Are Bullies more Productive? Empirical Study of Affectiveness vs. Issue Fixing Time

Marco Ortu*, Bram Adams ‡, Giuseppe Destefanis†, Parastou Tourani ‡, Michele Marchesi * Roberto Tonelli *
*DIEE, University of Cagliari, Italy, {marco.ortu,michele,roberto.tonelli}@diee.unica.it
†CRIM, Computer Research Institute of Montreal, Canada, {giuseppe.destefanis}@crim.ca
‡École Polytechnique de Montréal, Canada, {bram.adams,parastou.tourani}@polymtl.ca

***Abstract*—Human Affectiveness, i.e., the emotional state of a person, plays a crucial role in many domains where it can make or break a team's ability to produce successful products. Software development is a collaborative activity as well, yet there is little information on how affectiveness impacts software productivity. As a first measure of this impact, this paper analyzes the relation between sentiment, emotions and politeness of developers in more than 560K Jira comments with the time to fix a Jira issue. We found that the happier developers are (expressing emotions such as *JOY* and *LOVE* in their comments), the shorter the issue fixing time is likely to be. In contrast, negative emotions such as *SADNESS*, are linked with longer issue fixing time. Politeness plays a more complex role and we empirically analyze its impact on developers' productivity.**

***Index Terms*—Affective Analysis, Issue Report, Empirical Software Engineering**

I. Introduction

Team sports like soccer [1] are a primary example that the productivity of an organization is not only a product of the talent in a team, but depends heavily on human affectiveness, i.e., the way in which individuals feel and how they perceive their colleagues [2]. A rude coach without people management skills will only alienate his team, prompting them to just do anything to avoid his scorn rather than focusing on winning the next game. Highly talented players with family issues likely have difficulties to focus on their job, while selfish, greedy or opportunistic players disrupt the harmony in a team. On the other hand, a group of medium-level players could grow into a winning squad if they enjoy working together and form a cohesive team.

Similar to sports teams, human affectiveness in software engineering has a huge impact on the abilities of a software organization [3] [4], yet the need to collaborate with remote teams (both in closed and open source development) makes the situation even more challenging [5] [6]. The fact that people do not work physically in the same location not only makes coordination of tasks more difficult, it requires them to align with colleagues and interpret colleagues' feelings through emails, discussion boards (e.g., issue tracking systems) and conference calls. The exclusive use of such systems and the absence of face to face communication could encourage developers in pursuing impolite communicative behaviour [7], which is known to detract newcomers from a project [8]. Many famous examples of this exist on the Linux kernel mailing list, for example in exchanges between the creator of the Linux kernel and some of the Linux developers[1].

In previous research [9], the authors manually analyzed whether discussion boards like bug repositories contain emotional content. They indeed found evidence of gratitude, joy and sadness, and also weak evidence that the presence of emotions like gratitude was related with faster issue resolution time. However, due to the manual nature of the analysis, the data sample was relatively limited. Furthermore, emotions are but one of the possible human affectiveness measures, and might not have the strongest relation with issue resolution time.

In this paper, we empirically analyze more than 560K comments of the Apache projects' Jira issue tracking system to understand the relation between human affectiveness and developer productivity. In particular, we extract affectiveness metrics for emotion, sentiment and politeness, then build regression models to understand whether these metrics can explain the time to fix an issue. We aim to address the following research questions:

RQ1: ***Are emotions, sentiment and politeness correlated to each other?***

The considered affective metrics have a weak correlation with each other.

RQ2: ***Can developer affectiveness explain the issue fixing time?***

Affective metrics are significant for explaining the issue fixing time. Our logistic regression model has a Precision of 0.67 and a Recall of 0.671 against 0.319 and 0.565 for a Zero-R baseline model.

RQ3: ***Which affective metrics best explain issue fixing time?***

Emotions such as *JOY* and *LOVE* reduce the resolution time, whereas emotions such as *SADNESS* increase the issue resolution time. Issue average politeness also increases the issue fixing time.

The rest of the paper is organized as follows: we first discuss related work (Section II). In Section III, we describe how we measure affectiveness by measuring emotions,

[1] http://arstechnica.com/information-technology/2013/07/linus-torvalds-defends-his-right-to-shame-linux-kernel-developers/

sentiment and politeness in developers' comments. Section IV introduces the Apache projects' Jira Issue Tracking System dataset and our methodology. In Section V we present and discuss our findings, followed by a discussion of threats to validity in Section VI. We finally draw our conclusions in Section VIII.

II. Related Work

The Manifesto for Agile Development [10] indicates that individuals and interactions are more important than processes and tools. David Parnas defined software engineering as multi-person development of multi-version programs [11] [12].

As such, the study of social aspects and psychological states [13] in software engineering is gaining, lately, more and more importance. Roberts and al. [14] conducted a study that reveals how the different motivations of open source developers are interrelated, how these motivations influence participation, and how past performance influences subsequent motivations.

Researchers are focusing their effort on understanding how the human aspects of a technical discipline can affect the final results [15] [16][17]. Feldt et al. [18] focused on personality as one important psychometric factor and presented initial results from an empirical study investigating correlations between personality and attitudes to software engineering processes and tools.

To enhance emotional awareness in software development teams, Guzman et al. proposed a sentiment analysis approach for discussions in mailing lists and web-based software collaboration tools like Confluence [4]. They used lexical sentiment analysis to analyze the relationship between emotions expressed in commit comments, with different factors such as programming language, time and day of the week in which the commit was made. Results showed that projects developed in Java have more negative commit comments, and that commit comments written on Mondays tend to contain more negative emotion.

Steinmacher et al. [8] analyzed social barriers that hampered newcomers' first contributions. These barriers were identified considering a systematic literature review, students contributing to open source projects, and responses collected from OSS projects' contributors. The authors indicated how impolite answers are considered as a barrier by newcomers.

Rigby et al. [19] analyzed the five big personality traits of software developers in the Apache httpd server mailing. Bazelli et al. [20] studied the personality traits of authors of questions on StackOverFlow.com. 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. Tourani et al. [21] evaluated the usage of automatic sentiment analysis to identify distress or happiness in a development team. They extracted sentiment values from the mailing lists of two of the most successful and mature projects of the Apache software foundation considering both users and developers. They found that user and developer mailing lists bring both positive and negative sentiment and that an automatic sentiment analysis tool obtains only a modest precision on email messages due to their relatively long size compared to tweets or issue comments.

