Chapter 5
Emotion Analysis in Software Ecosystems

Nicole Novielli and Alexander Serebrenik

Abstract Software developers are known to experience a wide range of emotions while performing development tasks. Emotions expressed in developer communication might reflect openness of the ecosystem to newcomers, presence of conflicts, problems in the software development process, or source code itself. In this chapter, we present an overview of the state-of-the-art research on analysis of emotions in software engineering focusing on the studies of emotion in context of software ecosystems. To encourage further applications of emotion analysis in the industry and research, we also include a table summarizing currently available emotion analysis tools and datasets as well as outline directions for future research.

5.1 What Is a Software Ecosystem?

Several definitions of software ecosystems can be found in the literature [14, 64, 77, 78, 82, 83]. Rather than selecting one of these definitions a priori, we have decided to start with adopting a nominalistic approach, i.e., state that an ecosystem is whatever is being called an ecosystem. Following this approach, we conduct a literature review of sentiment and emotion in software ecosystems recording the definitions of the ecosystems used in the primary studies, the ecosystems considered, and the insights obtained in the primary studies.

N. Novielli (✉)
University of Bari, Bari, Italy
e-mail: nicole.novielli@uniba.it

A. Serebrenik
Eindhoven University of Technology, Eindhoven, The Netherlands
e-mail: a.serebrenik@tue.nl

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
T. Mens et al. (eds.), Software Ecosystems,
https://doi.org/10.1007/978-3-031-36060-2_5
To conduct the literature study, we reuse the collection of 186 articles collected by Lin et al. as part of their systematic literature review of opinion mining\(^1\) for software development [75]. The authors used ACM Digital Library, IEEE Xplore Digital Library, Springer Link Online Library, Wiley Online Library, Elsevier ScienceDirect, and Scopus. The following search query was used to locate primary studies in these online databases:

\[
(\text{“opinion mining” OR “sentiment analysis” OR “emotion”}) \text{ AND } (\text{“software”}) \text{ AND } (\text{“developer” OR “development”})
\]

We perform a full-text search for the term *ecosystem* in the 186 articles. While additional articles in the collection of Lin et al. [75] might have studied ecosystems without using the term “ecosystem,” the goal of this section is not to provide a comprehensive overview of emotion analysis in software ecosystems but rather to identify what kind of artefacts are usually being called “ecosystems” in the literature on opinion mining for software development. After excluding the articles that mention the term “ecosystem” only in the bibliography, we obtain 28 primary studies. None of them has provided a formal definition of an ecosystem.

Ten articles refer to the “software development ecosystem” or “social programmer ecosystem” as an entire collection of different channels and communication means available to a contemporary software developer, e.g., the Software Engineering Arousal lexicon (SEA) has been specifically designed to address the problem of detecting emotional arousal in the software developer ecosystem [80], and Novielli et al. applied sentiment analysis to such components of the ecosystem as GitHub and Stack Overflow [96]. Two articles explicitly talk about “a rich ecosystem of communication channels” [42, 97]. Four articles refer to the ecosystem of mobile apps: online reviews from an unnamed store [57] or the iTunes and Google Play marketplaces [41, 51, 79, 87] and StackOverflow questions about Android, iOS, and Windows phone [74]. Similarly to the studies of the “software development ecosystem,” this line of research seems to implicitly focus on the presence of a shared communication platform (e.g., app store, GitHub, or Stack Overflow) akin to the definition of Bosch and Bosch-Sijtsema [14]: “A software ecosystem consists of a software platform, a set of internal and external developers and a community of domain experts in service to a community of users that compose relevant solution elements to satisfy their needs.” Differently from this definition, “a software platform” in this line of research is also conceptualized as a collection of interrelated technical platforms or communication channels, e.g., GitHub and Stack Overflow.

Seven studies have focused on open-source software communities: Tourani et al. [130] and Ortu et al. [104] considered Apache; in a different paper Ortu et al. [107] further extended the data to include Spring, JBoss, and CodeHaus. In addition, Ferreira et al. studied the Linux kernel [38], Umer et al. [133] the

\(^1\) “Opinion mining” is a broader area than sentiment and emotion, but lions’ share of the opinion mining studies in the software engineering context have been dedicated to sentiment and emotion.
reports from the Mozilla issue tracker collected by Nizamani et al. [92], Tourani and Adams [129] Eclipse and Open Stack, and finally Boudeffa et al. [15] OW2. These studies tend to focus on several projects within the ecosystem chosen, e.g., on ten Open Stack and five Eclipse projects [129] or on XWIKI, Sat4j, and asm from OW2 [15]. The focus on projects within the ecosystems is shared with the way Lungu has approached ecosystems as collections of jointly developed projects [77].

