser. ICSE '10. New York, NY, USA: ACM, 2010, pp. 375–384.

[11] N. Mishra and C. K. Jha, "Article: Classification of opinion mining techniques," *International Journal of Computer Applications*, vol. 56, no. 13, pp. 1–6, October 2012, published by Foundation of Computer Science, New York, USA.

[12] M. Thelwall, "Heart and soul: Sentiment strength detection in the social web with sentistrength (summary book chapter)," In press.

[13] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, Jan. 2008.

[14] A. Murgia, P. Tourani, B. Adams, and M. Ortu, "Do developers feel emotions? an exploratory analysis of emotions in software artifacts," in *Proceedings of the 11th IEEE Working Conference on Mining Software Repositories (MSR)*, Hyderabad, India, May 2014.

[15] C. Osgood, G. Suci, and P. Tenenbaum, *The Measurement of meaning*. Urbana:: University of Illinois Press, 1957.

[16] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA: Association for Computational Linguistics, October 2013, pp. 1631–1642.

[17] A. Aue and M. Gamon, "Customizing sentiment classifiers to new domains: A case study," in *RANLP*, 2005.

[18] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Comput. Linguist.*, vol. 37, no. 2, pp. 267–307, Jun. 2011.

[19] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment in short strength detection informal text," *J. Am. Soc. Inf. Sci. Technol.*, vol. 61, no. 12, pp. 2544–2558, Dec. 2010.

[20] O. Kucuktunc, B. B. Cambazoglu, I. Weber, and H. Ferhatosmanoglu, "A large-scale sentiment analysis for yahoo! answers," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, ser. WSDM '12. New York, NY, USA: ACM, 2012, pp. 633–642.

[21] E. Guzman and B. Bruegge, "Towards emotional awareness in software development teams," in *Proc. of the 2013 9th Joint Meeting on Foundations of Software Engineering (ESEC/FSE)*, 2013, pp. 671–674.

[22] R. Kabacoff, *R in Action: Data Analysis and Graphics With R*. MANNING PUBN, 2013.

[23] A. Guzzi, A. Bacchelli, M. Lanza, M. Pinzger, and A. van Deursen, "Communication in open source software development mailing lists," in *The 10th Working Conference on Mining Software Repositories*, 2013, pp. 277–286.

[24] D. T. Campbell and J. C. Stanley, *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin, 1963.

[25] S. Ambler, *"Agile modeling: effective practices for extreme programming and the unified process"*. John Wiley & Sons, Inc. New York, 2002.

[26] "Linus Torvalds defends his right to shame Linux kernel developers," http://www.webcitation.org/6O2zErgzE, accessed: 01/10/2013.

[27] B. L. Fredrickson, "The role of positive emotions in positive psychology: The broaden-and-build theory of positive emotions." *American psychologist*, vol. 56, no. 3, p. 218, 2001.

[28] 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*, ser. CSCW '13. New York, NY, USA: ACM, 2013, pp. 303–316.

[29] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, pp. 82–89, Apr. 2013.

[30] V. Sehgal and C. Song, "Sops: Stock prediction using web sentiment," in *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, ser. ICDMW '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 21–26.

[31] G. Canfora, L. Cerulo, M. Cimitile, and M. D. Penta, "How changes affect software entropy: an empirical study," *Empirical Software Engineering*, vol. 19, no. 1, pp. 1–38, 2014.

[32] P. C. Rigby and A. E. Hassan, "What can oss mailing lists tell us? a preliminary psychometric text analysis of the apache developer mailing list," in *Proceedings of the Fourth International Workshop on Mining Software Repositories*, ser. MSR '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 23–.

[33] B. Bazelli, A. Hindle, and E. Stroulia, "On the personality traits of stackoverflow users." in *ICSM*. IEEE, 2013, pp. 460–463.

[34] Y. Zhang and D. Hou, "Extracting problematic api features from forum discussions," in *Proc. ICPC*. IEEE, 2013, pp. 142–151.