Compared to Tourani et al. [21], this paper focuses on developers' comments (more than 560K comments) and uses a wider corpus of 14 systems to study how affectiveness affects the issue resolution time.

Murgia et al. [9] manually analyzed whether development artifacts like issue reports carry any emotional information about software development. The significant result of the study, that paved the way to our study, is that issue reports express emotions towards design choices, maintenance activity or colleagues.

Gomez et al. [22] analyzed 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 is correlated with team members' personality traits like extroversion and team climate factors such as participation. They derived guidelines for software project managers with respect to team formation.

Ortu et al. [23] studied 14 open source software projects developed using the Agile board of the JIRA repository. They analysed all the issue comments written by the developers involved in the projects to study whether the politeness of the comments affected the number of developers involved over the years and the time required to fix any given issue. Results indicated that the level of politeness in the communication process among developers has an effect on both the time required to fix issues and the attractiveness of the project to both active and potential developers. The more polite developers were, the less time it took to fix an issue, and, in the majority of the analysed cases, the more the developers wanted to be part of a project, the more they were willing to continue working on the project over time.

Compared to Ortu et al. [23], this paper analyzes two additional affectiveness metrics (emotions and sentiment), as well as uses logistic regression to compare the impact of all affectiveness metrics and common issue report metrics together, instead of using a univariate model using only politeness.

III. Background

In this section, we describe the three kinds of affective metrics studied in this paper: politeness, sentiment and emotion. This three metrics have been used by other researchers, i.e., politeness [23] and [24], sentiment [25] and [26], and emotion [9].

A. Politeness

Politeness is "the ability to make all the parties relaxed and comfortable with one another[2]." Danescu et al. [24] proposed a machine learning approach for evaluating the politeness of

[2]http://en.wikipedia.org/wiki/Politeness

Comment	Confidence Level
Can you put more detail in description ? If you can attach what was done in 0.89-fb branch, that would be nice. Thanks, <dev_name_b>	0.83
<dev_name_a>, can you open a new Jira for those suggestions? I'll be happy to review.	0.919
<dev_name_a>, can you submit a patch against trunk? (Sorry, thought I tagged this 0.7 to begin with.)	0.8

TABLE I: Examples of polite comments.

Comment	Confidence Level
Why are you cloning tickets? Don't do that.	0.816
- why blow away rack properties? - how does this allow talking to non-dynamic snitch?	0.85
<dev_name_a>, What is the point of doing that?	0.81

TABLE II: Examples of impolite comments.

Wikipedia[3] and Stackoverflow[4] requests. Since Stackoverflow is well-known in the software engineering field and is largely used by software practitioners, the model that Danescu et al. used [24] is suitable for our domain, i.e., Jira [5] issues, where developers post and discuss about technical aspects of issues. The authors provide a Web application[6] and a library version of their tool.

Given some text, the tool calculates the politeness of its sentences providing as a result one of two possible labels: *polite* or *impolite*. Along with the politeness label, the tool provides a level of confidence related to the probability of a politeness class being assigned. We thus considered comments whose level of confidence was less than 0.5 as neutral (namely the text did not convey either politeness or impoliteness). Table I and II show some examples of polite and impolite comments as classified by the tool[7].

B. Sentiment

We measured sentiment using the state-of-the-art SentiStrength tool[8], which is able to estimate the degree of positive and negative sentiment in short texts, even for informal language. SentiStrength by default detects two sentiment polarizations:

- Negative: -1 (slightly negative) to -5 (extremely negative)
- Positive: 1 (slightly positive) to 5 (extremely positive)

It uses a lexicon approach based on a list of words in order to detect sentiment. SentiStrength was originally developed for English and was optimised for short social web texts. We used SentiStrenght to measure the sentiment of developers in issue comments (which often are short).

C. Emotions

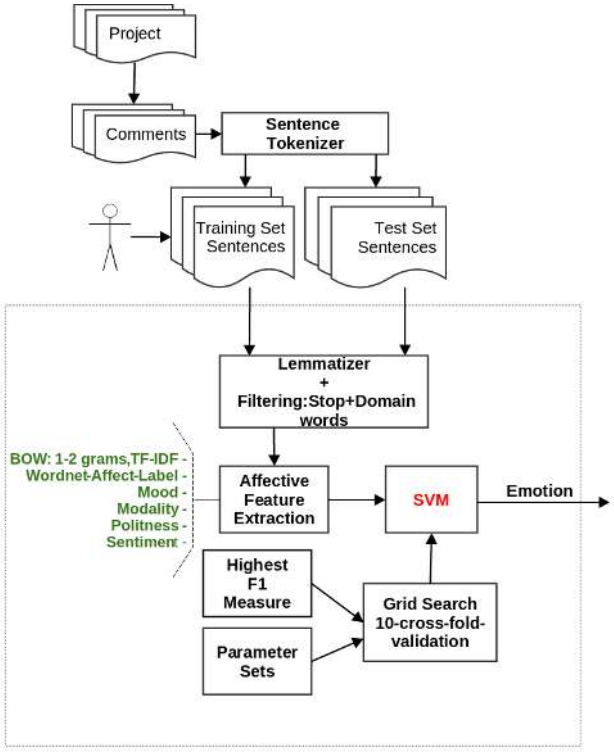

Fig. 1: Emotion Classifier Architecture

While sentiment is a measure of positive or negative emotion expressed in a given text relative to some topic, emotions are more fine-grained and relate to a particular *emotional state*. This corresponds to a variety of human feelings such as *LOVE* or *ANGER*. Different emotion framework exists, which decompose emotions into a basic set of emotions. Similar to Murgia et al. [9], we used Parrott's emotional framework, which consists of six basic emotions: joy, sadness, love, anger, sadness, and fear.

Despite conceptual frameworks like Parrott's Framework, to the best of our knowledge there is no available emotion analysis tool such as the ones available for measuring sentiment and politeness. For this reason, we built a machine learning classifier able to identify the presence of four basic emotions: *JOY*, *LOVE*, *ANGER* and *SADNESS* (these are the most popular emotions identified by Murgia et al. [9] in issue comments). Figure 1 shows the emotion classifier's architecture.

