Do Developers Feel Emotions? 
An Exploratory Analysis of Emotions in Software Artifacts

Alessandro Murgia 
University of Antwerp, Belgium
alessandro.murgia@ua.ac.be

Parastou Tourani 
École Polytechnique de Montréal, Canada
{parastou.tourani,bram.adams}@polymtl.ca

Bram Adams 
University of Cagliari, Italy
marco.ortu@diee.unica.it

Marco Ortu 

ABSTRACT
Software development is a collaborative activity in which developers interact to create and maintain a complex software system. Human collaboration inevitably evokes emotions like joy or sadness, which can affect the collaboration either positively or negatively, yet not much is known about the individual emotions and their role for software development stakeholders. In this study, we analyze whether development artifacts like issue reports carry any emotional information about software development. This is a first step towards verifying the feasibility of an automatic tool for emotion mining in software development artifacts: if humans cannot determine any emotion from a software artifact, neither can a tool. Analysis of the Apache Software Foundation issue tracking system shows that developers do express emotions (in particular gratitude, joy and sadness). However, the more context is provided about an issue report, the more human raters start to doubt and nuance their interpretation of emotions. More investigation is needed before building a fully automatic emotion mining tool.

Categories and Subject Descriptors
D.2.9 [Management]: Programming teams; H.5.3 [Group and Organization Interfaces]: Web-based interaction

General Terms
Human Factors, Measurement

Keywords
Emotion Mining, Issue Report, Empirical Software Engineering

1. INTRODUCTION
In July 2013, the Linux kernel mailing list was shaken up by an agitated discussion between Linus Torvalds and a senior developer [6]: “I am serious about this. Linus, you’re one of the worst offenders when it comes to verbally abusing people and publicly tearing their emotions apart.” Other people joined her, noting “scolding people [...] is not likely to encourage people to want to become senior developers” and “Thanks for standing up for politeness/respect. If it works, I’ll start doing Linux kernel dev. It’s been too scary for years.” On the other hand, Linus defended himself, claiming “not telling people clearly enough that I don’t like their approach, they go on to re-architect something, and get really upset when I am then not willing to take their work.”

This example suggests that a rational view of software development only provides a partial picture of how stakeholders really behave: the developers may underperform if they do not feel safe and happy. Positive emotions like happiness help people to be more creative [15], which is essential for successful software design [7]. If not, fear, or absence of courage, could refrain developers from changing or refactoring their code [1]. These effects of emotions are similar to other domains, where people have found that feelings and emotions dictate to a large extent our actions and decisions [33]. For example, consumer opinions on retailer sites influence potential buyer decisions [31]. The mood of people, evaluated through tweets, correlates with changes in the activity of buying or selling in the stock market [5].

Since even the most talented developer could underperform and eventually leave the project just because she is unhappy with her environment or colleagues it is important to support managers and project leads in detecting emotions in their team. Mining emotions from discussion boards is relevant when face-to-face meetings are not feasible or efficient. For example, in distributed development (both open and closed source), projects have almost no personal interaction except for sporadic conference calls. In such environments, gauging emotions across geographical locations is essential for managers to become immediately aware of new problems and be able to take typical managerial action to defuse the situation. They can then organize conference calls focused on the reasons of a developer’s unhappiness (e.g., sexual harassment), or schedule a special meeting if developers suddenly became anxious (e.g., for issue due to an OS update).

Since awareness of emotions in a team currently is a manual activity, we want to use messages posted on a project’s public discussion boards (i.e., one of the major means of communication) to mine developer emotions. In particular, as a first step towards building a tool for automatic emotion mining, we performed a pilot study (with the four authors) and a full user study (with 16 participants) to determine whether emotions can actually be detected from typical software maintenance artifacts like issue reports, and, if so, whether humans can...
actually agree on the emotions identified and how much information (context) they need for that. Without such agreement, automated tools would not make sense to build. In particular, we analyzed a significant sample of 792 developer comments (400 in the pilot study, 392 in the full study) of the Apache projects using Parrott’s emotional framework [30] to answer the following research questions:

**RQ1)** Can human raters agree on the presence or absence of emotions in issue reports?

We found that raters agree the most on the absence of an emotion, followed by the presence of love (i.e., gratitude) and (less strongly) joy and sadness. Having more than two raters does not significantly improve agreement.

**RQ2)** Does context improve the agreement of human raters on the presence of emotions in issue reports?

We found that providing human raters with more context about an issue seems to cause doubt (i.e., nuances) instead of more confidence in the identified emotions. Having more than two raters makes ratings more robust to this.

To our knowledge, this is the first feasibility study of emotion mining in development artifacts like issue reports. Based on our findings, issue comments have potential as data source for emotion mining, yet more work is needed to fully understand the role of context on the identification of emotions.

In the remainder of this paper, we first describe the background notions for emotion mining (Section 2). Next, we describe the experimental setup (Section 3), followed by a discussion up front of mined developer emotions (Section 4). We then address the two research questions (Section 5) and discuss our findings (Section 6). After a discussion of the threats to validity (Section 7) and related work (Section 8), we finish with conclusions (Section 9).

2. **BACKGROUND**

This section provides background about emotion mining, the Parrott emotional framework, and the development artifacts (i.e., issue reports) studied in this paper.

2.1 Emotion Mining

Emotion mining tries to identify the presence of human emotions like joy or fear from text, voice and video artifacts produced by humans. As such, it is different from sentiment analysis, which instead evaluates a given emotion as being positive or negative [29]. The field of sentiment analysis as a whole is currently moving towards emotion mining, since this provides more detailed insight into the behavior of people [39]. Since these research areas affect the decision-making process of people [29], a diverse range of actors, from marketing departments and investors to politicians make use of their techniques.

In software engineering, emotion mining applied to text artifacts could be used to provide hints on factors responsible for joy and satisfaction amongst developers (e.g., new release), or fear and anger (e.g., deadline or a recurring bug). Moreover, it provides a different perspective to interpret productivity and job satisfaction. Since several studies show that it is possible to “contract” emotions from others through computer-mediated communication systems [18, 21], development artifacts like mailing lists or the discussion board of an issue tracking system could be a promising source for mining developer emotions during software evolution.

2.2 Parrott’s Framework

Emotion is a “psychological state that arises spontaneously rather than through conscious effort and is sometimes accompanied by physiological changes” [22]. General types of emotions are joy, sadness, anger, surprise, hate and fear. However, many other categories and sub-categories can be identified. Since there is not one standard emotion word hierarchy, many studies in the cognitive psychology domain [36] have focused on research about emotions, resulting in various proposals for categorizing emotions [35, 32, 30].

One of the more recent classifications of emotions is Parrott’s framework [30], which classifies human emotions into a tree structure with 3 levels, as is shown in Table 1. Each level refines the granularity of the previous level, making abstract emotions more concrete. For example, level-1 of this classification consists of six primary-emotions, i.e., love, sadness, anger, joy, surprise and fear.

The concise and intuitive nature of the primary emotions make Parrott’s classification easy to understand by different stakeholders. In particular, the classification is not just aimed at the people rating a particular artifact as describing a particular emotion, but also appeals to people like team leads trying to benefit from the emotional classification to understand the emotions of their team members. In this paper, we only consider the six primary emotions, but in future work we plan to extend our results to secondary and tertiary emotions for the most popular primary emotions.

2.3 Issue Tracking System

The issue tracking system is a repository used by software companies to organize software maintenance and evolution. Such repositories, like Jira\(^1\), provide a shared environment where team members can submit and discuss issues (e.g., bugs and feature requests), ask for advice and share opinions useful for maintenance activities or design decisions. These

\(^1\)https://www.atlassian.com/software/jira
discussions reveal how committers feel towards a bug, feature, project or even other members of the community.

An issue report is characterized by standard fields useful for its processing (e.g., priority, status etc.). In particular, the report contains a list of comments used by developers to discuss and share ideas about the issue resolution. These comments (see for example Figure 1) are the ones analyzed in our study.

### 3. EXPERIMENTAL SETUP

In our experiment, we analyze emotions reported in issue reports extracted from open source systems. This section discusses the data set used in our analysis, the general procedure used to rate issue report comments, and finally how pilot and full study where organized.

#### 3.1 Dataset

We mined the issue repository of the Apache software foundation\(^2\), since such a type of repository contains a significant amount of information related to a project’s dynamics [20]. The mined issue reports belong to 117 open source projects, ranging from large, long-lived projects such as Tomcat\(^3\), and Lucene\(^4\), to smaller projects like RAT\(^5\). For the experiment, we use a random sample of the project’s comments, so the larger the project the higher the chance that we analyzed some of its comments. Given this wide range of projects, we believe that this data source provides a representative overview of open source issue reports.

We parsed Apache’s Jira-based repository in July 2013, fetching all the issue reports since the 19th of October 2000. For each issue report, we extracted the developers’ comments, as well as the standard issue report fields mentioned in Section 2.3. Table 2 reports the statistics for our dataset.

Since an issue comment can consist of multiple sentences, and an issue report of multiple issue comments, we decided to perform our analyses at the level of issue comments to avoid a too coarse granularity. Given the large number of issue comments, we sampled enough issue comments to obtain a confidence level of 95% and confidence interval of 5%. This means that a proportion of X% in our sample of issue comments actually corresponds to X ± 5% in the population of issue comments. For this reason, we obtained samples of 392 or more comments (out of 271,416 comments).

Table 2: Apache Software foundation statistics.

<table>
<thead>
<tr>
<th>Projects</th>
<th>Issues</th>
<th>Comments</th>
<th>Users</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>117</td>
<td>81,523</td>
<td>217,416</td>
<td>20,537</td>
<td>10/2000</td>
<td>07/2013</td>
</tr>
</tbody>
</table>

Note that we only focus on publicly available communication data (in particular, issue reports). Even in an industry setting, similar publicly available data can be found to use as input for emotion mining, without privacy concerns.

#### 3.2 Emotion Mining

In order to rate a particular issue comment as having a particular emotion, each rater identified the emotions associated to each sentence of the comment. We use Parrott’s six primary emotions, i.e., love, joy, surprise, anger, sadness and fear, since we did not know the distribution of emotions.

We mined the issue repository of the Apache software foundation\(^2\), since such a type of repository contains a significant amount of information related to a project’s dynamics [20]. The mined issue reports belong to 117 open source projects, ranging from large, long-lived projects such as Tomcat\(^3\), and Lucene\(^4\), to smaller projects like RAT\(^5\). For the experiment, we use a random sample of the project’s comments, so the larger the project the higher the chance that we analyzed some of its comments. Given this wide range of projects, we believe that this data source provides a representative overview of open source issue reports.

We parsed Apache’s Jira-based repository in July 2013, fetching all the issue reports since the 19th of October 2000. For each issue report, we extracted the developers’ comments, as well as the standard issue report fields mentioned in Section 2.3. Table 2 reports the statistics for our dataset.

Since an issue comment can consist of multiple sentences, and an issue report of multiple issue comments, we decided to perform our analyses at the level of issue comments to avoid a too coarse granularity. Given the large number of issue comments, we sampled enough issue comments to obtain a confidence level of 95% and confidence interval of 5%. This means that a proportion of X% in our sample of issue comments actually corresponds to X ± 5% in the population of issue comments. For this reason, we obtained samples of 392 or more comments (out of 271,416 comments).

Note that we only focus on publicly available communication data (in particular, issue reports). Even in an industry setting, similar publicly available data can be found to use as input for emotion mining, without privacy concerns.

To measure the degree of inter-rater agreement on identified emotions, we calculate either Cohen’s \( \kappa \) value [10] (two raters) or Fleiss’ \( \kappa \) value [14] (more than two raters). Both values can be interpreted according to Table 3. In order to determine whether inter-rater agreement values differ statistically significantly, we also provide the values’ corresponding confidence interval (with \( \alpha \) value of 0.05). If this interval overlaps with another value’s interval, we cannot conclude that the two agreement values are significantly different. In addition to these statistical agreement values, we also provide the more basic percentage of cases for which raters agree on a particular emotion or set of emotions.

Table 3: Interpretation of Cohen and Fleiss \( \kappa \) values.

<table>
<thead>
<tr>
<th>( \kappa ) value</th>
<th>interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;0</td>
<td>poor</td>
</tr>
<tr>
<td>0–0.20</td>
<td>slight</td>
</tr>
<tr>
<td>0.21–0.40</td>
<td>fair</td>
</tr>
<tr>
<td>0.41–0.60</td>
<td>moderate</td>
</tr>
<tr>
<td>0.61–0.80</td>
<td>substantial</td>
</tr>
<tr>
<td>0.81–1.0</td>
<td>almost perfect</td>
</tr>
</tbody>
</table>

\(^2\)https://issues.apache.org/jira/secure/Dashboard.jspa
\(^3\)http://tomcat.apache.org/
\(^4\)http://lucene.apache.org/
\(^5\)http://creadur.apache.org/rat/
In order to compare both groups’ ratings, we mapped each member in group A to a member in group B with similar experience (for example, person 1 of group A (p1A) and person 1 of group B (p1B)). In order to compare the ratings between two groups, each couple like (p1A, p1B) received the same assignment (modulo random reordering).

Third, since we want to verify the influence of context on emotion rating, we divided the experiment in two rounds where the participants rate each of the comments assigned to them twice: once without its context and once with. So, given a particular couple’s assignment, we randomly added context for some of the comments in one round, while we added context for the other comments in the second round, as shown in Table 4.

Fourth, to reduce the impact of seeing first a comment with or without context, we made it such that the assignment of p1A (after randomly adding context) for round 1 corresponds to the assignment of p1B in round 2, while the assignment of p1A in round 2 corresponds to the assignment of p1B in round 1. In each round, all participants rate 98 comments (and each group 392 comments).

Finally, to counter the learning effect and at the same time obfuscate the goal of the study, the two rounds were separated by a time gap of at least 6 days in between submitting the results of the first round and starting the second round.

Similar to the pilot study, in each round each participant received a file with comments. As a means of training, he or she received an explanation about the Parrot framework as well as examples of each emotion based on the results of the pilot study (Section 4). Then, the participants analyzed the list of comments to mark all Parrot emotions that they were able to identify. For comments with context, the participants rate only the emotions in the comment (not the other comments of the context). The dataset of all experiments will be available by contacting the authors.

This section presents the typical emotions that we identified in issue reports during the pilot case. We opted to discuss these emotions up front in order to provide a better understanding of the emotional content of issue reports. For each of the six primary Parrott emotions, we report the most representative text snippets as well as an explanation of why the snippet contains that emotion. Whenever appropriate, we also report the secondary (e.g., shame) or tertiary (e.g., guilt) emotion.

## Love

1. Thanks for your input! You’re, like, awesome
2. Thanks very much! I appreciate your efforts
3. [I] would love any

Love is generally presented in sentences that express gratitude, i.e., a developer exhibits that he “likes” the person (example 1), the person’s activity (example 2) or the software artifact delivered (e.g., patch). Love can be expressed also as a kind of desire (example 3). In issue comments, love is oriented primarily towards co-workers.

## Joy

1. I’m happy with the approach and the code looks good
2. great work you guys!
3. Hope this will help in identifying more usecases

Joy is normally associated to positive achievements, in the form of satisfaction (example 1) or enthusiasm (example 2). In the first case, the text reports keywords like “good” or “great”. In the second case, the phrase ends with

6http://bit.ly/1g1olgq
a “!”. A less common case of joy is optimism expressed by a positive outlook for a successful achievement (example 3). Joy is expressed towards software artifacts or co-workers.

Surprise
1. I still question the default, which can lead to surprisingly huge memory use
2. I also documented an unexpected feature with the SlingServletResolver
3. Oops. It needs to be added to Makefile

Surprise is expressed for unexpected, generally negative, behavior of the software (example 1 and 2). A second case is represented by mistakes introduced accidentally by a developer and discovered later on (example 3). We did not document any case where surprise referred to co-workers.

Anger
1. I will come over to your work and slap you
2. WTF, a package refactoring and class renaming in a patch?
3. This is an - ugly - workaround

Anger generally goes along with menaces (e.g., “slap” or “kill”), negative adjectives (e.g., “ugly”) or profanity (e.g., “WTF”). These emotions reveal hostility and bullying towards co-workers (example 1) or dislike towards software artifacts (examples 2 and 3).

Sadness
1. Sorry for the delay Stephen.
2. Sorry of course printStackTrace() wont work
3. wish i had pay more attention in my english class .... now its pay back time .... :-(
4. Apache Harmony is no longer releasing. No need to fix this, as sad as it is.

Sadness is generally expressed by developers that feels guilty, i.e., they apologize for a delay (example 1) or for the unsatisfactory code produced (example 2). Sadness can be expressed also for reasons not dependent on the issue handled (example 3), or on the developer actions (example 4).

Fear
1. I’m worried about some subtle differences between char and Character
2. I’m most concerned with some of the timeouts
3. I suspect that remove won’t work either in this case.

Fear is expressed by a developer in a state of worry or anxiety. This emotion is expressed explicitly using the keyword “worry” or its synonyms like “concern” (example 1 and 2). Another common case is to express a negative outlook with respect to a particular development choice (example 3). We observed that fear, differently from other emotions, is limited to software artifacts only.

Issue comments contain emotional content of developers. Some emotions can refer to software artifacts and co-workers (e.g., joy, anger and sadness), while others target only software artifacts (e.g., surprise and fear) or co-workers (e.g., love).

Table 5: Percentage of comments in which raters agreed on presence or absence of all 6 emotions, as well as the number of those comments with at least one emotion present.

<table>
<thead>
<tr>
<th></th>
<th>#common</th>
<th>%common present</th>
<th>#common</th>
<th>%common present</th>
<th>#common</th>
<th>%common present</th>
</tr>
</thead>
<tbody>
<tr>
<td>pilot study</td>
<td>155</td>
<td>41.25±5</td>
<td>49.23±5</td>
<td>52.81±5</td>
<td>197</td>
<td>40.95±0</td>
</tr>
<tr>
<td>round 1 group A</td>
<td>215</td>
<td>54.85±5</td>
<td>32.40±5</td>
<td>36.16±5</td>
<td>207</td>
<td>52.00±5</td>
</tr>
<tr>
<td>round 2 group A</td>
<td>29</td>
<td>5.00±5</td>
<td>9.18±5</td>
<td>9.18±5</td>
<td>40</td>
<td>10.00±5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. EXPERIMENTAL RESULTS

For each research question, we first discuss its motivation, followed by the approach we used and our results.

RQ1. Can human raters agree on the presence or absence of emotions in issue reports?

Motivation. Emotion mining from software development artifacts like issue reports, emails or change logs is not trivial, since such artifacts consist of unstructured data [2, 3]. I.e., they are relatively short, written in an informal way with emoticons, and, contrary to regular text parsed in sentiment analysis, they typically contain technical content like stack traces or code snippets interleaved with regular text. Because of this, it could prove to be difficult to glance emotional content from software development artifacts, let alone agree between different human raters.

Approach. We use the comment ratings of the pilot study and full study to address this research question. As a first step, we measured the percentage of agreement on the presence and absence of emotions. As a second step, we used Cohen’s κ to calculate agreement across all raters for each comment. For the full study, we measured the agreement (a) for each pair of (round, group), and (b) across both groups. In the latter case, we merge the corresponding ratings of (round 1, group A) with (round 2, group B), and of (round 1, group B) with (round 2, group A). Comparing cases (a) and (b) allows to evaluate whether agreement changes when having two, three or four raters for a comment.

Findings. Only in on average 46.11±5% of the comments, both raters had the same rating for all 6 emotions. Table 5 shows for each study the number and percentage of comments for which both raters assigned to the same comment agreed on all 6 emotions. The highest number of such agreement occurred for group A in round 1, while the same group obtained the lowest agreement in round 2, statistically significantly lower than in the four other cases (except for the pilot study). Since (round 1, group A) and (round 2, group B) both considered the same comments (and context), that configuration of comments and context seems more easy for raters to agree on.

Furthermore, on average for 7.47±5% of the comments for which raters agreed on all 6 emotions, at least one emotion

Table 6: Percentage of agreement (absolute number in parentheses) and Cohen κ values (with confidence intervals) for each emotion in RQ1 (pilot study).

<table>
<thead>
<tr>
<th></th>
<th>%agreement present (%)</th>
<th>%agreement absence (%)</th>
<th>Cohen κ (lower, upper)</th>
</tr>
</thead>
<tbody>
<tr>
<td>love</td>
<td>90.75±5 (363)</td>
<td>5.75±5 (23)</td>
<td>0.38 0.51 0.64</td>
</tr>
<tr>
<td>joy</td>
<td>69.75±5 (279)</td>
<td>6.50±5 (26)</td>
<td>0.11 0.19 0.27</td>
</tr>
<tr>
<td>surprise</td>
<td>96.75±5 (387)</td>
<td>0.50±5 (10)</td>
<td>0.02 0.01 0.00</td>
</tr>
<tr>
<td>anger</td>
<td>90.75±5 (363)</td>
<td>2.00±5 (2)</td>
<td>0.07 0.06 0.19</td>
</tr>
<tr>
<td>sadness</td>
<td>80.75±5 (323)</td>
<td>3.50±5 (14)</td>
<td>0.06 0.18 0.29</td>
</tr>
<tr>
<td>fear</td>
<td>93.25±5 (373)</td>
<td>2.50±5 (2)</td>
<td>0.07 0.10 0.26</td>
</tr>
</tbody>
</table>
why, despite the high percentage of general agreement, the

Table 6 and Table 7 show the percentage and Cohen’s κ was present. Typically, raters agreed on absence of emotions

Table 8: Percentage of agreement (absolute number in parentheses) and Cohen’s κ values (with confidence intervals) for each emotion in RQ1 (full study).

Table 7: Percentage of agreement (absolute number in parentheses) and Cohen’s κ values (with confidence intervals) for each emotion in RQ1 (full study).

For example, group A did not have any agreement at all for surprise in round 2, contrary to most of the other cases.

The percentage of agreement for joy in the pilot study was the lowest, with 30.25±5% of the comments containing disagreement. However, all other emotions and cases had less disagreement than 19.25±5% (sadness in pilot study).

At most 7.91±5% (love) of the comments agreed on the presence of a particular emotion, whereas up to 96.75±5% (surprise) agreed on the absence of a particular emotion. Table 6 and Table 7 indeed show that most of the comments were rated as not having a particular emotion (an agreed presence of 0% means that there were no comments where an emotion was present). This is the reason why, despite the high percentage of general agreement, the corresponding κ values are low. The emotions with the lowest κ values (fear, anger, and surprise) sometimes have only 0, 1 or 2 agreed occurrences, while the most frequently agreed emotion (love) had up to 31 occurrences (group A, round 1).

Only for joy, three raters agree significantly more on an emotion than only two raters. Table 8 breaks these percentages down for each individual emotion, together with the Fleiss’ κ inter-rater agreement and confidence intervals (4th to 6th column). In parentheses the number of analyzed cases of a particular emotion in which raters agreed on the presence of the emotion.

(a) (round one, group A) and (round 2, group B)
(b) (round one, group B) and (round 2, group A)

was present. Typically, raters agreed on absence of emotions for dry comments like “committed” and “done”.

Only for love, the raters achieved moderate agreement, while joy and sadness obtained fair agreement. Table 6 and Table 7 show the percentage and Cohen’s κ values (with confidence interval) for agreement of each emotion individually, for the pilot and large study respectively. Love clearly obtains the highest κ agreement, corresponding to a moderate value. Except for the pilot study, joy and sadness have a strong fair agreement. Fear, anger, and (especially) surprise only obtained poor/slight agreement. These numbers are more or less stable across the five cases, with some fluctuations. For example, group A did not have any agreement at all for surprise in round 2, contrary to most of the other cases.

RQ2. Does context improve the agreement of human raters on the presence of emotions in issue reports?

Motivation. Rating a comment without its context can be compared to eavesdropping on a group conversation and only catching the last phrase of the conversation. It is likely that the interpretation of that phrase depends on the previous discussion (i.e., context) of the conversation. For example, the sentence “yeah, right” can have a different meaning (both sarcastically and otherwise) [38], when following a sentence

While some emotions obtain higher agreement than others, only love, joy and sadness obtained at least fair agreement. Although comments clearly contain emotions, raters agree the most on the absence of an emotion. Having more than two raters does not seem to make a significant difference.
Table 10: How often raters changed their rating from the one in a row (comment without context) to the one in a column (comment with context). 0 means that a particular emotion was not selected, while 1 means that it was selected.

<table>
<thead>
<tr>
<th>context</th>
<th>love</th>
<th>joy</th>
<th>surprise</th>
<th>anger</th>
<th>sadness</th>
<th>fear</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 11: How often raters went from disagreement (d) to agreement (a) or vice versa when comparing the set of comments without context (rows) to the set of comments with context (columns), for groups A, B, and when combining both groups (at least three raters agreeing).

<table>
<thead>
<tr>
<th></th>
<th>love</th>
<th>joy</th>
<th>surprise</th>
<th>anger</th>
<th>sadness</th>
<th>fear</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>d</td>
<td>a</td>
<td>d</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>group A</td>
<td>25</td>
<td>32</td>
<td>19</td>
<td>24</td>
<td>34</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>29</td>
<td>15</td>
<td>13</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>group B</td>
<td>19</td>
<td>24</td>
<td>19</td>
<td>24</td>
<td>32</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>347</td>
<td>30</td>
<td>29</td>
<td>345</td>
<td>348</td>
</tr>
<tr>
<td>3-rater</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>361</td>
<td>26</td>
<td>349</td>
<td>375</td>
<td>370</td>
</tr>
</tbody>
</table>

like “with java 8 we fix all problems” when following “breaking backward compatibility is risky”. However, due to the technical and unstructured nature of software development artifacts, the impact of context might be different than in literary English documents. Here, we want to analyze the impact of context on agreement between raters.

**Approach.** This research question only considers the full study. Since each group considers 392 comments once without and once with context (they were randomly distributed across two rounds), here we merge the results of both rounds such that, for each group, we can compare the ratings without and with context. For this comparison, we calculate similar agreement percentage and Cohen κ values as for RQ1. Furthermore, we measure how often raters made a different decision for a particular emotion when seeing context or not, and whether such different decisions led from agreement to disagreement, disagreement to agreement or did not have any net effect.

**Findings.** Adding context slightly reduces rater agreement, but not significantly. Table 9 compares, for each group, the agreement amongst the rating results of the comments without context (odd rows) and those with context (even rows). Except for surprise in group A, the κ agreement is not significantly different (the confidence intervals still overlap) with or without context, even though the actual κ values seem lower with context than without. Similarly, the percentages of agreement seem lower with context, but not in a significant way. Sometimes context finds more evidence of the presence of emotion than without context, while sometimes the inverse situation holds. Similar to RQ1, both groups have similar results, except for anger, for which group B had a much lower agreement (since less occurrences were agreed upon).

Most of the raters pick the same answer without or with context, yet they tend to switch more from absence to presence of an emotion than the other way around. Table 10 shows for both groups how many raters picked a different answer for an emotion in the absence or presence of context. Clearly, in most cases (between 90.3% to 95.7% of the time) raters did not change their rating, which suggests that (1) ratings for a particular comment are fairly stable, and (2) context does not add substantially new information for the interpretation of a particular comment.

At the same time, we can also see that if a rater changes his or her mind, he or she rather tends to mark a previously (i.e., without context) absent emotion as present, than the other way around (except for love, surprise and fear in group A). In the case of sadness the relative difference between both cases goes from 50% to more than 100%. This would suggest that although context does not play a major role in agreement, in cases when it does, raters become less sure and tend to mark an additional emotion as being present.

The change of mind due to context pushes more pairs of raters from agreement to disagreement than the other way around. Table 11 shows for each comment and emotion whether the raters’ change of mind has an impact on the agreement between the raters. Even though the vast majority of cases did not change agreement/disagreement, the results also show for all emotions and both groups that more raters went from agreement to disagreement when showing context (row “a”, column “d”), than the other way around (row “d”, column “a”). Again, context does not seem to have a major impact, but when it does, it causes more uncertainty (disagreement) than agreement.

This observation is less pronounced when using agreement between at least three raters, as shown in the bottom two rows of Table 11. Only for love and joy, there is still more agreement turning into disagreement than the inverse, but for the other four emotions, the usage of three or more raters makes the results more robust to fluctuations introduced by context. Hence, even though more raters do not significantly improve agreement (RQ1), they make ratings more robust.

**6. DISCUSSION**

This section discusses our findings in more detail.

**6.1 Impact of Context**

At first sight, our findings for RQ2 seem counter-intuitive: while one would expect that the addition of context strengthens agreement due to the availability of more information, we seem to observe quite the opposite, i.e., raters starting to doubt and changing their mind. Although more experiments are needed to confirm and understand this phenomenon, we briefly discuss a couple of hypotheses.

The worst case scenario would be that emotion mining is so subjective and nuanced that even for humans it is impossible to correctly determine the presence of a specific emotion in an issue report. However, we believe that the truth is more subtle. For example, in RQ2 we only rated the last comment of an issue report, and reports with context contain (by definition) the viewpoint of multiple commenters, for which it is not always clear how they relate to the last commenter’s viewpoint.

Consider a hypothetical example of the following three comments by three different commenters: “Class FooBar is a total...
Table 9: Percentage of agreement and Cohen $\kappa$ values (with confidence intervals) for comments without and with context (RQ2). The percentages are relative to the 392 comments without and with context, respectively odd and even rows.

<table>
<thead>
<tr>
<th>(a) Group A</th>
<th>(b) Group B</th>
</tr>
</thead>
<tbody>
<tr>
<td>%agreement</td>
<td>%agreed</td>
</tr>
<tr>
<td>love</td>
<td></td>
</tr>
<tr>
<td>88.23±5.3 (346)</td>
<td>3.91±3.5 (314)</td>
</tr>
<tr>
<td>joy</td>
<td>86.22±5.3 (338)</td>
</tr>
<tr>
<td>surprise</td>
<td>91.07±5.3 (357)</td>
</tr>
<tr>
<td>anger</td>
<td>92.86±5.3 (364)</td>
</tr>
<tr>
<td>sadness</td>
<td>96.41±5.3 (374)</td>
</tr>
<tr>
<td>fear</td>
<td>93.11±5.3 (365)</td>
</tr>
</tbody>
</table>

waste of time, just nuke it!”, “We do have users relying on its features, I’m afraid we should fix this bug” and “I share your view, working on it”. Although the first comment clearly contains anger and the second one sadness, the third one is quite ambiguous regarding which view is shared. Without context, the comment might be neutral, while with context it might be neutral, anger, sadness or a combination. As such, context does not necessarily filter the set of possible emotions. On the contrary, it enforces the nuances on the emotions perceived by a rater and can lead to different interpretations.

Another hypothesis is that using a simple yes/no decision as rating is too large a simplification. Maybe one should provide multiple ratings, which would allow to model uncertainty in a rating.

6.2 Do Emotions Matter for Issue Reports?

Our premise was that, similar to other domains, emotions could have an impact on software development activities like bug fixing or development of new features. To get an initial impression of whether this claim contains a grain of truth, we performed a short analysis with data stemming from this study and from a smaller, preliminary one made internally. Here, we check whether reports with different emotions tend to be fixed faster, have more comments or have more people following (‘watching’) the issue report.

We found a significant difference for the number of comments, i.e., reports with a comment rated as love tend to have a lower number of comments (median value of 5) than joy (median value of 7.5) or sadness (median value of 12). Similarly, the number of watchers of reports with a comment rated as love has a median value of 0 whereas for sadness the median value is 1, i.e., less people monitor the former reports. Although not strictly significantly different, the Kruskal-Wallis test for the fixing time of reports obtained a low p-value of 0.057, with reports containing a love comment taking a median number of 20 days to be resolved, compared to 53.5 for joy and 68.5 for sadness.

Of course, more analysis is needed to fully investigate the link between emotions on software development, however, our initial findings for number of comments, watchers and (to a lesser degree) fixing time suggest that there might indeed be a link.

7. 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 emotions and what he or she writes in issue report comments, based on empirical evidence (in another domain) [29]. Moreover, since developer communication has as first goal information sharing, removing or disguising emotions may make comments less meaningful 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.

Threats to construct validity focus on how accurately the observations describe the phenomena of interest. Rating of emotions from textual issue report comments presents some difficulties due to ambiguity and subjectivity. To reduce these threats, the authors adopted Parrott’s framework as a reference for emotions. Finally, to avoid bias due to personal interpretation, in all experiments each commit was analyzed by at least two participants.

Threats to external validity correspond to the generalizability of our experimental results [8]. In this study, we manually analyze a sample of issue reports belonging to 117 open source projects. We chose 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 emotions of developers from issue reports, hence no ground truth exists to compare our findings. We defined
the ground truth through agreement or disagreement of the raters. Since we involved 16 people with a wide variety in experience, nationalities and culture, other groups of raters might obtain agreement on different emotions and comments, possibly leading to different results. However, RQ2 showed that both groups of the full study and, to some extent, the pilot study obtained similar levels of agreement.

This study is focused on text written by developers for developers. To correctly depict the emotions embedded in such comments, it is necessary to understand the developers’ dictionary and slang. This assumption is supported by Elfenbein and Nalini’s work that 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 [13]. Since all the participants of this study have a background in computer science, we are confident that participants may interpret the issue comments in the same manner as the developers. We did not involve raters with different background (such as linguists or psychologists), because they may make oversights or misinterpret the terms used by developers.

8. RELATED WORK

Software development is a collaborative activity dependent on human interaction between developers towards the timely completion of a high quality software system [26]. Existing software development studies are not focused on the analysis of individual feelings, even though the morale and emotions of individuals can impact on the collaboration process and subsequently the success of a product [6, 1]. In this paper, emotions are the first-class object, we study how developers feel towards software artifacts and/or colleagues.

A project that is not appealing and unable to motivate developers to join, can be destined to fail [16]. For this reason, there has been a great deal of research on the reasons of developers joining and leaving a software project. Sim et. al have investigated dynamic strategies and conditions that characterize the joining process of software immigrants in a software development team [37]. Ducheneaut analysed the socialization process of new developers by visualizing the dynamic networks of both human and material resources incorporated in the email and code databases of open source software [11]. Herraiz et al. studied the duration and basic characteristics of the joining process for the developers and found two groups with clearly different joining patterns: new professionals and volunteer developers [23]. They could relate those patterns to the different behavior of volunteers and hired developers.

Rigby et al. analyzed the five big personality traits [27] of software developers in the Apache httpd server mailing list [34]. They analyzed the personality of four top developers, assessing the personality of two top developers who have left the project. Bazelli et al. replicated Rigby’s work on Stack Overflow to analyze the personality traits of different authors [4]. Although both studies deal with the behaviour of developers, they do not consider the emotions of project members and how these impact the motivation of project members.

Typically, emotion and sentiment mining studies focus on emotions that people convey through twitter, question and answer sites and product reviews [28, 24, 25]. Choudhury et al. [9] applied sentiment analysis on microblogging data of a large global Fortune 500 software corporation used to “disseminate product/service updates to the larger enterprise community, such as details of new upcoming features, news about future team meetings, releases or trade shows”. They found that there is a potential for building systems that assess feelings of employees in an enterprise.

To improve 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 [19]. They used Latent Dirichlet Allocation to find the topics discussed in email and web discussions of students in the context of a class project, then used lexical sentimental analysis to obtain an average emotion score for each of the topics. Dulemond et al. [12] extended a microblogging tool with a happiness indicator, then deployed the tool across distributed teams in a company. Employees used the tool to share their message and emotions with their colleagues in order to stay more connected and be aware of each other while collaborating. Gómez, using a grep-like approach, mined GitHub for extracting developers commit messages [17]. In his blog he collected several expressions ascribable as anger, joy and surprise, used during coding activity by developers.

Differently from sentiment analysis on high-level feature and meeting announcements, this paper performs emotion mining on technical artifacts, i.e., the comments of issue reports. Although emails and web discussions contain more technical details than the microblogging data sources, issue reports contain even more technical detail, since they are used by team members to ask for advice, express opinion and share opinions related to software maintenance and evolution. In contrast to sentiment analysis on a given emotion, we tried to identify and mine the different types of emotions.

9. CONCLUSION

Emotions influence human behavior and interaction. Software development, as collaborative activity of developers, cannot be considered exempt from such influence. Emotion mining, 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 in their team as well as discover and promote factors that bring serenity and productivity in the community.

As a first step towards evaluating the feasibility of a tool for automatic emotion mining, we performed an exploratory study of developer emotions in almost 800 issue comments during software maintenance and evolution. Our study confirms that issue reports do express emotions towards design choices, maintenance activity or colleagues. Regarding agreement amongst human raters, we found that some emotions like love, joy and sadness are easier to agree on, but that additional context can cause doubt for raters, unless more raters are used.

Our findings suggest that for love, joy and sadness it makes sense and eventually might be feasible to automate emotion mining. However, some challenges, such as the impact of context, need to be studied more closely. Moreover, more systems and data sources need to be taken into account.

10. ACKNOWLEDGMENTS

This work is sponsored by the Institute for the Promotion of Innovation through Science and Technology in Flanders through a project entitled Change-centric Quality Assurance (CHAQ) with number 120028. Regione Autonoma della Sardegna (RAS), Regional Law No. 7-2007, project CRP-17938, “LEAN 2.0”
11. REFERENCES