Finally, several papers have used the word “ecosystem” in a very generic sense not necessarily disclosing a particular meaning [4, 39].

**Definition:** In the context of the sentiment and emotion studies in software engineering, “ecosystems” are often seen as:

- either platforms or collections of interrelated communication platforms supporting software development (e.g., GitHub, Stack Overflow, Google Play app store)
- or as collections of interrelated software projects (e.g., Apache, Eclipse, OW2).

### 5.2 What Is Emotion?

Emotions have always been in the center of human inquiry with pre-Socratic philosophers being among the first to think about this topic [122]. Emotions have been studied by numerous philosophers [122], historians [109], sociologists [71], psychologists [91], biologists and neurophysiologists [1], economists [136], musicologists [68], literature scientists [53], and computing researchers [23]. Despite this, or maybe due to this, a definition of emotion proved to be elusive [40, 122]: as aptly stated by Fehr and Russel, “everyone knows what an emotion is, until asked to give a definition. Then, it seems, no one knows” [37]. It should come as no surprise then that multiple theories of emotion have been proposed in the literature. Gross and Barrett [47] and Meiselman [81] propose to arrange emotion theories along a continuum ranging from theories of basic emotion through theories of appraisal to psychological construction theories. Among these theories, those on the extremes have found their way into studies of emotion in software engineering: theories of basic emotions such as the one of Ekman [36] consider emotions to be universal and distinct from each other, and while those psychological construction theories tend to see emotions as continuous space organized along several dimensions, e.g., Russel’s circumplex model of affect [114].

Russel [114] has observed that the distinction between such emotions as *sadness* and *anger* present in English is absent from some African languages, while English misses words for Bengali *obhiman*, which refers to sorrow caused by the insensitivity of a loved one or German *Schadenfreude*, which refers to pleasure derived from another’s displeasure. Based on these and similar arguments, he argued
that emotions can be operationalized along several dimensions: valence, arousal, and dominance. Valence expresses the degree of pleasantness of the emotion and typically can be characterized on the scale from negative to positive. Arousal corresponds to degree of activation in the emotion and can be scale from low to high. Dominance is related to feeling in control or feeling controlled. Using these dimensions, Russel states that both excitement and calmness can be characterized by positive valence and feeling in control, with arousal being high for excitement and low for calmness. Anger shares with excitement high arousal and dominance but differs from it by negative valence.

Ekman [36] believes emotions to be separate and distinct from each other. He associated emotions with facial expressions [35] and distinctive patterns of activation of the autonomic nervous system [34], as well as connected emotions in humans to comparable expressions observed in other primates [34]. Ekman further argued that there are more emotional words than actual emotions and that only emotions satisfying specific criteria can be seen as basic emotions. These emotions are anger, surprise, disgust, enjoyment, fear, and sadness; later research suggests that contempt should be seen as a basic emotion too. Starting from a similar list of basic emotions, Shaver et al. [119] have proposed a tree-like structure gradually refining these emotions to emotion names. This hierarchy of emotion labels includes basic (primary) emotions, which are further refined into secondary and tertiary ones, e.g., anger is refined to such secondary emotions as envy, rage, and exasperation, with such tertiary subspecies of rage as outrage, hatred, or dislike.

Plutchik’s wheel [3] of emotions [111] combines discrete and dimensional elements: while Plutchik argues that only a small number of basic emotions exist (and other emotions can be synthesized by combining the basic ones), he also recognizes that each emotion can exist at different levels of arousal, distinguishing between, e.g., “blues”, sadness, and grief. Moreover, emotions on the opposite sides of the wheel are opposing: e.g., joy and sadness and expectancy and surprise.

The aforementioned models have been used when studying emotions in the context of software engineering: for example, Murgia et al. [89] have used the model by Shaver et al. [119], as presented in Parrott [108], Khan and Saleh [72] chose the Plutchik’s wheel; Girardi et al. [44] opted for Russel’s circumplex model of emotion. Similarly to the latter work, many studies of emotion in software engineering implicitly adopt a dimensional model; however, as opposed to it, these studies focus only on the valence of emotion. Such studies tend to call valence “sentiment” and consider it to be negative, neutral, or positive [12, 110, 123].

---

2 Russel [114] used the term “pleasure” for “valence,” but “valence” is more commonly used in subsequent publications.

3 “Wheel of emotions” is a latter term; the original paper by Plutchik referred to a “three-dimensional emotion solid” with degree of arousal providing the third dimension.
Theories: While multiple theories of emotion can be found in the literature, studies of emotion in the context of software engineering either use theories with a small number of distinct emotions or those positioning emotions in a continuous one- or multidimensional space.

5.3 Why Would One Study Emotions in Software Engineering?

Software development has been often stereotyped as a job with few interpersonal requirements [31], and, hence, one might doubt the importance of studying emotions experienced and expressed by software developers.

Our answer to the question in the section title is twofold. First, software development has long been recognized as a problem-solving activity [112], and emotions are known to impact problem-solving skills and creativity [5, 46]. Second, software development is a collaborative process [10], and such sites as GitHub and Stack Overflow further require communication to facilitate knowledge sharing and co-creation of software [124]. Previous research in organizational psychology investigated the relation between emotions and knowledge sharing in a global IT organization. The study found that pride and empathy positively impact the willingness to share knowledge and are influenced by knowledge-sharing intentions in their turn [54]. Wurzel Gonçalves et al. [141] investigated interpersonal conflicts in code review and found that they are common and often perceived as an opportunity to learn from disagreement, thus highlighting the need for developing strategies for constructive resolution of conflicts. On the other hand, Murphy-Hill et al. [48] demonstrated the potential negative impact on motivation to continue working with colleagues after receiving destructive criticism in code reviews. Indeed, the presence of negative emotions is one of the dimensions of an interpersonal conflict [7]. To illustrate the latter point, Wurzel Gonçalves et al. report the following comment made in a Linux mailing list in October 2015: “Christ people. This is just sh*t. The conflict I get is due to stupid new gcc header file crap. But what makes me upset is that the crap is for completely bogus reasons.” Along the same line, anger in the statements such as “Is there any progress on this issue??” has been considered by Gachechiladze et al. [42] for identification of actionable insights in issue handling.

This is why substantial research effort has been dedicated to understanding emotions experienced and expressed by developers, their triggers, and consequences. However, in order to answer these questions, one has to measure emotions first.

Emotions in Software Engineering: Emotions influence both cognitive processes such as problem-solving and interpersonal interaction, both important elements of software engineering.
5.4 How to Measure Emotion?

Emotion measurement is an important topic in emotion research \cite{27, 81}. In the following, we summarize the recent advancements in emotion recognition in software engineering. Specifically, we report about available tools and dataset, specifically designed to support emotion recognition in the context of software development. For further discussion of the ways emotions of software developers are measured, we refer the reader to the recent article by Sánchez-Gordón and Colomo-Palacios \cite{115} and for discussion of software-engineering-specific sentiment analysis tools and datasets to Lin et al. \cite{75} and Obaidi and Klünder \cite{102}.

5.4.1 Tools

Scientific literature on emotion measurement covers a broad spectrum of techniques including psychophysiological signals (e.g., electrodermal skin response, neuroendocrine factors, or heart rate) \cite{93}, observation of behavior (e.g., vocal and verbal characteristics or body expressions and postures) \cite{63}, measurement of facial expressions \cite{56}, self-reporting questionnaires \cite{24}, and text analysis \cite{84}. Many of these techniques have been applied in context of software engineering as well: e.g., Girardi et al. have analyzed psychophysiological signals to recognize emotions of developers in the lab \cite{45, 86} and in the field \cite{44, 135}; Novielli et al. \cite{100} have argued that facial expressions should be used as a gold standard; self-reporting questionnaires such as SAM \cite{16} and PANAS \cite{137} have been used by Çalikli et al. \cite{22} and Schneider et al. \cite{117}, respectively. As the communication between software developers is to a large extent text-based, the use of vocal information for emotion detection has not been explored in the software engineering context. The work of Herrmann and Klünder \cite{52} takes the first step in this direction. The authors advocate usage of audio to analyze emotion present in software project meetings. However, their approach starts with converting audio to text and hence ignoring the tone and intonation that can reflect emotion experienced by the meeting participants. The very same dominance of text-based communication has led to lions’ share of emotion measurement techniques to focus on textual artefacts produced by software developers, e.g., code review comments, Stack Overflow questions, commit messages, or bug descriptions. Early studies of sentiment and emotion in software engineering used text analysis tools developed for very different kinds of text: e.g., a number of studies \cite{43, 49, 104} have used SentiStrength \cite{127}, a tool originally designed for and evaluated on social Web datasets (Myspace, Twitter, YouTube, Digg, Runner’s World, BBC Forums) \cite{126}. However, as observed by Novielli et al. \cite{96}, when such tools are applied in the context of software engineering, they produce unreliable results, threatening validity of the previously published conclusions \cite{66}. This observation has led to emergence of a series of software-engineering-specific sentiment analysis \cite{3, 11, 18, 25, 32, 62, 118} and emotion
detection tools [20, 59, 61]. As most of these tools are based on machine learning, retraining is recommended when applying them to a different kind of text than the one they have been designed for [94], as indeed different software-engineering-specific sentiment analysis tools might lead to contradictory results at a fine-grain level, when used off the shelf [99]. Further empirically driven recommendations include carefully choosing the emotion model in line with the research goals, as the operationalization of emotions adopted by the designer of an emotion detection tool might not necessarily match the focus and goal of a given empirical study. Furthermore, when a manually annotated gold standard is not available for retraining, lexicon-based tools such as SentiStrengthSE might represent a viable option [98].

To encourage further applications of emotion analysis in the industry and research, we also include Table 5.1 summarizing currently available emotion analysis tools.

5.4.2 Datasets

As an output of recent empirical research in this field, several annotated datasets have been released by SE researchers to further encourage training and fine-tuning of software engineering-specific sentiment analysis tools. Murgia et al. [89] release a dataset of Jira comments labeled according to the Shaver’s primary emotions, namely, joy, love, surprise, anger, fear, and sadness. The dataset, initially composed of 400 text items, was further extended using the same annotation schema by Ortu et al. [107]. Using the same taxonomy, Calefato et al. annotated more than 4000 questions, answers, and comments from Stack Overflow [18]. Other than releasing the emotion labels, the authors also provide a mapping to the valence dimension, thus labeling each text item as either positive (joy, love), negative (anger, fear, sadness), or neutral (absence of emotion label). Surprise was mapped to either positive or negative valence depending on the context. Novielli et al. [94] adopted the same coding guidelines for labeling emotions and mapping them to positive, negative, and neutral in annotating about 7000 comments from GitHub projects. Stack Overflow posts (1500 overall) were also annotated by Lin et al. in the scope of an empirical study on mining positive and negative opinion of developers about software libraries [76]. Jira comments (500 overall) were also annotated by Kaur et al. [70] according to polarity labels, as well as by Islam and Zibran [61], who labeled 1800 Jira comments to identify the presence of excitement, stress, depression, and relaxation. Beyond Jira, Stack Overflow, and GitHub, other data sources were used such as code review comments [3] and tweets [140]. In their survey, Lin et al. [75] report a detailed list of available datasets for sentiment polarity/emotion/politeness detection, which can be used as a gold standard for training supervised classifiers. They also include consideration of dataset annotate including a broader set of emotion-related mental states, such as confusion [33].
Table 5.1 Tools for sentiment/emotion detection in software engineering. Based on previous literature reviews [75, 102, 103] and updated