As input, the classifier requires all comments posted on a project's issue tracking system. For each comment, we used a sentence tokenizer [9] that divides a comment into sentences. For each sentence, we applied a classic text preprocessing approach, removing all the stop words and the domain words.

[3]https:en.wikipedia.orgwikiMain_Page
[4]http:stackoverflow.com
[5]Jira Issue Tracking System https://www.atlassian.com/software/jira
[6]http://www.mpi-sws.org/cristian/Politeness.html
[7]User's names are reported as *<dev_name_a>* for the sake of privacy.
[8]http://sentistrength.wlv.ac.uk/
[9]http://nlp.stanford.edu/software/tokenizer.shtml

Emotion	Accuracy	Precision	Recall	F1
ANGER	0.770	0.746	0.737	0.736
JOY	0.892	0.788	0.733	0.746
SADNESS	0.855	0.847	0.798	0.812
LOVE	0.881	0.798	0.772	0.775

TABLE III: Emotion classifier performance

Developers' comments often contain code, such as code snippets or stack traces, and in order to remove this text (which is irrelevant for emotion detection), we filtered out non-English words within a sentence using Wordnet[10]. The output of the *Lemmatizer* block is a vector containing all the words of a sentence. We enhanced each sentence vector considering the bi-grams (all individual words and all pairs of consecutive words) before performing the affective feature extraction. Using bi-grams is useful for considering negation such as *"don't like"*, which would not be considered using single words.

The *Affective Feature Extraction* block then extracts the following affective features:

- Affective labels: we used the Wordnet Affect label [27] to obtain an affective label [11] for each sentence's words.
- Mood: we used the tool based of De Smedt et al. [28] to measure the grammatical mood, i.e., the presence of auxiliary verbs (e.g., could, would) and adverbs (e.g., definitely, maybe) that express uncertainty.
- Modality: we used the same tool to measure the degree of uncertainty expressed in a whole sentence.
- Sentiment: the sentence's sentiment measured using Sentistrength.
- Politeness: the sentence's politeness measured using Danescu et al.'s tool [24].

For each of the four emotions, we built a dedicated Support Vector Machine classifier, since this kind of classifier has proven to be particularly suitable for text classification. It has several parameters and we used a grid search algorithm [12] using the F1 score [13] in order to find the optimum tuning configuration. We used a manually annotated corpus of comments and their emotion for training the machine learning Classifiers, one for each emotion. The training set consisted of 4000 sentences (1000 for each emotion), which was manually annotated by three raters having a strong background in computer science (Elfenbein et al. [29] provided evidence that for members of the same cultural and social group it is easier to recognize emotions than for people belonging to different groups).

A sentence was marked as containing a particular emotion if at least two out of three raters marked the presence of that particular emotion. If not, the sentence was marked as not having that emotion (and also added to the training set). We validated our emotion classifier using Bootstrap validation with 1000 iterations [14]. Bootstrap validation splits a dataset in training and test set according to a given ratio (we used 90% training - 10% testing) and generates *N* sets (1000 in our case) uniformly sampled with replacement from the initial dataset. This technique yields more stable measures of accuracy precision and recall, compared to other validation techniques such as cross-validation or leave-one-out validation.

Table III shows the performance obtained during bootstrap for each of the four machine learning classifiers. The models obtained a very high performance on the annotated corpus of comments. Given the (still) limited size of the training set, this may be due to some degree of overfitting. However, for emotions like LOVE and SADNESS, the most influential words used by the classifiers are "thanks" and "sorry", which are extremely common words across issue comments. In that sense, the models are relatively general. Since these models are a first attempt to design an emotion classifier, we decided to adopt the models in our study. Future research should focus on enhancing emotion classification.

IV. Case Study Setup

A. Dataset

We built our dataset collecting data from the Apache Software Foundation Issue Tracking system, Jira [15], since Apache is one of the most studied software ecosystems [21]. An Issue Tracking System (ITS) is a repository used by software developers as support for corrective maintenance activities like Bug Tracking, along with other types of maintenance requests. We mined the ITS of the Apache Software Foundation, collecting issues from 2002 to December 2013. Table IV shows the corpus of 14 projects selected for our analysis, highlighting the number of comments recorded for each project and the number of developers involved. We chose the top 14 projects with the highest number of comments since our focus is to measure the affectiveness expressed in developers' comments. However, our corpus still contains popular projects such as Lucene and Hadoop.

B. Experiment Design

In order to evaluate the impact of affective metrics on the issue fixing time we designed our experiment as follows. We built a logistic regression model[16] for classifying the issue fixing time as short or long based on a set of independent variables characterising Jira issues [30]. The output of the logistic regression model, given the metric values of a particular issue, is the probability of the issue to be fixed in a short or long time. One then needs to select a threshold probability above which the logistic outcome is interpreted as "long fixing time". Since the logistic regression model has a binary output, we had to transform the numeric issue fixing times of Jira into

[10] http://wordnet.princeton.edu/

[11] An affective-label is a label assigned to a word and its synonyms that indicates the emotional state of that word. For example, the word "sad" has X and Y as affective label, see http://wndomains.fbk.eu/wnaffect.html

[12] http://en.wikipedia.org/wiki/Hyperparameter_optimization

[13] http://en.wikipedia.org/wiki/F1_score

[14] http://en.wikipedia.org/wiki/Bootstrapping_(statistics)

[15] https://www.atlassian.com/software/jira

[16] http://en.wikipedia.org/wiki/Logistic_regression

Project	# issues	# comments	# developers	issues' average # comments	issues' average # commenters
HBase	9353	91016	951	9.73	2.93
Hadoop Common	7753	61958	1243	7.99	2.98
Derby	6101	52668	675	8.63	2.74
Lucene Core	5111	50152	1107	9.81	2.96
Hadoop HDFS	4941	42208	757	8.54	2.9
Cassandra	6271	41966	1177	6.69	2.54
Solr	5086	41695	1590	8.19	3.18
Hive	5124	39002	850	7.61	2.8
Hadoop Map/Reduce	4747	34793	875	7.32	2.74
Harmony	6291	28619	316	4.54	2.22
OFBiz	5098	25694	578	5.04	2.23
Infrastructure	6804	25439	1362	3.60	1.95
Camel	6147	24109	908	3.92	1.76
ZooKeeper	1606	16672	495	3.32	1.87

TABLE IV: Statistics of the selected projects (developers correspond to the Jira users that are involved in a project, i.e. *committers*, *issue reporters* and *comment posters*.)

a binary value, with 1 meaning that the issue fixing time will be *longer* than the issue fixing time median, and zero meaning *shorter* than the median.

