

# Investigating the Impact of Developers Sentiments on Software Projects

Glauco de Figueiredo Carneiro and Rui Carigé Júnior

## Abstract

Several areas of knowledge are subject to the interference of social aspects in their processes. Sentiment Analysis uses Data Science techniques to support automated or semi-automated identification of human behavior and has been widely used to characterize the perception of issues from different areas from Politics to E-commerce. The objective of this paper is to analyze the impact of developers' sentiments on open source software projects based on evidence from the literature. To achieve this goal, we selected papers from Google Scholar reporting the impact of sentiments on software practices and artifacts. We have found studies that analyzed this impact based on extracted data from different sources. Productivity, collaboration, and the software product quality can be impacted by developers' sentiments.

## Keywords

Sentiment analysis · Software practices · Software artifacts · Software projects

## 34.1 Introduction

There has been a growing interest in the use of Sentiment Analysis (SA) in topics related to Computing, including Software Engineering (SE). The way programmers interact among themselves through different types of messages in

G. de Figueiredo Carneiro (✉)  
PPGCOMP, Salvador University (UNIFACS), Salvador, Bahia, Brazil  
e-mail: [glauco.carneiro@unifacs.br](mailto:glauco.carneiro@unifacs.br)

R. C. Júnior  
Federal Institute of Bahia (IFBA), Salvador, Bahia, Brazil  
e-mail: [ruicarige@ifba.edu.br](mailto:ruicarige@ifba.edu.br)

different development environments can reveal perceptions and behaviors that can have some sort of relationship with software projects choices and results in which they work. This relationship would not be trivially unveiled through traditional data analysis techniques. Considering this scenario, we identified in the peer-reviewed literature studies that investigated the impact of these perceptions in software artifacts and practices.

The objective of this paper is to analyze the impact of developers' sentiments on software projects based on evidence from the literature. To achieve this goal, we conducted an ad hoc search on Google Scholar and selected papers that report the impact of sentiments on software practices and artifacts.

Our Research Question (RQ) is “*What is the impact of sentiments of developers on software projects?*”. This research question is in line with the goal of this review. The motivation behind RQ is due to the understanding that the sentiments of developers can positively or negatively affect the quality of the software. Thus, we hope to strengthen the discussion about the need to promote an emotionally healthy work environment for software project development.

The remainder of this paper is organized as follows: Section 34.2 discusses related works. The Sect. 34.3 presents the methods we adopted to conduct this research. The Sect. 34.4 reports the results of study of selected papers. We discuss these results in Sect. 34.5, presenting the answer to the research question stated. Finally, we conclude and mention future work in Sect. 34.6.

## 34.2 Related Works

Interactions between developers can provide hints regarding the sentiments of programmers during the software life cycle. Initiatives have been taken to examine this in the context of software engineering. The term Behavioral Software Engineering (BSE) was proposed as an attempt to fill the gap

in which most research on software process improvement focused on the actual change rather than the people that will have to change their behavior [1]. Lenberg et al. proposed a definition of BSE and presented the results of a Systematic Literature Review (SLR) in order to clarify BSE concerned with human aspects of software engineering and create a common platform to support future research. The authors discuss the importance of a clear definition of a specific area related to more realistic notions of human nature in order to contribute with the understanding and improvement of software development processes and practices.

Thus, the primary focus of the systematic review by Lenberg et al. not is the individual Behavioral Software Engineering concepts, but the BSE research area as a whole. The SLR was based on the guidelines described by Kitchenham [2]. Because they consider BSE to be interdisciplinary, the researchers chose to use electronic databases that cover both technical as well as psychological research: PsycINFO, Google Scholar, ACM Digital Library and IEEE Xplore Digital Library.

Interdisciplinarity caused the authors to generate multiple search strings due to the impossibility of fully covering the research area with a single string. They used a three-step process: (1) identified concepts in work and organizational psychology textbooks; (2) sought information in papers published in related conferences and workshops; and (3) interviewed experts in the field of organizational psychology and social psychology. The study selected over 10,000 papers.