<table>
<thead>
<tr>
<th>Tool</th>
<th>Methodology</th>
<th>Based on</th>
<th>Theoretical model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sentiment detection</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SentiStrength-SE [62]</td>
<td>Lexicon-based</td>
<td>issue reports</td>
<td>positive, neutral, negative</td>
</tr>
<tr>
<td>Senti4SD [18]</td>
<td>Traditional</td>
<td>Stack Overflow posts</td>
<td>positive, neutral, negative</td>
</tr>
<tr>
<td>SEntiMoji [25]</td>
<td>Deep learning</td>
<td>issue reports, Stack Overflow posts, code reviews</td>
<td>positive, neutral, negative</td>
</tr>
<tr>
<td>SentiSW [32]</td>
<td>Deep learning</td>
<td>issue reports</td>
<td>positive, neutral, negative</td>
</tr>
<tr>
<td>SentiCR [3]</td>
<td>Traditional</td>
<td>code reviews</td>
<td>non-negative, negative</td>
</tr>
<tr>
<td>SentiSE [118]</td>
<td>Traditional</td>
<td>code reviews</td>
<td>positive, neutral, negative</td>
</tr>
<tr>
<td>Unnamed classifier [143]</td>
<td>Transformer models</td>
<td>issue reports, Stack Overflow posts, code reviews, app reviews, GitHub pull-request and commit comments</td>
<td>positive, neutral, negative</td>
</tr>
<tr>
<td>Unnamed classifier [11]</td>
<td>BERT-based</td>
<td>Stack Overflow posts</td>
<td>positive, neutral, negative</td>
</tr>
<tr>
<td>EASTER [125]</td>
<td>Deep learning</td>
<td>Stack Overflow posts, app reviews, JIRA issues</td>
<td>positive, neutral, negative</td>
</tr>
<tr>
<td>Sentisead [131]</td>
<td>Ensemble</td>
<td>Stack Overflow posts, issue reports, app reviews</td>
<td>positive, neutral, negative</td>
</tr>
<tr>
<td><strong>Emotion detection</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SO BERT emotion classifier [13]</td>
<td>BERT-based</td>
<td>Stack Overflow posts</td>
<td>Distinct emotions: love, joy, surprise, anger, sadness, fear</td>
</tr>
<tr>
<td>DEVA [61]</td>
<td>Lexicon-based</td>
<td>issue reports</td>
<td>discretization of the two-dimensional valence/arousal model: excitement, stress, depression, relaxation, neutral</td>
</tr>
<tr>
<td>MarValous [59]</td>
<td>Traditional</td>
<td>Stack Overflow posts, issue reports</td>
<td>discretization of the two-dimensional valence/arousal model: excitement, stress, depression, relaxation, neutral</td>
</tr>
<tr>
<td>Unnamed classifier [88]</td>
<td>Traditional</td>
<td>issue reports</td>
<td>Distinct emotions: joy, love, sadness, neutral</td>
</tr>
<tr>
<td>EmoTxT [20]</td>
<td>Traditional</td>
<td>Stack Overflow posts, issue reports</td>
<td>Distinct emotions: joy, love, sadness, anger, surprise, fear. Neutral is assigned in absence of other emotions</td>
</tr>
<tr>
<td>Unnamed classifier [17]</td>
<td>Traditional</td>
<td>Stack Overflow posts</td>
<td>Distinct emotions: joy, love, sadness, anger, surprise, fear, objective</td>
</tr>
</tbody>
</table>
Measurement: While a broad specter of emotion measurement techniques can be found in the psychological literature, and many of them have been applied in the software engineering context, text-based techniques (sentiment analysis) remain dominant. Multiple sentiment analysis techniques have been designed especially for software engineering.

5.5 What Do We Know About Emotions and Software Ecosystems?

Following our observations in Sect. 5.1, we organize this section in two subsections according to the two interpretations of the concept of an “ecosystem,” as a (collection of interrelated) communication platform(s) or as a collection of interrelated projects. A word of caution is in place though: due to use of different datasets and tools, conclusions derived by similar studies might appear contradictory. Moreover, validity conclusions about texts created by software engineers but derived using general-purpose sentiment analysis tools that have not been adjusted to the software engineering domain, should be reassessed as those tools are known to be unreliable in the software engineering context [66, 96]. In particular, this is the case for all results published prior to 2017 as the first software engineering specific sentiment analysis tool has been published in 2017. For example, such insights of Guzman et al. [49] as “Java (GitHub) projects tend to have a slightly more negative score than projects implemented in other languages” or that comments on Monday were more negative than comments on the other days could not have been confirmed when a different sentiment analysis tool has been used [66].

5.5.1 Ecosystems as Communication Platforms

In this section, we discuss two popular developer communication platforms, Stack Overflow and GitHub.

5.5.1.1 Stack Overflow

Stack Overflow is a major Q&A platform that has been frequently considered in the research literature in general [2, 6, 9, 90] and through the lens of emotion analysis in particular [17–19, 21, 65, 66, 76, 85, 95, 97, 131, 132]. However, many papers have merely used the data from Stack Overflow to evaluate the sentiment analysis rather than to obtain insights in the development practices on Stack Overflow. We exclude these papers from the subsequent discussion.
Several studies have tried to relate sentiment expressed in Stack Overflow posts to success (e.g., ability to receive an answer) or quality (e.g., as expressed in terms of upvotes and downvotes). Mondal et al. [85] have observed that the upvoted questions tend to be more positive than the downvoted ones. Jiarpakdee et al. [65] have shown that inclusion of sentiment-related variables improves prediction of whether a Stack Overflow question will get an accepted answer. Refining this insight, Calefato et al. [21] recommend the users to write questions using a neutral emotional style as expressing emotions is associated with lower probability of success, i.e., receiving an answer that is accepted as a solution. Finally, Calefato et al. [19] observed that comments rather than questions and answers tend to express emotions and that this can be attributed to the fact that comments do not influence the reputation scores and hence can be seen as a kind of “lawless region” where anything goes.

Focusing on the Stack Overflow discussions about API, Uddin and Khomh [132] have observed that certain aspects of APIs such as performance triggered more opinionated statements than other aspects of APIs such as security. Among these opinionated statements, security-related ones are predominantly positive, while the opinions related to performance and portability are much more mixed. Zooming in on specific domains, Uddin and Khomh observed that the distribution of opinions for a given aspect varies, e.g., the opinions about performance are mostly positive for API features related to serialization but mostly negative with regard to the debugging of the performance issues of the APIs.

Finally, Cagnoni et al. [17] have used sentiment information to complement indicators of the popularity of programming languages such as the one by TIOBE. They have observed that programming languages associated with the highest share of positive posts on Stack Overflow are not necessarily the same as those developers indicate as the most loved languages in a survey: while MATLAB and R trigger the highest share positive emotions in Stack Overflow posts, Rust and Kotlin are the languages indicated as being “loved” by the highest percentage of the Stack Overflow survey. In fact, Python is only one programming language shared by the top-10 of the most loved languages and the top-10 of the programming languages that have triggered the highest share of positive emotions. One might wonder what makes the language “loved”: the insights of Cagnoni et al. [17] suggest that there is more to this than positive atmosphere in the support community.

5.5.1.2 GitHub

Similarly to Stack Overflow, GitHub has been extensively studied in the research literature [28, 128, 134], even triggering a methodological research on how GitHub-based studies should be conducted [69]. Emotion analysis has been also repeatedly conducted on GitHub data [29, 32, 49, 55, 58, 60, 67, 105, 106, 110, 116, 120, 121, 123, 138, 139, 142]. Also similarly to Stack Overflow, manually labeled datasets

---

4 https://www.tiobe.com/tiobe-index/.
derived from GitHub have been used to design and evaluate sentiment analysis tools; we do not discuss those papers below.

The first group of studies has considered the impact of negative or positive GitHub-related artefacts such as issues and commits on the software development process. Souza and Silva [123] have observed that commits with negative sentiment are slightly more likely to result in broken builds. In a similar vein, Huq et al. [55] have observed negative emotions in contributor commits to indicate that fix-inducing change might be needed, while the statistical model of Ortu et al. [106] suggests that issues expressing dominance and sadness are less likely to be merged. Taken together, these studies [55, 106, 123] suggest that commits expressed more negatively deserve a more careful review as they might have undesirable consequences. When it comes to positive emotions, Huq et al. [55] also claimed that “too much positive emotions in discussion may lead to buggy code” as positivity “can turn developers overconfident and careless, reducing their ability to scrutinize their own code” and potentially biasing other reviewers. The latter point might also be related to the observation that positive valence and specifically emotion of joy is linked with a higher probability of merge [106]. Preference of neutral emotional style stemming from these studies is reminiscent of a similar recommendation of Calefato et al. [21] to write Stack Overflow questions using a neutral emotional style.

The second group of studies has investigated the software engineering context where specific emotions can be observed. For example, Souza and Silva [123] have shown that commits following a build breakage tend to be more negative. Singh and Singh [120] found that developers express more negative sentiments than positive sentiments when performing refactorings. This finding contradicts the observation of Islam and Zibran [60] that positive emotions are significantly higher than negative emotions for refactoring tasks, despite the fact that both studies [60, 120] have considered the general-purpose sentiment analysis tool SentiStrength [127] and in both cases the tool has been adjusted to the software engineering context. However, the adjustment has not necessarily been carried out in the same way, and the datasets considered have been different, which might explain the difference between the results. Furthermore, differences between the conclusions might be related to different kinds of refactoring activities carried out by the developers: e.g., Singh and Singh [120] observed that high negativity could be particularly attributed to move-refactorings, rename classes, or attributes being pulled up, while no such information is available for the study of Islam and Zibran [60]. Rather than distinguishing between specific types of software development activities such as refactoring or bug fixing, Pletea et al. [110] have focused on the application domain these activities take place in and compared security-related GitHub entities and non-security-related ones. The authors have concluded that security-related entities are more negative than the rest of the entities and that this does not depend on whether one considers as “entities” commits or pull requests and individual comments or entire discussions. This conclusion has been confirmed by subsequent replication studies [66, 99].
Several studies have focused on the influence of the day of the week on the sentiment. Guzman et al. [49] have stated that comments on Monday were more negative than comments on the other days, but this finding could not be confirmed through replication [66]. Sinha et al. [121] reported that overall, the most negative day was Tuesday, while for projects with the highest number of commits in their dataset, the most negative days were Wednesday and Thursday, suggesting that the differences in the distribution of the sentiment over the week might be project-specific. Islam and Zibran [60] observed negative emotions to be slightly higher in commit messages posted during the weekends than those posted in weekdays and not much differences are visible in the emotional scores for commit messages posted in the five weekdays. Ultimately, evidence on presence of day-related differences in developers’ sentiment is inconclusive at the very least.

Ortu et al. [105] and Destefanis et al. [29] compared sentiment in communication of developers and users. According to Ortu et al. [105], when commenting users express more love, sadness, joy, and anger than developers; for replies, however, the situation is partially reversed: developers tend to have expressed more positive emotions (love and joy) and less negative ones (sadness and anger). Using a complementary perspective on the theory of emotion, Destefanis et al. [29] observed that commenters expressed fewer emotions than users, while they communicated with higher levels of arousal, valence, and dominance.

Finally, Yang et al. [142] and Jurado and Rodríguez Marín [67] have conducted studies on collections GitHub projects focusing on similarities and differences between these projects. We discuss these papers in Sect. 5.5.2.1.

5.5.2 Ecosystems as Interrelated Projects

5.5.2.1 GitHub

Yang et al. [142] and Jurado and Rodríguez Marín [67] have conducted studies on collections of GitHub projects focusing on similarities and differences between these projects, i.e., as opposed to the studies discussed in Sect. 5.5.1.2, they interpret the notion of an ecosystem as a collection of projects rather than as a shared communication space. Yang et al. [142] observed that the rate of bug-fixing speed increases with emotional values increasing for 13 projects of their dataset, while it decreases for seven projects. Unfortunately, no explanation has been provided for this phenomenon. Jurado and Rodríguez Marín [67] have studied distribution of emotions across nine projects: they have observed that at least 80% of the communication does not express emotion and that the most expressed emotion is joy accounting for 4.66–11.94%. Disgust is the least present emotion barely found in the dataset. The authors have also observed differences between the projects: e.g., fear is overrepresented in Raspberry Pi compared to other projects. Pandas has shown two instances of fear; however, one of these instances merely reflected presence of lexicon related to fear rather than actual fear experienced by developers:
“The terrible motivating example was this awful hack.” These findings suggest that (a) different projects might have different project culture impacting frequency of different emotions being expressed and (b) pure lexicon-based approaches cannot capture complexity of software engineering communication.

5.5.2.2 Apache

Several studies have considered the Apache ecosystem [26, 70, 88, 89, 113, 130]. Based on the Parrott’s model [108], Murgia et al. [88, 89] have observed that developers express all emotions from the model. While some emotions can refer both to software artifacts and coworkers (e.g., joy, anger, and sadness), others target only artifacts (e.g., surprise and fear) or coworkers (e.g., love). One should keep in mind though that in this study, love is mostly represented as gratitude (“Thanks very much! I appreciate your efforts”), joy as satisfaction of the development process or its results (“I’m happy with the approach and the code looks good”), and sadness as developers apologizing for their mistakes (“Sorry for the delay Stephen”) or expressing their dissatisfaction (“Apache Harmony is no longer releasing. No need to fix this, as sad as it is”). Rigby and Hassan [113] describe “developer B,” the top committer for 1999 and 2000 who has left the project later on. As developer B was preparing to leave, their language shifted from describing new insights to explaining previously taken decisions, and the number of positive emotions decreased as well. Focusing on emoticons, Claes et al. [26] have observed that in more than 90% of occurrences, they are used to express joy. Moreover, emoticons are used more often in Apache projects during the weekend than during weekdays; the effect size was, however, small. Tourani et al. [130] have studied the Apache mailing lists and observed that almost 70% of the communication is neutral, about 20% are positive, and slightly more than 10% are negative. For emails with the positive sentiment, user mailing lists contain substantially more curiosity than developer mailing lists, while for emails with negative sentiment, user mailing lists contain more sadness and less aggression than developer mailing lists.

5.5.2.3 Other Ecosystems

Next we review studies of sentiment and emotion in other ecosystems.

Two studies have targeted the Eclipse ecosystem [101, 129]. Using SentiStrength, Tourani and Adams observed that increase in the lowest sentiment score, i.e., most negative score becoming less negative, has a positive but small effect on defect proneness. By manually analyzing the initial posts of threads at the Eclipse forum and the corresponding first replies, Nugoro et al. [101] have observed that Junior contributors and Members post more positive messages than Senior contributors; moreover, Juniors both start positive interactions and receive positive responses, while Seniors post initiate positive, neutral, and negative communication. These observations seem to concur with the idea that the relative power of the
actor and target affects the extent to which display rules require controlling one’s expressions [30, 50].

Two further studies have focused on Mozilla. Umer et al. [133] have shown that inclusion of sentiment improves prediction of whether the enhancements proposed by Mozilla contributors will be integrated. While similarly to Apache the most popular emoticon in Mozilla represents *joy*, the share of *sad* and *surprised* emoticons in Mozilla is more than twice higher [26].

Ferreira et al. [38] have studied sentiment expressed on the Linux Kernel mailing list. While no differences in sentiment across releases, months, and weeks have been observed, Sunday, Tuesday, Wednesday, and Thursday had more positive than negative sentiment. Referring to the specific event of Linus Torvalds taking a break from the community in September 2018, the authors investigated whether it has affected sentiment within the community. While the difference in the sentiment was not immediate, positive sentiment at the level of months and weeks after his break has increased.

Lanovaz and Adams [73] compared the sentiment in two mailing lists of the R community: one targeting developers and another one helping users. Developers showed marginally more positive and negative tones than users, and while negative messages by the users did not receive response, this was not the case for developers. One might wonder whether this difference could be attributed to developers seeing users as customers and hence neutralizing their emotions [30].

**Sentiment in Ecosystems:** When ecosystems are treated as communication platforms, sentiment is used to predict the outcome of developer’s activities on these platforms or to understand differences in the context where different sentiment is observed. When ecosystems are treated as collections of projects, the studies focus on differences between different kinds of contributors, different projects, and popularity of different emotions.

### 5.6 What Next?

In this chapter, we have provided an overview on the state-of-the-art resources for sentiment analysis in software engineering, with specific focus on their application to software ecosystems. The good performance achieved by the available SE-specific sentiment analysis tools provides an evidence that reliable sentiment analysis in software development is possible provided that SE-specific tools are used. Still, open challenges remain for sentiment analysis, in general, and on developers’ communication traces, in particular.

Tools based on supervised machine learning might produce a different performance on different data sources due to platform-specific jargon and communication style [98]. As such, we recommend retraining supervised tools using a gold standard from the same domain and data source being targeted. Model retraining or fine-tuning might be required also in a within-platform setting, as language constantly
evolve, especially in the context of online interaction. It is the case of emoji, for example, that recently emerged as a predominant way of conveying emotional content [25].

Traditionally, sentiment analysis research has predominantly focused on the English language, also due to the availability of resources for this language (e.g., sentiment lexicons and toolkit). We highlight the need for future research to focus on different languages, in order to support effective interaction in projects and communities that do not use English as a predominant language in their online communication.

Finally, while most of the approaches used so far have focused on a single type of measurement, Lisa Feldman Barrett has recently developed a constructionist approach for measurement of emotions, advocating a multimodal approach toward measurement going beyond solely facial analysis or self-reporting or psychophysiology [8]. Along the same dimension, Novielli et al. advocate in favor of the design and implementation of tools combining multiple approaches for emotion assessment, to fully support emotion awareness during software development. Specifically, they envisage the emergence of tools and practices including both self-reporting of emotions through experience sample and emotion detection through using biometrics, as they might provide complementary information on the emotional status of an individual [100].

5.7 What Have We Discussed in This Chapter?

Software engineering processes depend on the emotions experienced and expressed by software developers. To get insights in these emotions, psychological theories and automated tools have been developed. Using these theories and tools, multiple studies have investigated software ecosystems through the lens of emotion. Most such studies have considered ecosystems as interrelated communication platforms supporting software development such as GitHub or Stack Overflow, e.g., recommending developers how to ask questions on Stack Overflow or aiming at understanding the impact of emotions on software engineering or context where emotions are likely to emerge. Other studies investigate on ecosystems as collections of projects such as Apache or Eclipse focusing on experiences of developers in these communities.

References


(1958)
analysis of the Apache developer mailing list. In: International Workshop on Mining
115. Sánchez-Gordón, M., Colomo-Palacios, R.: Taking the emotional pulse of software engineer-
in GitHub on the software developer’s mood. In: International Conference on Software
18293/SEKE2018-153
117. Schneider, K., Klünder, J., Kortum, F., Handke, L., Straube, J., Kauffeld, S.: Positive affect
through interactions in meetings: the role of proactive and supportive statements. J. Syst.
118. SentiSE. https://github.com/amiangshu/SentiSE
120. Singh, N., Singh, P.: How do code refactoring activities impact software developers’
sentiments? An empirical investigation into GitHub commits. In: Asia-Pacific Software
1109/APSEC.2017.79
121. Sinha, V., Lazar, A., Sharif, B.: Analyzing developer sentiment in commit logs. In: Inter-
https://doi.org/10.1145/2901739.2903501
Guilford Press (2008)
org/10.1109/MSR.2017.27
Software Repositories (MSR). IEEE (2012)
125. Sun, K., Shi, X., Gao, H., Kuang, H., Ma, X., Rong, G., