Proceedings of the 7th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'21)
~~Prague, Czech Republic~~ Virtual Conference - July, 2021
Paper No. CIST 301
DOI: 10.11159/cist21.301

Detecting Emotional Contagion in OSS Projects

Rashmi Dhakad and Luigi Benedicenti[1]
[1]University of New Brunswick
Fredericton, New Brunswick, Canada
Luigi.Benedicenti@unb.ca

Abstract **-** In recent times, due to an increase of open source software working models, the behaviour and emotional affective state of developers are becoming increasingly relevant as a subject of research in the software engineering community. One important aspect of such research is emotional contagion: the phenomenon by which the affective state of an individual is transferred to another one through communications. In this paper, propose a simple model that determines the presence of emotional contagion, negative or positive, in an open source project in order to enrich the set of tools used to help the developers..

Keywords**:** Hybrid open source · hybrid communities · sentiment analysis · open source software · OSS · open source communities · free/libre/open source software · FLOSS.

1. Introduction

Emotions are an important part of the human behaviour. They are essentially a building block of human interaction. Emotional contagion is a phenomenon that leads people to mimic the intents and emotional states of others at any given moment, even subconsciously. Emotional contagion is important for the human behaviour because it is the main mechanism that can explain how emotions form emotional social influence among individuals [1]. Weiss and Cropanzano have found that there is a link between emotions and work productivity. Their Affective Theory formalizes their findings [2].

With the increasingly popular demand in current software engineering community on the role of developers and their dealing with sentiment and the emotional aspects of the development process, it has become more and more crucial to conduct research that provides the development community with better tools for a more productive and satisfying work environment.

Emotional contagion can be best seen as an influence of emotions or behaviour of an individual on another one, either consciously or subconsciously, resulting in a change of the emotional state and/or behaviour of the other individual [3]. Another important aspect that Barsade et al mention is that it not only contributes to the emotional state and how people feel but also the thinking and action that people take because of the influence of the emotions of another individual [3].

In their study of primitive emotional contagion, Barsade et al also explain how emotional contagion can occur at a subconscious level, as the first and foremost part of the process is mimicry of display of expressions from each other. The second part of the process then focuses on how these facial and behaviour mimicry leads to physiological results. Results from their studies show that the when emotional contagion happens at a subconscious level, the recipient of the emotional contagion is not able to recognize it, and in turn cannot detect the emotional social influence.

When talking about conscious emotional contagion, one where the person is aware of the change, people tend to compare their moods to others and respond accordingly. An interesting example of emotions and moods in the setting of team and individual activities was observed as well: whenever a group or a team activity was conducted as part of the experiment, one where the members depended on one another, the mood linkage was positive and coordinated to make the team effort, but in case of individual activity, it was negative.

Another important discovery that can be derived from these studies is that emotional contagion can occur in groups, which means that emotions or moods collectively of a group can affect one another resulting in unexpected results and conclusions.

From Barsade's study, it can be concluded that in cases where people are able to assess the situations through behavioural and facial expressions and respond accordingly it is easier to control the results of the contagion, as compared to cases where the communication between coworkers or members of the team occurs via textual mediums like emails, chat, online forums and discussions panel. Here, people must assess the expression of the person through what has been communicated via text, which may or may not lead to completely different results and consequences than what the sender of the text intended.

Other studies on emotional contagion found that group emotions exist. This is defined as group affective tone and it can influence the work outcomes, creating different results than expected.

Thus, understanding whether the emotions of an individual developer in a software development team influence other developers emotions is an important research topic that has not yet been sufficiently explored in Software Engineering.

Software Development is a human centered process. Software engineers are creative human beings despite working in engineered processes [5]–[9]. Software development is a sense-making process as well as a decision-making process, and sense-making requires the conscious assessment of the environment and can be influenced by individual or group emotions [10]. Additionally, software development requires cognitive processing abilities and problem-solving skills.

Psychology studies show that emotional contagion causes a change in the emotional dynamic of the overall team. Scientifically it has been proven that emotional contagion is a type of social influence, in which one person's emotions (short-term reactions to an environmental stimulus) can affect another, resulting in changed or varied results than what was expected. However, this knowledge has seldom been applied to software development.

In this position paper, we attempt to detect the presence of emotional contagion in a development environment. The proposed concept is to conduct an analysis of commit messages of an open source project repository to gather the result of whether there is a presence of emotional contagion. Such an analysis has been done before to detect an individual's affective state, but it has not previously been performed to detect emotional contagion as a group dynamic.

The rest of the paper is structured as follows. In Section 2, we present the background of the paper. In Section 3, we introduce our case company and case data set. In Section 4, we present the analysis done for the project and key findings. Towards the end of the paper, in Section 5, we draw some conclusions.

2. Background and Motivation

Many literature studies and research contributions have proven that emotions play a vital role in an individual's performance and productivity while completing a task. For example, in case of cognitive processing ability, "emotion has a substantial influence on the cognitive processes in humans, including perception, attention, learning, memory, reasoning and problem solving" [11].

To improve the software development process, we have to focus on its main aspect: people. People drive the process, and its performance [5].

Emotions and moods of an individual affect their decision making abilities [12]. Under either positive or negative emotions, a decision maker makes different decisions despite other factors being the same or very similar. In fact, decision makers adopt different processing strategies depending on their affective state [13]. Individuals with positive emotions are inclined towards risk seeking, whereas individuals with negative emotions are inclined to be risk averse. Numerous other studies relate the role of emotions, moods and affect with agile decision making [14]–[17].

In case of pull requests in open source software and other online open source platforms, it has been observed that the issues with higher level of negative emotions like sadness, arousal and dominance are less likely to be merged as compared to issues with higher level of valence and joy [18].

In the last two decades, several studies have been conducted on the emotions of developers. A very good example is a study conducted in 2019 [19], in which the authors used an off the shelf sentiment analysis tool to assess whether maintainers' sentiment changed after one of the main developer of the community, Linus Torvalds decided to take a temporary break from the community. From the study of the various community records, it can be concluded that even in case of textual communication, a negative tone, as described in the study, may impact the individuals creativity, productivity, and contribution towards the project. This study was especially important to explain how sentiment influence development during unexpected events.

Several studies have been conducted in the past for deeper and better understanding towards the role of human aspect in an engineered process as software development. Many of these studies address why people get affected or influenced by emotions whilst other studies talk about when people get affected by emotions [8], [9], [20]. Despite these studies and more literature evidence, software engineering lacks a solid foundation of role of emotions in software development. A different study concluded that many other researchers have called for research on this subject [21].

Many organizations try to capture the emotions overflow of an individual and their emotional attachment to their job, or job, or job satisfaction via interviews and surveys. But in a global software development team, where the team is geographically distributed across the globe, it is difficult to capture these results. Especially as mentioned earlier, in case of facial and behavioral mimicry, people tend to act accordingly to the situation present, as opposed to textual communication where the assessment of the text is via the language and the tone of the text.

3. Proposed Model

Many studies call for software engineering to focus on emotional contagion and the understanding of the flow of emotions in development teams spread geographically across globe and how we can minimize the negative consequences of this flow and provide solution for this.

The software development process works in collaborations and teamwork. It is centered around working together and interacting with one another. With interaction, it is bound to have an exchange and invoke of emotions amongst participants in the process. This exchange of emotions often can be of positive emotions and at times of negative emotions.

Studies conducted in the past have made use of sentiment analysis tools and analysis by researchers themselves to find the answer to the question of whether there are emotions influencing the software development process.

Some of these studies raise the validity of the sentiment analysis tools. For example, in Murgia's study [22], the question of verification of the feasibility of a sentiment analysis tool is raised. The study observes that if researchers, who are essentially humans, cannot detect the emotional information flow in the issue reports then the validity of an automatic tool is even more in question.

Emotions and display of emotions in discussion boards is pretty common, since most of the face-to-face meetings are not feasible or efficient nowadays, with software development being distributed over the globe. Geographically distributed project teams have almost no personal communications amongst themselves, which raises the potential of conflicts among the team, as described previously, because being able to gauge the facial and behavioral outcome of a text sometimes leads to misinterpretation of that text [23]. In environments such as these where the project team is global and distributed, it becomes difficult for project managers as well to deflect and resolve issues among the team on time, which then leads to unexpected and unfavorable outcomes [3].

Gaining awareness of the emotional flow amongst the team members and being able to diffuse conflicts and provide consistent results is still a manual process. Researchers have used various methods to mine emotions in different settings in software engineering. Many of the studies have focused on mining and analyzing the emotions of developers as individuals, and how they are affected by emotions and the emotions of others in the team. But little or no studies relate the emotions of a developer to another or link the affect of one individual's emotions to another one's. This gap in the area requires further call for research from the community.

Based on the previous literature review and studies conducted across a wide dynamic including psychological and empirical software engineering, we propose the following hypotheses for our model:

1. A negative response may evoke a negative contagion among the community.
2. A positive response may evoke a positive contagion among the community.
3. A neutral response can be deemed as neither positive or negative, and thus doesn't affect the curve, hence its contagion capability cannot be determined.
4. A negative sequence leads to more commits with errors.
5. A positive sequence leads to a smaller number of commits as well as fewer commits with errors.
6. The day of the week also affects the resulting sequence, as negative sequences or positive sequences seem to occur at different and recurring intervals of the week.

Using the model in Fig.1 we will investigate the relationship between the commit messages and time sequence.

To test all the hypotheses presented in this paper, we adopted the method described in Figure 1. The method is in four steps. First, we collect the dataset using GitHub's API. Github is a well-rounded source of many open source repositories. Moreover, in order to collect all the information on commits including the date, time, author, as well as the commit ID, there are several available APIs which give the desired results [24]. For the analysis, we will be adopting Mathematica, a technical computation software system with built-in functions for different computing services [25]. its built-in classifier we will be able to run more sophisticated and pattern sequence models. The underlying power of Mathematica is in the classifiers which are based on neural networks and can be trained on sample data with predictable levels of accuracy [26]. The resulting sequence of the analysis will then be plotted as a sequence of numerical values of -1,0,+1 against time and date of the commit message. Figure 4 shows an example in which we have trained the classifier with 750 examples from a training dataset. The Classify function tries many models and chooses the one best suited based on the different features of the dataset.

Through this method, we can test out the hypotheses discussed earlier. With the knowledge of the presence of emotional contagion we can further dive into answering questions based on the consequences of this contagion. Whether the positive contagion or negative contagion affects the developers or whether the knowledge of their presence not having a neutral contagion affects their decision-making are some of the aspects that can be answered through this analysis.

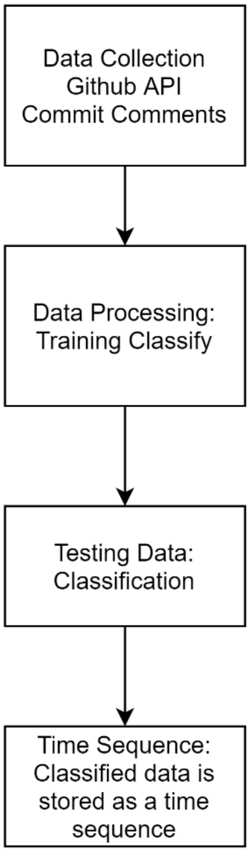
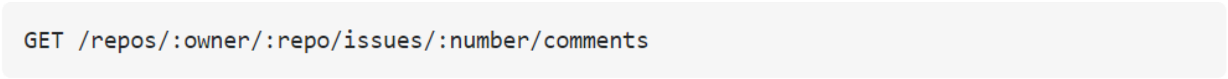
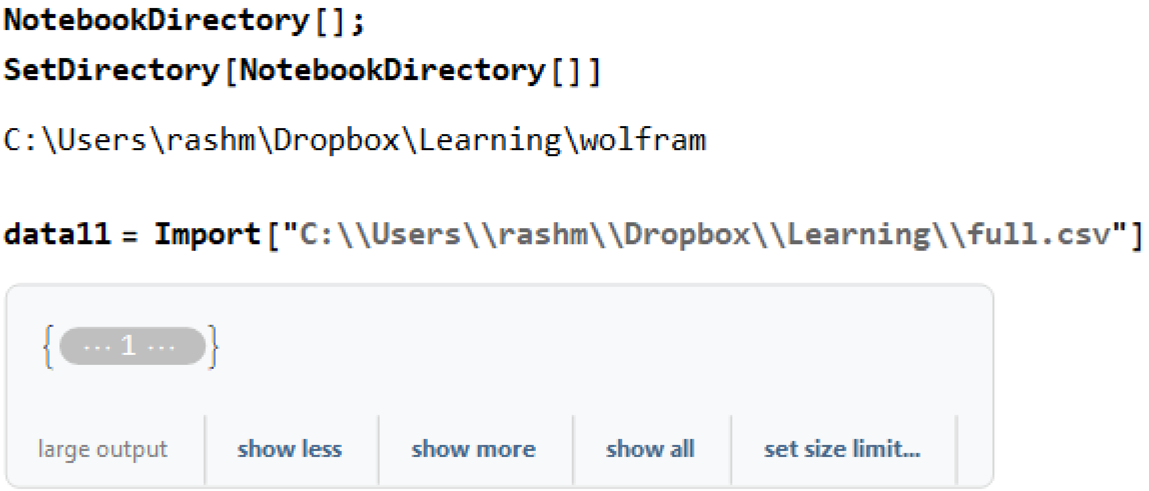


Fig. 1: Method for determining the presence of emotional contagion.



```
GET /repos/:owner/:repo/issues/:number/comments
```

Fig. 2: Sample Git Clone

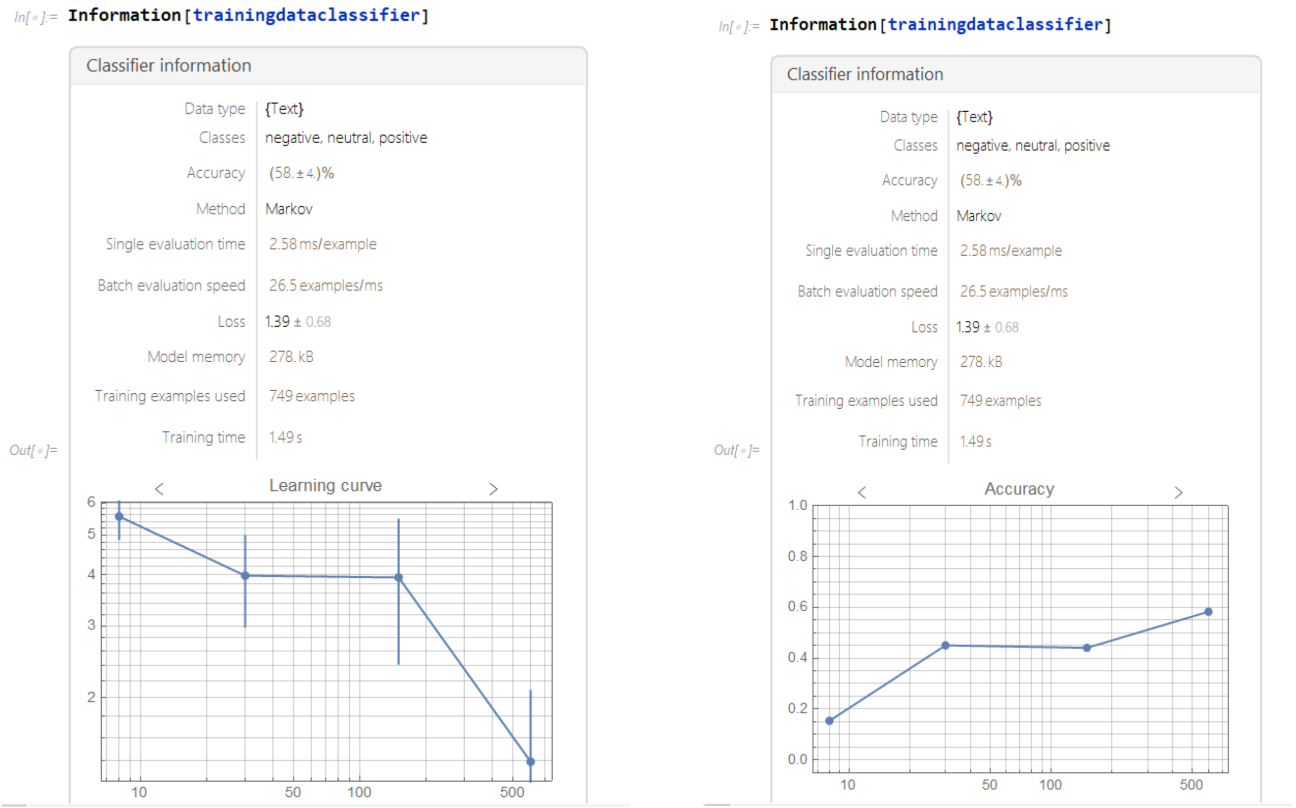


```
NotebookDirectory[];
SetDirectory[NotebookDirectory[]]

C:\Users\rashm\Dropbox\Learning\wolfram

data11 = Import["C:\\Users\\rashm\\Dropbox\\Learning\\full.csv"]
```

Fig. 3: Sample Mathematica Code.



```
In[ ]:= Information[trainingdataclassifier]
```

Classifier information

Data type	{Text}
Classes	negative, neutral, positive
Accuracy	(58. ± 4.)%
Method	Markov
Single evaluation time	2.58 ms/example
Batch evaluation speed	26.5 examples/ms
Loss	1.39 ± 0.68
Model memory	278. kB
Training examples used	749 examples
Training time	1.49 s

```
In[ ]:= Information[trainingdataclassifier]
```

Classifier information

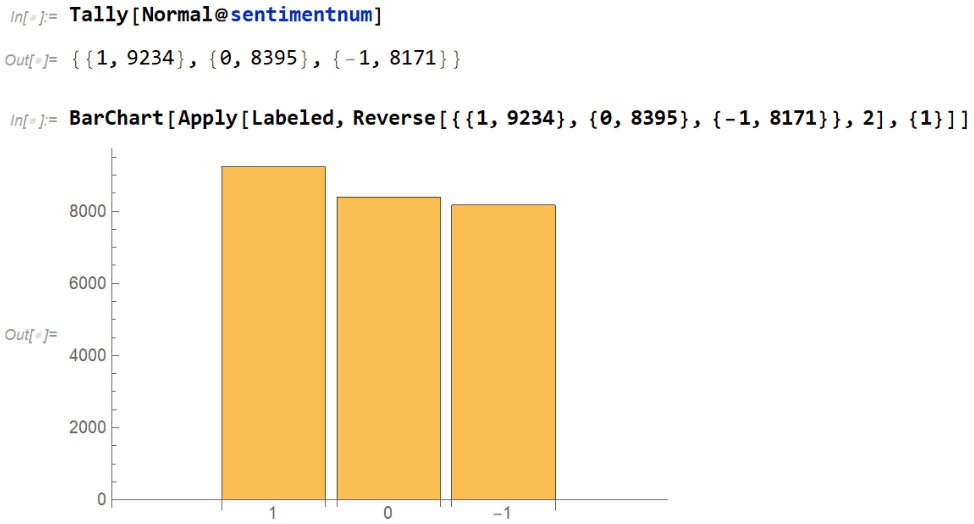
Data type	{Text}
Classes	negative, neutral, positive
Accuracy	(58. ± 4.)%
Method	Markov
Single evaluation time	2.58 ms/example
Batch evaluation speed	26.5 examples/ms
Loss	1.39 ± 0.68
Model memory	278. kB
Training examples used	749 examples
Training time	1.49 s

Fig. 4: Classifier Accuracy.

4. Results

In order to test the hypotheses, we followed two different approaches, for the first approach we used the Classify which is available in Mathematica. Classify function as discussed previously is based on supervised machine learning algorithms, which we can train for any type of data. In our case, we used this classifier to identify our commit messages. the classification step which involves artificial neural networks, we train the machine with training dataset, with larger training data is, the efficiency of the classifier is increases. The training step is followed until the machine gains a potential level of accuracy. Mathematica, automatically chooses the best algorithms based on the different models stored. For this present method, it relied on the Markov method, which is generally used for textual analysis.

For the second approach, we used Mathematica’s built in classifier, Sentiment. This is a classifier which tries to gather the sentiment from a text. This classifier relies on the basis of assuming that the text conveys only one sentiment. In the figure as we can see, a total of 12433 messages were determined as neutral, where as in comparison to the trained method, only 8395 messages were determined as neutral. Similarly, in case of positive and negative messages , in case of trained method, a total of 9234 messages were identified as positive and 8171 messages as negative, where as the built in classifier identified a total of 3942 messages as positive and 9346 messages as negative. Notice in Fig. 6, there are some messages which are left Indeterminate since the classifier couldn’t identify these cases, where as the trained model was able to classify all the messages across the dataset.



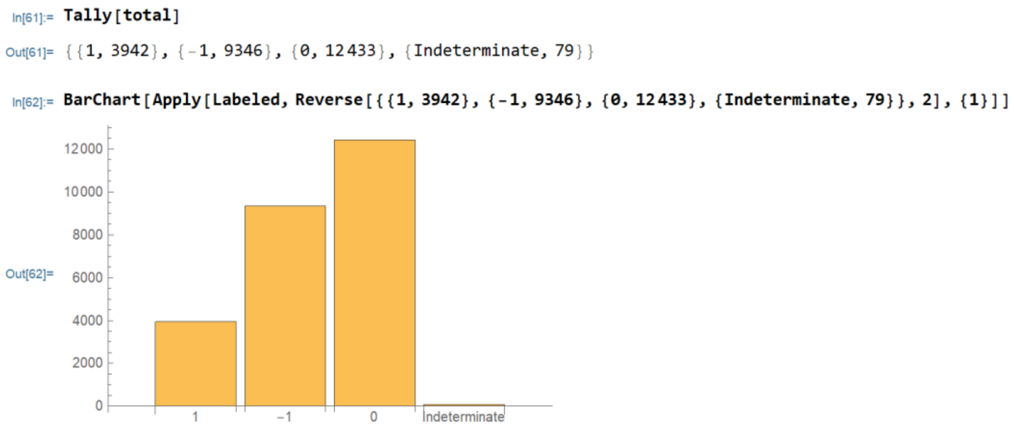
```
In[•]:= Tally[Normal@sentimentnum]

Out[•]= {{1, 9234}, {0, 8395}, {-1, 8171}}

In[•]:= BarChart[Apply[Labeled, Reverse[{{1, 9234}, {0, 8395}, {-1, 8171}}, 2], {1}]]

Out[•]=
```

Fig. 5: Total Commit Messages (Trained Classifier)



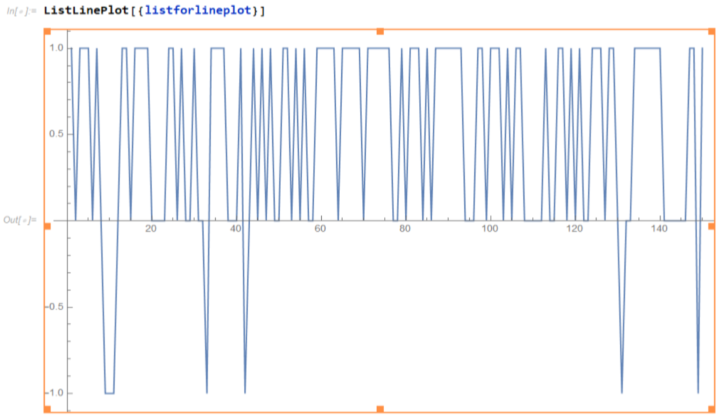
```
In[61]:= Tally[total]
Out[61]= {{1, 3942}, {-1, 9346}, {0, 12433}, {Indeterminate, 79}}

In[62]:= BarChart[Apply[Labeled, Reverse[{{1, 3942}, {-1, 9346}, {0, 12433}, {Indeterminate, 79}}, 2], {1}]]
Out[62]=
```

Fig. 6: Total Commit Messages (Sentiment- Built in Classifier)

This approach can be used further to answer our initial hypotheses on contagion. Here in Fig. 7, we have used a small small snippet to focus on the fluctuation of contagion. As we can see, there are more accumulation of positive contagion in comparison to negative. Similarly in case of neutral contagion, the sequence highlights that either there is a positive contagion from neutral message, or it remains the same.

In case of neutral contagion, as of this period it is perhaps difficult to say whether the contagion is influential in nature or not, from the statistical analysis, it is safe to assume that positive contagion often results in positive response whilst negative contagion can't always be considered as a consequence for negative response.



```
In[ ]:= ListLinePlot[{listforlineplot}]
Out[ ]=
```

Fig. 7: Emotional Sequence

5. Conclusions

This paper presented a method for identifying the sentiment polarity of commit messages across an open source software project. Our primary goal for this research is to contribute to the emotional, perhaps, human aspect of software development. Whether positive or negative contagion influences the software development or not is likely an issue which calls for deeper understanding of the emotional contagion amongst development environments. The consequential results of negative contagion on the whole process are far more impactful and hence should be considered for future analysis.

References

[1] W. Xiao, L. Wen-zhong, and D. Jian-gang, “The development research of the emotional contagion theory,” in 2010 IEEE International Conference on Software Engineering and Service Sciences, Jul. 2010, pp. 628–632. doi: 10.1109/ICSESS.2010.5552269.

[2] H. M. Weiss and R. Cropanzano, “Affective Events Theory: A theoretical discussion of the structure, causes and consequences of affective experiences at work,” in Research in organizational behavior: An annual series of analytical essays and critical reviews, Vol. 18, US: Elsevier Science/JAI Press, 1996, pp. 1–74.

[3] S. G. Barsade, C. G. V. Coutifaris, and J. Pillemer, “Emotional contagion in organizational life,” Res. Organ. Behav., vol. 38, pp. 137–151, Jan. 2018, doi: 10.1016/j.riob.2018.11.005.

[4] “The Ripple Effect: Emotional Contagion In Groups,” ResearchGate. https://www.researchgate.net/publication/23754321_The_Ripple_Effect_Emotional_Contagion_In_Groups (accessed Jun. 24, 2020).

[5] D. Graziotin, X. Wang, and P. Abrahamsson, “Software Developers, Moods, Emotions, and Performance,” IEEE Softw., vol. 31, no. 4, pp. 24–27, Jul. 2014, doi: 10.1109/MS.2014.94.

[6] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, “Unhappy Developers: Bad for Themselves, Bad for Process, and Bad for Software Product,” in 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), May 2017, pp. 362–364. doi: 10.1109/ICSE-C.2017.104.

[7] D. Graziotin, X. Wang, and P. Abrahamsson, “Understanding the affect of developers: theoretical background and guidelines for psychoempirical software engineering,” in Proceedings of the 7th International Workshop on Social Software Engineering, Bergamo, Italy, Sep. 2015, pp. 25–32. doi: 10.1145/2804381.2804386.

[8] D. Graziotin, X. Wang, and P. Abrahamsson, “Are Happy Developers More Productive?,” in Product-Focused Software Process Improvement, Berlin, Heidelberg, 2013, pp. 50–64. doi: 10.1007/978-3-642-39259-7_7.

[9] D. Graziotin, X. Wang, and P. Abrahamsson, “The Affect of Software Developers: Common Misconceptions and Measurements,” in 2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering, May 2015, pp. 123–124. doi: 10.1109/CHASE.2015.23.

[10] S. Maitlis and S. Sonenshein, “Sensemaking in Crisis and Change: Inspiration and Insights From Weick (1988),” J. Manag. Stud., vol. 47, no. 3, pp. 551–580, 2010, doi: 10.1111/j.1467-6486.2010.00908.x.

[11] C. M. Tyng, H. U. Amin, M. N. M. Saad, and A. S. Malik, “The Influences of Emotion on Learning and Memory,” Front. Psychol., vol. 8, Aug. 2017, doi: 10.3389/fpsyg.2017.01454.

[12] D. Wang, Y. Hu, and Y. Hu, “The Effect of Emotion on Risk Decision Making: A Computer Stimulate,” in 2010 International Conference on Multimedia Technology, Oct. 2010, pp. 1–4. doi: 10.1109/ICMULT.2010.5629749.

[13] J. E. Araña and C. J. León, “Understanding the use of non-compensatory decision rules in discrete choice experiments: The role of emotions,” Ecol. Econ., vol. 68, no. 8, pp. 2316–2326, Jun. 2009, doi: 10.1016/j.ecolecon.2009.03.003.

[14] A. Alhubaishy and L. Benedicenti, “Toward a Model of Emotion Influences on Agile Decision Making,” in 2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion), May 2017, pp. 48–51. doi: 10.1109/SEmotion.2017.7.

[15] A. Alhubaishy and L. Benedicenti, “Affect and Affective Trust in Agile Requirements Engineering,” in 2018 1st International Workshop on Affective Computing for Requirements Engineering (AffectRE), Aug. 2018, pp. 9–12. doi: 10.1109/AffectRE.2018.00007.

[16] A. Cockburn and J. Highsmith, “Agile software development, the people factor,” Computer, vol. 34, no. 11, pp. 131–133, Nov. 2001, doi: 10.1109/2.963450.

[17] M. Perkusich, L. Chaves e Silva, A. Costa, F. Ramos, R. Saraiva, A. Freire, E. Dilorenzo, D. Santos, K. Gorgonio, H. Almeida, and A. Perkusich, “Intelligent software engineering in the context of agile software development: A systematic literature review,” Inf. Softw. Technol., vol. 119, p. 106241, Mar. 2020, doi: 10.1016/j.infsof.2019.106241.

[18] M. Ortu, M. Marchesi, and R. Tonelli, “Empirical Analysis of Affect of Merged Issues on GitHub,” in 2019 IEEE/ACM 4th International Workshop on Emotion Awareness in Software Engineering (SEmotion), May 2019, pp. 46–48. doi: 10.1109/SEmotion.2019.00017.

[19] “A Longitudinal Study on the Maintainers’ Sentiment of a Large Scale Open Source Ecosystem,” in 2019 IEEE/ACM 4th International Workshop on Emotion Awareness in Software Engineering (SEmotion), May 2019, pp. 17–22. doi: 10.1109/SEmotion.2019.00011.

[20] I. A. Khan, W.-P. Brinkman, and R. M. Hierons, “Do moods affect programmers’ debug performance?,” Cogn. Technol. Work, vol. 13, no. 4, pp. 245–258, Nov. 2011, doi: 10.1007/s10111-010-0164-1.

[21] M. R. Islam and M. F. Zibran, “Towards understanding and exploiting developers’ emotional variations in software engineering,” in 2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA), Jun. 2016, pp. 185–192. doi: 10.1109/SERA.2016.7516145.

[22] 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, New York, NY, USA, May 2014, pp. 262–271. doi: 10.1145/2597073.2597086.

[23] U. Hess and S. Blairy, “Facial mimicry and emotional contagion to dynamic emotional facial expressions and their influence on decoding accuracy,” Int. J. Psychophysiol., vol. 40, no. 2, pp. 129–141, Mar. 2001, doi: 10.1016/S0167-8760(00)00161-6.

[24] “Repositories - GitHub Docs.” https://docs.github.com/en/rest/reference/repos#commits (accessed Jul. 13, 2021).

[25] “Wolfram Mathematica: Modern Technical Computing.” https://www.wolfram.com/mathematica/ (accessed Jul. 13, 2021).

[26] “Classify—Wolfram Language Documentation.” https://reference.wolfram.com/language/ref/Classify.html (accessed Jul. 13, 2021).