Unlike Lenberg et al., our research used only one specific repository of software engineering, and that was not used by Lenberg et al. We also did not perform a systematic literature review. In addition, the study by Lenberg et al. does not focused on the impacts caused by sentiments of developers on practices and artifacts in open source software projects. Consequently, the results obtained in the two researches are different.

Sanchez et al. [3] made a systematic literature review to identify, evaluate and synthesize research published concerning software developers' emotions as well as the measures used to assess its existence. They analyzed primary studies in order to find empirical evidence of the intersection of emotions and software engineering, resulting a holistic view on the subject.

The authors sought to identify (1) the trend of studies related to developers' emotions; (2) the types of research methods used in the studies; (3) the citation landscape of the studies in this area; (4) what were the software developers' emotions addressed or investigated that have been reported; and (5) how software developers' emotions were measured. They developed the SLR using the guidelines pointed out

by Kitchenham [2]. The searches returned a total number of 7172 results.

The main contribution of SLR [3] was to identify and to understand how emotions have been investigated in the domain of Software Engineering and among software practitioners. The study by Sanchez et al. enables both practitioners and researchers to gain a high-level view of the research landscape. They not focused on the impacts caused by sentiments of developers on practices and artifacts in open source software projects. However, as future work, the authors proposed to further analyze and compare the primary studies more deeply, with particular emphasis on understanding the effect of emotions on the software development process expressed in terms such as performance, productivity, quality, and wellbeing.

Other works researched primary studies that relate personality and software engineering. In the paper [4], the authors discussed the effect of software engineers' personality traits and team climate on software team performance. The main findings of the systematic literature review by Soomro et al. presents 35 primary studies that addressed the relationship between personality and team performance without considering team climate. The study findings showed that the team climate comprises a wide range of factors that fall within the fields of management and behavioral sciences.

In the preliminary findings from a SLR about Personality in Software Engineering [5], Cruz et al. identified the methods used, topics addressed, personality tests applied, and the main findings produced in the research about personality in software engineering. Data extracted from 42 studies published between 1970 and 2010 shows that pair programming and team building are the most recurring research topics and Myers-Brigg Typ Indicator (MBTI) is the most used test.

Subsequently, in paper [6], Cruz et al. submitted new data related to [5]. Research related to software process allocation, pair programming, team effectiveness, education, software engineer personality characteristics, and individual performance concentrated over 88% of 90 studies selected. Team process, behavior and preferences, and leadership performance were the topics with the smallest number of studies.

Some works make a comparison between tools that propose the automated identification of sentiments [7–14]. In this research, they verify the efficiency of these tools, considering, in some cases, with the domain of software engineering and even the applicability in certain repositories. However, in addition to not addressing the impact of developers' sentiments on open source software projects, these studies do not review the literature.

### 34.3 Research Design

We conducted an ad hoc research to find evidence of the impact of sentiments on software projects. We decided to use this approach as it allows us greater flexibility in adopting research methods. Thus, we were able to experiment with different configurations during the stages of this investigation, taking into consideration the preliminary results identified in an agile manner. We conducted this ad hoc research based on the objective of the paper and selected studies pointed by an electronic indexer of academic papers to answer the stated research question.

Google Scholar (GS) is a freely accessible web search engine that indexes the full text or metadata of scholarly literature across an array of publishing formats and disciplines. The algorithms of GS favor the number of citations as an important criterion in the initial list of articles, with the publication date being another important criterion. Gehanno et al. [15] argued that the coverage of Google Scholar is enough to be used alone for systematic reviews.

We used Google Scholar search engine to find studies based on the following search string (*impact or influence*) and (*developers' sentiments or programmers' sentiments*) and (*software projects*). The search was performed on May 13, 2019. The search result returned 17,900 papers. We focus only on primary studies. Then, we initially selected 149 studies whose titles we consider to be more in line with the objective of this work. After we read the Abstract and the section Conclusion of these articles, we retired from list 127 works that not had relation with impacts of the sentiments of developers on software projects. Finally, after we read the full-text of the remaining 22 articles, we eliminated from list 12 works that did not effectively contribute to the discussion, leaving 10 sources to we continue the research.

The quantitative evolution of papers throughout this research is summarized in Fig. 34.1. It presents the procedures and results obtained in the selection process.

### 34.4 Results

The Table 34.1 shows the list of 10 selected papers by this study. All papers are identified with "SP" followed by the paper reference number through which the paper can be reached at the end of this document. The selected papers were published in conferences and journals.

The authors of the SP01 article analyzed more than 560,000 comments from software developers of the Apache projects' Jira issue tracking system. They used the SentiStrength [1] tool to identify the feelings of developers. SentiStrength assigns scores positives and negatives to sentiments identified by it in short sentences. The authors also built a machine learning classifier to identify four of

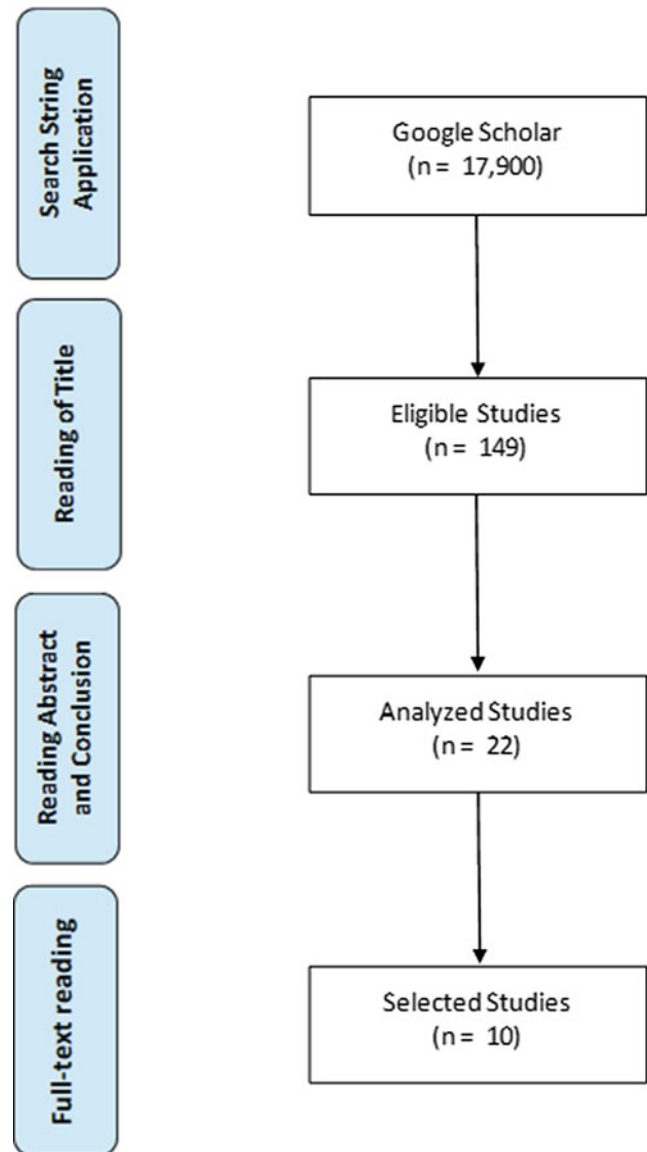


Fig. 34.1 Procedures and its results in the papers selection process

the 6 basic emotions of Parrott's Framework (joy, love, anger, and sadness) and used the solution of Danescu-Niculescu-Mizil et al. [16] to measure the politeness of the comments. Subsequently, the researchers related the information obtained with the time to fix a Jira issue. They found that the happier the developers, the shorter issue fixing time. However, they also found that negative emotions lead to a longer time to fix a Jira issue. That is, according to the authors, comments containing JOY and LOVE emotions require a shorter time to fix issues, while comments containing emotions of SADNESS have a longer fixing time.

In the paper SP02, the researchers sought to discover how software developers perceive the influence of unhappiness during the development process. They were able to identify

**Table 34.1** Summary list of selected papers

Id	Title	Authors	Venue
SP01	Are bullies more productive? Empirical study of Affectiveness vs. issue fixing time	M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi and R. Tonelli	MSR 2015
SP02	Consequences of unhappiness while developing software	D. Graziotin, F. Fagerholm, X. Wang and P. Abrahamsson	SEmotion 2017
SP03	Emotions in the software development process	M. R. Wrobel	HSI 2013
SP04	Empirical analysis of affect of merged issues on GitHub	M. Ortu, M. Marchesi and R. Tonelli	SEmotion 2019
SP05	Mining communication patterns in software development: A GitHub analysis	M. Ortu, T. Hall, M. Marchesi, R. Tonelli, D. Bowes and G. Destefa	PROMISE 2018
SP06	Sentiment analysis of Travis CI builds	R. Souza and B. Silva	MSR 2017
SP07	Sentiments analysis in GitHub repositories: An empirical study	M. Thelwall, K. Buckley and G. Paltoglou	APSECW 2017
SP08	The role of emotions in contributors activity: A case study of the Gentoo community	B. Yang, X. Wei and C. Liu	CGC 2013
SP09	Towards understanding and exploiting developers' emotional variations in software engineering	M. R. Islam and M. F. Zibran	SERA 2016
SP10	Would you mind fixing this issue? An empirical analysis of politeness and attractiveness in software developed using agile boards	M. Ortu, G. Destefanis, M. Kassab, S. Counsell, M. Marchesi and R. Tonelli	XP 2015

49 unhappiness consequences that were reported by 181 interview participants. Low productivity was consequence of the unhappiness most reported by interview participants. The findings point to consequences that undermine developers' mental well-being, the software development process, and the artifacts they produce. The study revealed that unhappiness can impact negatively practices and artifacts of software projects, resulting in lower productivity, delayed expedition, decreased process adherence, work flow broken, low quality of source code and source code discard.

The authors of the paper *SP03* conducted a research to identify software developers' experiences related to their emotions at work. A qualitative analysis made it possible to identify information about the emotions that affect programmers their frequency and impact on their performance. The study revealed that emotions of positive states increase productivity, while emotions of negative states as reducing it.

In the study *SP04*, the authors analyzed the relations with the affect expressed of GitHub's contributors in pull-request issues' comments and whether an issue is merged in the main branch or not. They found that issues with higher level of anger and sadness are less likely to be merged while issues with higher level of joy are more likely to be merged. The finding shows that a healthy collaboration is likely to increase the acceptance of pull-requests.

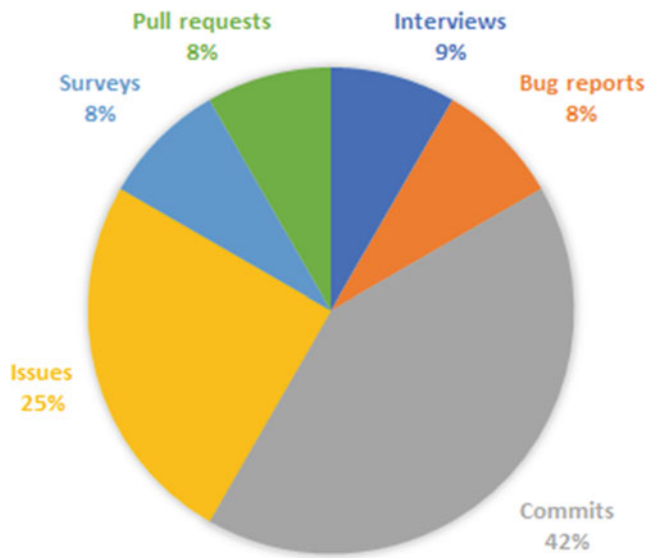
In the study *SP05*, the researchers used a tool proposed by Danescu-Niculescu-Mizil et al. to identify a binary output of politeness (polite or uneducated) of particular text and found that developers who are less active in open source software projects, when committing with less polite comments, have a higher likelihood that your commits will be rejected in the main project repository. The authors of *SP05* also found that developers who are less active with lower levels of politeness were more likely to be unmerged, with a longer reviewing process.

In the paper *SP06*, the researchers analyzed the commit messages from 1262 open source projects in the Travis CI Builds continuous integration service. They identified the sentiments of developers in commit messages using the SentiStrength tool. Results showed that messages with highly negative sentiments tend to result in broken constructions. The results also showed that broken buildings are more likely to generate negative sentiments.

In the paper *SP07*, the authors analyzed comments from GitHub to investigate the correlation between emotional factors and the speed of bug fixes in open source software development. The researchers calculated the average number of issues resolved over a period of time, which they termed Bug Fixing Speed (BFS). They found that when the sentiment expressed by the developers was positive, the BFS was lower.

In *SP08*, the authors analyzed the relation between the emotions and the activity of contributors in the Open Source Software project Gentoo. The researchers used a data set that was extracted from bug tracking platform Bugzilla and the mailing list of developers of project. All comments and messages extracted from these repositories were processed using the SentiStrength tool. The study found that the expressing strong positive or negative emotions from developers in Bugzilla, or deviating from the expected value of emotions on the mailing list, increases the likelihood that collaborators will become inactive. The researchers found that positive and/or negative polarity do not necessarily defines the project activity. According to the results, the activity is defined by the emotional intensity of developers.

Islam and Zimbra (*SP09*) extracted emotions from the developers' commit messages using SentiStrength tool. The authors found that the emotional state of developers statistically significantly affects the size of commit messages. Commit comments posted by developers are more extensive when they are emotionally active.



**Fig. 34.2** Artifacts from which sentiments were analyzed

The authors of the paper SP10 [17] researched the politeness of the developers participating in 14 open source software projects. The study analyzed comments during communication between the developers working together in a Jira repository over time. They realized that the time needed to fix issues and the attractiveness of the project to active and potential developers are affected by the level of politeness presented in peer communication. The authors realized that the more polited developers were, the shorter the time to fix a issue. And in most of the analyzed cases where politeness prevailed, more developers wanted to be part of the project and more they were willing to continue working on the project over time.

The Fig. 34.2 presents the sources from which sentiments were analyzed. Commits stand out as the main source of sentiments analyzed in the selected studies. They represent 42% of the sources of sentiments in these studies.

Table 34.2 highlights the relationship between software development practices and the impacts generated by sentiments considered of positive or negative polarity.

Table 34.3 presents the impacts on software artifacts. We can to see that much of the articles portray practices that have been affected, and a smaller part address artifacts. This portrays the ease of identification of impacts on practices. However, in these cases artifacts will also be indirectly influenced. For example, decreasing process adherence may contribute to building a low quality code, as shown by study SP02. Similarly, practices can influence each other. For example, a longer reviewing process (SP05) may decreases productivity (SP03).

The Tables 34.2 and 34.3 reveal the tendency of a positive sentiment to positively affect both software practices and/or artifacts. The results of SP03 indicated that emotions of

**Table 34.2** Impacts of sentiments on software practices

Polarity	Sentiment	Software practice	Impact	Study
Positive	Joy	Issue fixing time	Shorter	SP01
		Acceptance of pull-requests	Increase	SP04
	Love	Issue fixing time	Shorter	SP01
		Politeness	Issue fixing time	Shorter
	Not specified	Fix an issue	Agility	SP10
			Collaboration	Increase
		Productivity	Increase	SP03
		Bug fixing speed	Highest	SP07
		Inactivity	Increase	SP08
		Acceptance of pull-requests	Decrease	SP04
Issue fixing time		Shorter	SP01	
Negative	Anger	Commit rejection	Increase	SP05
		Commit review	Longer	SP05
		Acceptance of pull-requests	Decrease	SP04
	Rudeness	Productivity	Lower	SP02
		Expedition	Delayed	SP02
	Sadness	Process adherence	Decrease	SP02
		Work flow	Broken	SP02
		Productivity	Decrease	SP03
	Unhappiness	Builds	Broken	SP06
		Inactivity	Increase	SP08

**Table 34.3** Impacts of sentiments on software artifacts

Polarity	Sentiment	Software artifact	Impact	Study
Positive	Not specified	Acceptance of pull-requests	Longer	SP09
Negative	Unhappiness	Source code	Low quality	SP02
		Source code	Deleted	SP02
	Not specified	Commit comments	Longer	SP09

positive state increase productivity. Likewise, sentiments of negative polarity reducing the productivity. However, we also identified a exception in the selected studies. The authors of SP01 found that less polite comments are linked with shorter fixing time.

We found both studies investigating the influence of sentiments on practices and artifacts as studies that investigated the influence of practices and artifacts on sentiments [18–22].

### 34.5 Discussion

In this section, we answer the research question based on the results presented during the analysis of the selected studies.

*RQ: “What is the impact of sentiments of developers in software projects?”*

The results obtained with the research show the impacts of the sentiments of developers on practices and artifacts of software projects. We realize that positive sentiments tend to positively impact software practices. That negative sentiments negatively affect software practices, but they can

also positively affect, reducing downtime or making longer reviewing process.

The studies show that developers' sentiments mostly affect productivity [23]. However, increased productivity can also be identified by the impact of developers' sentiments on other practices. For example, shortening issue fixing time increases productivity. Similarly, decreasing developer inactivity increases collaboration.

We realize that the impact of sentiments on software practices is often reported when compared to the impact of sentiments on artifacts. The impacts of developer sentiments on software artifacts are not explicitly revealed in most of the articles found. This is due to the direct relationship of developers with practices. However, the artifacts of software are directly and indirectly influenced by the sentiments of the developers through the practices.

Consequences of sentiments in artifacts can affect software practices. For example, poor quality and source code disposal can decrease the productivity of the software development team.

There is then a vicious cycle between software practices and artifacts that is fed positively or negatively by the sentiments of the developers. These findings corroborate the understanding of the need for project managers to engage in promoting a healthy software development environment.

Studies *SP01*, *SP08* and *SP09* stand out from the others because they present a paradox: the impacts caused by the sentiments revealed by them are the same, regardless of the polarity of sentiment.

The results of *SP01* showed that politeness and rudeness could decrease the fixing time of issues. *SP08* found that contributors are more likely to become inactive when they express strong positive or negative emotions. *SP09* concluded that developers tend to write longer comments when they remain emotionally active (positively or negatively).

The fact that articles *SP08* and *SP09* have the same impacts for sentiments of inverse polarity indicates the need for studies on the balance of sentiments of software developers or on the neutrality of sentiments, revealing new possibilities for research in the area.

## 34.6 Conclusion

This study sought to investigate the impact of developers' sentiments on software projects. For this, we use articles from peer reviewed literature indexed in Google Scholar. Analysis of the articles reveals that developers' sentiments can affect software practices and artifacts such as productivity, collaboration, and source code.

Future work allows us to study whether there is variation in the sentiments revealed by programmers throughout the software development process. Can varying sentiments

change in impacts on software practices and artifacts? Open source software projects that have regular release cycles are characterized as rapid releases. The release of frequent releases is associated with productivity. We would like to investigate what sentiments are revealed in rapid release projects and their association with productivity in software development. We also want to know if the sentiments of developers vary between releases.

## References

1. Lenberg, P., Feldt, R., Wallgren, L.G.: Behavioral software engineering: a definition and systematic literature review. *J. Syst. Softw.* **107**, 15–37 (2015)
2. Kitchenham, B.: Procedures for performing systematic reviews. Keele University, Keele, UK, vol. 33, p. 08 (2004)
3. Sanchez-Gordón, M., Colomo-Palacios, R.: Taking the emotional pulse of software engineering — a systematic literature review of empirical studies. *Inf. Softw. Technol.* **115**, 23–43 (2019)
4. Soomro, A.B., Salleh, N., Mendes, E., Grundy, J., Burch, G., Nordin, A.: The effect of software engineers' personality traits on team climate and performance: a systematic literature review. *Inf. Softw. Technol.* **73**, 52–65 (2016)
5. Cruz, S.S.J.O., da Silva, F.Q.B., Monteiro, C.V.F., Santos, P., Rossilei, I.: Personality in software engineering: Preliminary findings from a systematic literature review. In: 15th Annual Conference on Evaluation Assessment in Software Engineering (EASE), pp. 1–10 (2011)
6. Cruz, S., da Silva, F.Q., Capretz, L.F.: Forty years of research on personality in software engineering: a mapping study. *Comput. Hum. Behav.* **46**, 94–113 (2015)
7. Gonçalves, P., Araujo, M., Benevenuto, F., Cha, M.: Comparing and combining sentiment analysis methods. In: Proceedings of the First ACM Conference on Online Social Networks (COSN), pp. 27–38 (2013)
8. Jongeling, R., Datta, S., Serebrenik, A.: Choosing your weapons: on sentiment analysis tools for software engineering research. In: 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 531–535 (2015)
9. Islam, M.R., Zibrán, M.F.: A comparison of software engineering domain specific sentiment analysis tools. In: 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 487–491 (2018)
10. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: a survey. *Ain Shams Eng. J.* **5**(4), 1093–1113 (2014)
11. Imtiaz, N., Middleton, J., Girouard, P., Murphy-Hill, E.: Sentiment and politeness analysis tools on developer discussions are unreliable, but so are people. In: Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering (SE-motion), pp. 55–61 (2018)
12. Lin, B., Zampetti, F., Bavota, G., Di Penta, M., Lanza, M., Oliveto, R.: Sentiment analysis for software engineering: how far can we go? In: 2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE), pp. 94–104 (2018)
13. Novielli, N., Girardi, D., Lanubile, F.: A benchmark study on sentiment analysis for software engineering research. In: Proceedings of the 15th International Conference on Mining Software Repositories (MSR), pp. 364–375 (2018)
14. Novielli, N., Calefato, F., Lanubile, F.: The challenges of sentiment detection in the social programmer ecosystem. In: Proceedings of the 7th International Workshop on Social Software Engineering (SSE), pp. 33–40 (2015)

15. Jean-François, G., Rollin, L., Darmoni, S.: Is the coverage of Google scholar enough to be used alone for systematic reviews. *BMC Med. Inform. Decis. Mak.* **13**(1), 7 (2013)
17. Ortu, M., Destefanis, G., Kassab, M., Counsell, S., Marchesi, M., Tonelli, R.: Agile Processes in Software Engineering and Extreme Programming. In: Lassenius C., Dingsøyr T., Paasivaara M. (eds), *Lecture Notes in Business Information Processing*, vol 212, pp 129–140 (2015). Springer, Cham
16. Danescu-Niculescu-Mizil, C., Sudhof, M., Jurafsky, D., Leskovec, J., Potts, C.: A computational approach to politeness with application to social factors. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (volume 1: Long Papers)*, pp. 250–259 (2013)
18. Zhao, M., Wang, Y., Redmiles, D.F.: Using playful drawing to support affective expressions and sharing in distributed teams. In: *2nd IEEE/ACM International Workshop on Emotion Awareness in Software Engineering (SEmotion)*, pp. 38–41 (2017)
19. Singh, N., Singh, P.: How do code refactoring activities impact software developers' sentiments?—an empirical investigation into GitHub commits. In: *2017 24th Asia-Pacific Software Engineering Conferenc (APSEC)*, pp. 648–653 (2017)
20. Guzman, E., Azocar, D., Li, Y.: Sentiment analysis of commit comments in GitHub: an empirical study. In: *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR)*, pp. 352–355 (2014)
21. E. H. Trainer, A. Kalyanasundaram, and J. D. Herbsleb: E-mentoring for software engineering: A socio-technical perspective. In: *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track (SEET)*, pp. 107–116 (2017)
22. Licorish, S.A., MacDonell, S.G.: Exploring software developers' work practices: task differences, participation, engagement, and speed of task resolution. *Inf. Manage.* **54**(3), 364–382 (2017)
23. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment strength detection for the social web. *J. Am. Soc. Inf. Sci. Technol.* **63**(1), 163–173 (2012)