As independent variables, we considered a set of *control metrics* as control variables for our case study, and a set of *affective metrics* as controlled variables. Table V shows the considered metrics. The controlled variables are the issue characteristics proposed by Giger et al. [31] as listed in the first half of Table V. These control metrics cover all dimensions of Giger et al.'s work [31]. In particular, Giger et al. found that assignee and reporter experience have the strongest influence on bug fixing time. The second set of independent variables, i.e., the controlled variables, are different variations of the three affectiveness metrics of Section III that we deemed related to issue fixing time (these variations are non-exhaustive).

Instead of building one model with all metrics at once, we used a hierarchical modelling approach where one metric at a time is added, a model is built, then the model is compared using an ANOVA test to the previous model (without that metric) to check whether the addition of the metric leads to a statistically significant improvement of the model. We then considered in our final model, only those metrics that were significant, i.e., those metrics with a *p-value* <0.01 (marked with ** or ***). The significant metrics are shown in bold in Table VI.

Finally, we evaluated the impact of each metric in the model as shown in Figure 2, using the general approach proposed by Shihab et al. [30]:

- First, we gave as input to the logistic regression model the median values of each metric, since those values represent a "common" value for the metric. The corresponding output probability is called *baseline output*.
- One metric at a time, we add one a standard deviation to the considered *metric k* leaving all other metrics unchanged on their median values. This yields a probability that we call *metric k output*.
- For each *metric k*, we calculated the relative increase of the *metric k output* relative to the *baseline output*, i.e., $(\textit{metric k output} - \textit{baseline output})/\textit{baseline output}$.
- We can then compare the relative increase of each metric to determine the metric with the largest impact (relative increase), as well as the sign of the increase (positive/negative), independent of the unit/type of the metric. For categorical metrics, we used the mode (most frequently used value) instead of the median.

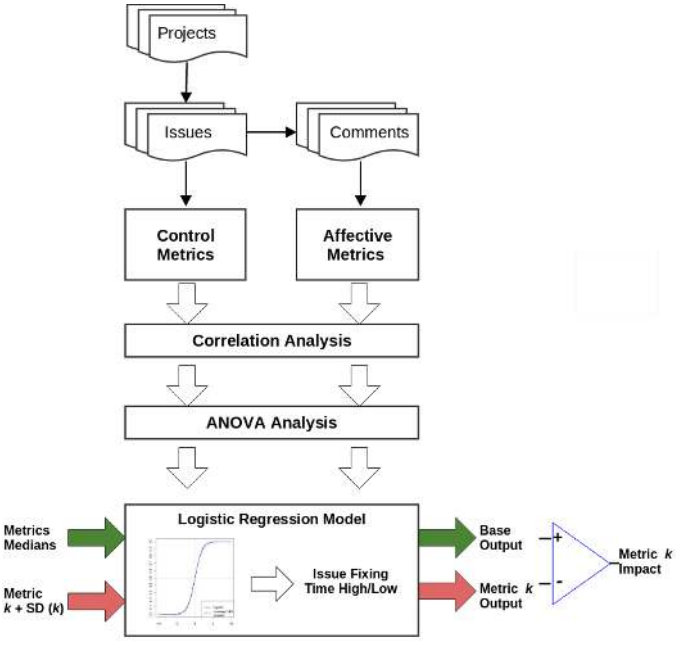

Fig. 2: Experiment Schema

V. Results

A. RQ1: Are emotions, sentiment and politeness correlated to each other?

Motivation. Our final goal is to understand the impact of affectiveness on the issue fixing time. For this purpose, we build a regression model using affective metrics in RQ2. However, since all affective metrics measure something about the feelings of stakeholders we first need to understand whether

Control Metrics			
metric	Type	Range	Description
reporter previous # comments	Number	>=0	# comments previously posted by the issue reporter
assignee previous # comments	Number	>=0	# comments previously assigned to the issue assignee
issue priority	Category	TRIVIAL..CRITICAL	The priority assigned to the issue (Major, Minor, Critical etc.)
issue type	Category	BUG..NEW_FEATURE	The issue maintenance type (Bug, New Feature, Task etc.)
issue # watchers	Number	>=0	The number of Jira users watching the issue
issue # developers	Number	>=0	The total number of Jira users that commented on an issue, including reporter and assignee
issue # status changes	Number	>=0	The total number of times an issue has been changed (such as changing status, resolution, type, priority etc.)
issue # comments	Number	>=0	The total number of comments posted on an issue report
Affective Metrics			
metric	Type	Range	Description
issue avg sentiment	Number	[0,1]	The average sentiment expressed in the issue comments
issue avg politeness	Number	[0,1]	The average politeness expressed in the issue comments
issue love comments proportion	Proportion	%	The percentage of issue comments expressing love emotion
issue joy comments proportion	Proportion	%	The proportion of issue comments expressing joy emotion
issue sadness comments proportion	Proportion	%	The proportion of issue comments expressing sadness emotion
issue anger comments proportion	Proportion	%	The proportion of issue comments expressing anger emotion
issue title sentiment	Number	[0,1]	The sentiment expressed in an issue's title
issue title politeness	Number	[0,1]	The politeness expressed in an issue's title
issue first comment sentiment	Number	[0,1]	The sentiment expressed in the issue's first comment
issue first comment politeness	Number	[0,1]	The politeness expressed in the issue's first comment
issue last comment sentiment	Number	[0,1]	The sentiment expressed in the issue's last comment
issue last comment politeness	Number	[0,1]	The politeness expressed in the issue's last comment

TABLE V: Metrics used in our study

sentiment, emotion and politeness are really independent measures, or if there is overlap between them, in which case we should filter out some of the metrics.

Approach. In order to evaluate the correlation between the considered affective metrics, we measured the sentiment, emotions and politeness of developer comments using metrics in Table V, considering only issues with at least two comments. For each issue, we used the love/joy/sadness/anger comment proportion, average politeness and sentiment *per* issue considering all comments posted on the same issue. We first calculated for each issue comment a politeness value according to the following rules:

- Value of +1 for those comments marked as polite by the tool;
- Value of 0 for those comments marked as neutral (confidence level<0.5);
- Value of -1 for those comments marked as impolite.

Then we averaged the assigned politeness across all comments, obtaining a number in a range from -1 to 1. We finally normalize the average issue politeness in a range from 0 to 1.

Similar to the average issue politeness, we evaluated the average issue sentiment measuring for each comment of an issue, the sentiment using SentiStrenght. As described in Sec. III-B, SentiStrenght yields a value in a range from -5 to 5. Averaging all comments' sentiments we obtain the issue average sentiment as a number in the range from -5 to 5, which we normalize again in a range from 0 to 1. After normalization, issue with average sentiment and politeness 0 means respectively extremely impolite and negative (sentiment), 0.5 means neutral politeness and sentiment and 1 extremely polite and positive (sentiment).

We calculated the emotion proportions, average sentiment and politeness of about 560K comments (about 68K issues) then computed the Pearson correlation coefficient among all the considered metrics, except for the non-numeric issue type and priority [31]. As is commonly done, we considered *weak* a correlation less than 0.4, *moderate* a correlation from 0.4 to 0.7, and *strong* a correlation greater than 0.7.

Findings. Weak correlation exists between *issue average politeness* and *issue first comment politeness*, and between *issue last comment politeness* and *issue last comment sentiment* . Table VII shows the correlations larger than 0.3. The affective metrics have a maximum weak correlation of 0.36 between the *issue average politeness* and *issue first comment politeness*. Some of the control metrics instead have a moderate to strong correlation with a maximum value of

Feature	z-value	p-value
assignee # previous comments	-19.322	<2e-16 ***
reporter # previous comments	-0.933	<2e-16 ***
issue priority:Critical	7.194e-02	5.94e-09 ***
issue priority:Major	12.263	< 2e-16 ***
issue priority:Minor	14.200	< 2e-16 ***
issue priority:Trivial	6.687	2.28e-11 ***
issue type:Bug	-1.230	0.218550
issue type:Improvement	-0.872	0.383073
issue type:New Feature	-0.415	0.677798
issue type:Sub-task	-1.050	0.293538
issue type:Task	-0.621	0.534872
issue type:Test	-1.277	0.201539
issue type:Umbrella	1.136	0.256108
issue type:Wish	0.049	0.961256
issue # watchers	3.590	0.000330 ***
issue number of developers	27.559	< 2e-16 ***
issue number of changes	40.329	< 2e-16 ***
issue avg sentiment	-5.594	2.22e-08 ***
issue avg politeness	11.485	< 2e-16 ***
issue avg love	-16.329	< 2e-16 ***
issue avg joy	-9.099	< 2e-16 ***
issue avg sadness	14.388	< 2e-16 ***
issue avg anger	-0.212	0.831741
issue title sentiment	2.884	0.003922 **
issue title politeness	3.512	0.000444 ***
issue first comment sentiment	1.676	0.093723 .
issue first comment politeness	2.108	0.035053 *
issue last comment sentiment	4.839	1.30e-06 ***
issue last comment politeness	-9.843	< 2e-16 ***

TABLE VI: Coefficient and p-values for the metrics of the logistic regression model. Metrics in bold are significant to the model.

0.7 between *issue # developers* and *issue # comments*. Given the strong correlation between *issue # developers* and *issue # comments*, we considered all metrics except *issue # comments* in the remainder of our analysis.

B. RQ2: Can developer affectiveness explain the issue fixing time?

Motivation. Productivity is an important factor for a software organization to be successful, i.e., achieving shorter time to market, for this reason understanding the factors that impact software productivity is crucial during software development. Although there are many factors that impact the issue fixing time [31], there is little information about the impact of developers's affectiveness on the issue fixing time. In this RQ, we investigate a possible relation between the affective metrics for emotions, politeness, and sentiment with issue fixing time.

Approach. As explained in Section IV-B, we used the metrics in Table V to build a logistic regression model for explaining the issue fixing time.

Findings. Affective metrics are significant for the explanation of the issue fixing time. Our logistic regression model has a Precision of 0.67 and Recall of 0.671 against respectively 0.319 and 0.56 for the ZeroR model. Table VI shows how significant the metrics are for the logistic regression model. We considered significant all metrics with a *p-value*<0.01. As expected, the control metrics such as the *issue priority*, *issue reporter/assignee previous comments* and the *issue number of developers/changes* are significant. However, more interesting is that affective metrics such as the *issue percentage of emotion x* and *issue average politeness/sentiment* are significant.

	issue average politeness	reporter # previous comments	issue # watchers	issue last comment sentiment	issue # changes	issue # developers
assignee # previous comments	n.s	**0.49**	n.s	n.s	n.s	n.s
issue first comment politeness	**0.36**	n.s	n.s	n.s	n.s	n.s
issue last comment politeness	n.s	n.s	n.s	**0.36**	n.s	n.s
issue # developers	n.s	n.s	**0.55**	n.s	**0.48**	n.s
issue # comments	n.s	n.s	**0.48**	n.s	**0.67**	**0.7**

TABLE VII: Weak and moderate correlations in our dataset (RQ1)

To calculate the total performance of the model, we chose only the metrics from Table VI that are significant (*p-value*<0.01), then built a final logistic regression classifier. Table VIII shows a comparison between the classification performance of our logistic regression model and a ZeroR classifier. The latter is a baseline model that always answers the same output ("long"), and often is used as a baseline to compare a model to (models performing worse are not worth the effort). By definition, the ZeroR model has perfect recall for "Long", but its precision suffers, and recall for the "Short" class is zero, which results in an average weighted precision and recall (across both classes) of 0.319 and 0.565 respectively. On the other hand, our model obtains good precision and recall for both classes, resulting in a much higher average precision and recall. The precision, recall and AUC of our model are comparable to those obtained by Giger et al. [31] and are better than the precision and recall of the ZeroR classifier. AUC is the area under the receiver operating characteristic curve. It can be interpreted as the probability that, when randomly selecting a positive ("Long") and a negative ("Short") example the model assigns a higher score to the positive example [32]. For a random model, this probability would be 0.5, which is the AUC obtained for the ZeroR model in our case. Our logistic model obtains an AUC value higher than 0.5, better than random. We compared the logistic regression model with and without the affective

Classifier	Class	Precision	Recall	F1	AUC
ZeroR	Short	0	0	0	0.5
	Long	0.565	1	0.722	
	Weighted Avg.	0.319	0.565	0.408	
Logistic **without** affective metrics	Short	0.602	0.6	0.601	0.715
	Long	0.69	0.7	0.695	
	Weighted Avg.	0.655	0.656	0.655	
Logistic **with** affective metrics	Short	0.626	0.607	0.616	0.734
	Long	0.704	0.72	0.712	
	Weighted Avg.	0.67	0.671	0.67	

TABLE VIII: Logistic regression model performance

metrics using the ANOVA analysis (using a Chi-squated test) and we obtained a p-value of **2.2e-16 *** ** confirming that the two models are statistically significantly different and that by adding the affective metrics to our model, precision, recall and AUC are all increased.

C. RQ3: Which affective metrics best explain issue fixing time?

Motivation. We found that the affective metrics are significant for the logistic regression model that we built, as shown in Table VI. Since not all are equally influential in a regression model, we now are interested in quantifying which metrics have the strongest link with issue fixing time. In particular, are affectiveness measures as important as traditional issue-related measures?

Approach. In order to understand the impact of affective metrics, we evaluated the impact of each metric on the logistic regression model as described in Sec. IV-B.

Findings. Sentiment and emotions such as *JOY* and *LOVE* reduce the resolution time whereas sentiment and emotions such as *SADNESS* increase the issue resolution time. Issue average politeness increases the issue fixing time.

Table IX shows the relative increase in the logistic regression *baseline output* when fixing all metrics but one on their median values and adding one standard deviation to one metric's median value. The two control metrics *issue number of developers* and *issue number of changes* have the highest impact (>100%): the more developers involved or changes being made, the longer the fixing time. In contrast, the *issue assignee/reporter previous comments*, which are a measure of developer experience, have a negative impact on the issue fixing time, i.e., the more the issue's assignee or reporter is experienced the more likely the issue fixing time will be shorter.

Apart from the above control variables, some affective metrics also have a significant impact. The more polite an issue's last comment is, the more likely the issue fixing time was shorter. Similarly, the issue average sentiment impact is -10.52%, which means that the more positive the average sentiment is, the faster an issue is fixed. *JOY* and *LOVE* have an impact of -26.42% and -50.19% respectively, whereas the *SADNESS* emotion has an impact of 38.49%. SADNESS is linked with longer issue fixing time, whereas JOY and LOVE are linked to shorter fixing times.

Feature	% of increment of logistic reg. output when the adding one SD
issue # changes	192.09%
issue # developers	134.23%
issue average politeness	49.76%
% sadness comments	38.49%
issue last comment sentiment	13.72%
watchers	10.92%
issue reporter prev. comments	-9.18%
issue avg sentiment	-10.52%
% joy comments	-26.42%
issue last comment politeness	-29.10%
% love comments	-50.19%
assigne # previous comment	-54.45%

TABLE IX: Metrics impact on issue fixing time. Affective metrics are highlighted in bold.

Similar to the *% of sadness comments*, the *issue's average politeness* increases the likelihood of a long issue fixing time by 49.76%. This result is somehow unexpected. One would expect that the more developers communicate in a polite way, the more they are able to be productive. We discuss the impact of politeness in the next section.

VI. Discussion

This section investigates in more detail the role played by the *issue's average politeness*, since it is somehow unexpected that the issue average politeness is related to longer issue fixing time. To enable a deeper analysis, we distinguished between three groups of issues:

- *High-Politeness*: issues with average politeness 1.
- *Medium-Politeness*: issues with average politeness in the range]0,1[. This category corresponds to issues that are more or less neutral.
- *Low-Politeness*: issues with average politeness 0.

We use box plots and hexbin plots [17] to understand how the issue fixing time is distributed across these three categories.

Figure 3 shows the box plot in logarithmic scale of the issue fixing time for the three categories of average politeness considered. Issues with *Low-Politeness* and *High-Politeness* have the shortest fixing time, containing respectively 38.8% and 10.4% of the total number of issues. This finding is further confirmed by the hexbin plot of Figure 4, where we can see that for *Medium-Low-Politeness* the majority of issues are shifted up towards higher values of issue fixing time compared to Low- and High-Politeness. In other words, the extreme cases of politeness, both in positive and negative

[17] A hexagon bin plot is a kind of scatterplot where instead of individual dots for each data point, all data points in a hexagonal area are collapsed and the color of the hexagon shows how many data points are in that area. Hexbin plots are very informative in cases where many data points would overlap and one would not know how many points are overlapping.

sense, are linked with faster fixing time compared to more neutral cases. Such a non-linear link between an independent variable and the dependent variable cannot be captured by a logistic model, which is why the model suggested in RQ2 that higher politeness is linked with longer issue fixing time (since the median fixing time of High-Politeness is slightly higher than for Low-Politeness). This finding for High-Politeness confirms the findings of Ortu et al. [23].

What is still unclear is why the extreme cases have lower fixing times. One plausible reason for Low-Politeness issues (which captures 38.8% of all issues, i.e., the majority of extreme politeness cases) is such issues quickly conclude an issue because of the negative or positive tone of the comments. Alternatively, issues of the extreme politeness cases (positive and negative) might have attracted more participants, resulting in more discussion and hence longer fixing time.

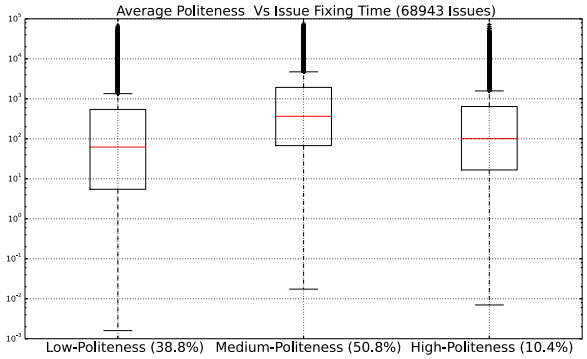

Fig. 3: Average Issue Politeness versus Issue Fixing Time Boxplot

Figure 5 shows that *Low-Politeness* issues indeed have the lowest number of sentences with Medium- and High-Politeness containing most of the sentences. In other words, negative discussions seem to conclude with less discussion.

Furthermore Figure 6 shows the box plot of *issue # developers* for the three categories of average politeness. Here, the extreme politeness cases both have the lowest number of participants, with a median value of 2 developers. *Medium-Politeness* issues have a median value of the *issue # developers* of 4. Taken together, issues with extreme politeness involve less developers and (at least for negative politeness) have shorter comments, both of which could provide part of the reason why their issue fixing time is shorter. More research is needed to fully understand these observations.

VII. Threats To Validity

Threats to internal validity concern confounding factors that can influence the obtained results. We assume a causal relationship between a developer's emotional state and what he or she writes in issue report comments, based on empirical evidence (in another domain) [33]. Moreover, since developer communication has as first goal information sharing, removing or disguising emotions *may* make comments less meaningful

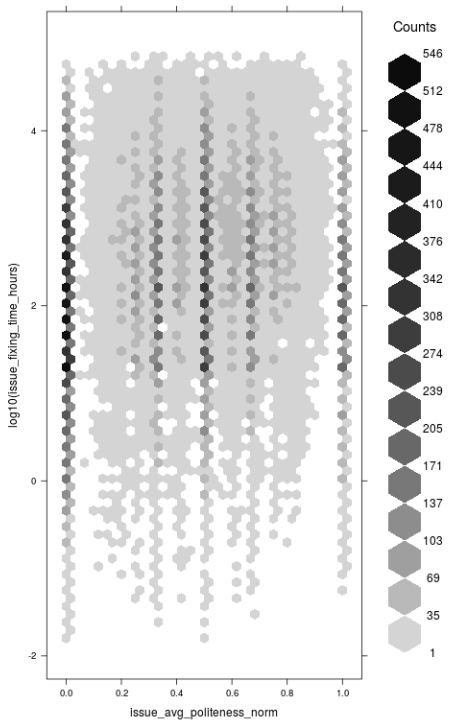

Fig. 4: Average Issue Politeness versus Issue Fixing Time Hexbin Plot

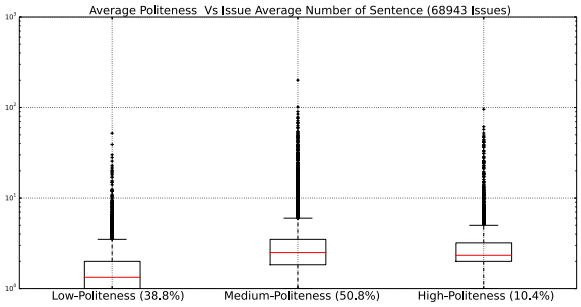

Fig. 5: Distribution of Average Politeness versus Average Number of Sentences for the three groups of issues.

and cause misunderstanding. Since the comments used in this study were collected over an extended period from developers not aware of being monitored, we are confident that the emotions we mined are genuine. This is also why we could not involve the authors of the comments in our study. That said, we do not claim any causality between any of our metrics and the issue fixing time. We mainly built an explanatory model to understand the characteristics of issues with short and long fixing time.

Threats to construct validity focus on how accurately the observations describe the phenomena of interest. Mining of emotions from textual issue report comments presents difficulties due to ambiguity and subjectivity. To reduce these threats,

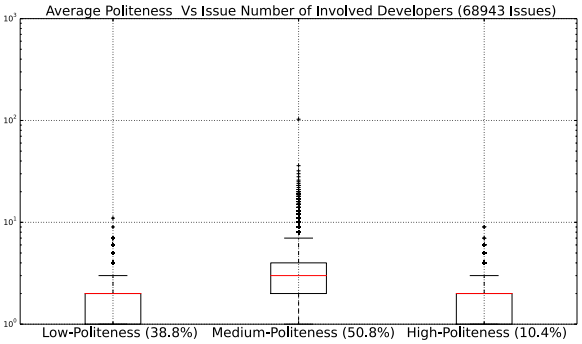

Fig. 6: Distribution of number of developers versus Politeness for the three groups of issues.

the authors adopted Parrott's framework as a reference for emotions. Finally, to avoid bias due to personal interpretation, during the annotation of 4000 sentences for the training corpus of the emotion classifier, each sentence was analyzed by at least two raters. Furthermore the affectiveness measures are approximations and cannot 100% correctly identify the correct affective context, given the challenges of natural language and subtle phenomena like sarcasm. To deal with these threats, we used state-of-the-art tools like SentiStrength, the tool of Desmedt et al. [28] and Danescu et al.'s politeness tool, in addition to our own emotion classifier.

Threats to external validity correspond to the generalizability of our experimental results [34]. In this study, we manually analyze a sample of 4000 sentences of comments from issue reports belonging to 14 open source projects. We consider the projects as a representative sample of the universe of open source software projects, with different development teams and satisfying different customers' needs. Replications of this work on other open source systems and on commercial projects are needed to confirm our findings.

Threats to reliability validity correspond to the degree to which the same data would lead to the same results when repeated. This research is the first attempt to manually investigate different measures of affectiveness from issue reports, and their impact on the issue fixing time, hence no ground truth exists to compare our findings. We defined the ground truth through agreement or disagreement of the raters for measuring emotions and existing tools provided for measuring sentiment and politeness.

This study is focused on text written by developers *for* developers. To correctly depict the affectiveness embedded in such comments, it is necessary to understand the developers' dictionary and slang. This assumption is supported by Murgia et al. [9] for measuring emotions. We are confident that the tools used for measuring sentiment and politeness are equally reliable in the software engineering domain as in other domains.

VIII. Conclusion

Human *Affectiveness* such as the emotional state of a person influences human behaviour and interaction. Software development is a collaborative activity and thus it is not exempt from such influence. Affective analysis, e.g., measuring emotions, sentiment and politeness, applied to developer issue reports, can be useful to identify and monitor the mood of the development team, allowing project leaders to anticipate and resolve potential threats to productivity (especially in remote team settings), as well as to discover and promote factors that bring serenity and productivity in the community. This study is a first attempt to highlight the impact of developer affectiveness on productivity in the form of issue fixing time.

First, we showed that the three affective metrics, i.e., emotions, sentiment and politeness, are independent, showing a weak correlation of at most 0.36, in contrast to some of the control metrics who obtained a moderate to strong correlation among themselves of at most 0.7.

Then, we showed how affectiveness metrics statistically improve an explanation model of issue fixing time compared to a model based on control metrics. The 4th, 5th and 6th most important metrics in the model correspond to *% of love comments* (-50.19%), *issue average politeness* (+49.76%) and *% of sadness comments* (+38.39%). In other words, comments containing *JOY* and *LOVE* emotions have shorter issue fixing time, while comments containing *SADNESS* emotion have a longer fixing time. Although we found that the politeness of the last comment has a shorter issue fixing time, it is *unexpected* that less polite comments are linked with shorter fixing time.

After investigation we found that for about the 50% issue reports with extreme politeness (polite and impolite) have shorter issue fixing time. Those reports tend to only have a median number of 2 developers discussing the issue, and the negative issues have the lowest number of sentences in the comments. Whereas Ortu et al. [23] also found that issues with positive politeness have lower fixing time, the fact that issues with negative politeness have the same characteristics is a novel finding in our paper.

References

[1] T. U. Grund, "Network structure and team performance: The case of english premier league soccer teams," Social Networks, vol. 34, no. 4, pp. 682–690, 2012.

[2] J. H. Fowler, N. A. Christakis et al., "Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study," Bmj, vol. 337, p. a2338, 2008.

[3] B. Curtis, H. Krasner, and N. Iscoe, "A field study of the software design process for large systems," Communications of the ACM, vol. 31, no. 11, pp. 1268–1287, 1988.

[4] E. Guzman and B. Bruegge, "Towards emotional awareness in software development teams," in Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering. ACM, 2013, pp. 671–674.

[5] A. Begel, N. Nagappan, C. Poile, and L. Layman, "Coordination in large-scale software teams," in Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering. IEEE Computer Society, 2009, pp. 1–7.

[6] L. F. Capretz and F. Ahmed, "Making sense of software development and personality types," IT professional, vol. 12, no. 1, pp. 6–13, 2010.

[7] I. Rowe, "Civility 2.0: a comparative analysis of incivility in online political discussion," Information, Communication & Society, vol. 18, no. 2, pp. 121–138, 2015.

[8] I. Steinmacher, T. U. Conte, M. Gerosa, and D. Redmiles, "Social barriers faced by newcomers placing their first contribution in open source software projects," in Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing, 2015, pp. 1–13.

[9] 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 Working Conference on Mining Software Repositories. ACM, 2014, pp. 262–271.

[10] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries et al., "Manifesto for agile software development," 2001.

[11] D. L. Parnas, "Software engineering or methods for the multi-person construction of multi-version programs," pp. 225–235, 1975.

[12] ——, "Software engineering: multi-person development of multi-version programs," 2011.

[13] W. Ke and P. Zhang, "The effects of extrinsic motivations and satisfaction in open source software development," Journal of the Association for Information Systems, vol. 11, no. 12, pp. 784–808, 2010.

[14] J. A. Roberts, I.-H. Hann, and S. A. Slaughter, "Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the apache projects," Management science, vol. 52, no. 7, pp. 984–999, 2006.

[15] A. P. Brief and H. M. Weiss, "Organizational behavior: Affect in the workplace," Annual review of psychology, vol. 53, no. 1, pp. 279–307, 2002.

[16] A. Erez and A. M. Isen, "The influence of positive affect on the components of expectancy motivation." Journal of Applied Psychology, vol. 87, no. 6, p. 1055, 2002.

[17] E. Kaluzniacky, Managing psychological factors in information systems work: An orientation to emotional intelligence. IGI Global, 2004.

[18] R. Feldt, R. Torkar, L. Angelis, and M. Samuelsson, "Towards individualized software engineering: empirical studies should collect psychometrics," in Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering. ACM, 2008, pp. 49–52.

[19] 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. IEEE Computer Society, 2007, p. 23.

[20] B. Bazelli, A. Hindle, and E. Stroulia, "On the personality traits of stackoverflow users," in Software Maintenance (ICSM), 2013 29th IEEE International Conference on. IEEE, 2013, pp. 460–463.

[21] P. Tourani, Y. Jiang, and B. Adams, "Monitoring sentiment in open source mailing lists - exploratory study on the apache ecosystem," in Proceedings of the 2014 Conference of the Center for Advanced Studies on Collaborative Research (CASCON), Toronto, ON, Canada, November 2014.

[22] M. N. Gómez, S. T. Acuña, M. Genero, and J. A. Cruz-Lemus, "How does the extraversion of software development teams influence team satisfaction and software quality?: A controlled experiment," International Journal of Human Capital and Information Technology Professionals (IJHCITP), vol. 3, no. 4, pp. 11–24, 2012.

[23] M. Ortu, G. Destefanis, M. Kassab, S. Counsell, M. Marchesi, and R. Tonelli, "Would you mind fixing this issue? an empirical analysis of politeness and attractiveness in software developed using agile boards," in XP2015, Helnsiki. Springer, 2015, p. in press.

[24] C. Danescu-Niculescu-Mizil, M. Sudhof, D. Jurafsky, J. Leskovec, and C. Potts, "A computational approach to politeness with application to social factors," in Proceedings of ACL, 2013.

[25] 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.

[26] D. Pletea, B. Vasilescu, and A. Serebrenik, "Security and emotion: sentiment analysis of security discussions on github," in Proceedings of the 11th Working Conference on Mining Software Repositories. ACM, 2014, pp. 348–351.

[27] C. Strapparava, A. Valitutti et al., "Wordnet affect: an affective extension of wordnet." in LREC, vol. 4, 2004, pp. 1083–1086.

[28] T. De Smedt and W. Daelemans, "Pattern for python," The Journal of Machine Learning Research, vol. 13, no. 1, pp. 2063–2067, 2012.

[29] H. A. Elfenbein and N. Ambady, "On the universality and cultural specificity of emotion recognition: a meta-analysis." Psychological bulletin, vol. 128, no. 2, p. 203, 2002.

[30] E. Shihab, Z. M. Jiang, W. M. Ibrahim, B. Adams, and A. E. Hassan, "Understanding the impact of code and process metrics on post-release defects: a case study on the eclipse project," in Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. ACM, 2010, p. 4.

[31] E. Giger, M. Pinzger, and H. Gall, "Predicting the fix time of bugs," in Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering. ACM, 2010, pp. 52–56.

[32] J. A. Nevin, "Signal detection theory and operant behavior: A review of david m. green and john a. swets' signal detection theory and psychophysics. 1," Journal of the Experimental Analysis of Behavior, vol. 12, no. 3, pp. 475–480, 1969.

[33] 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.

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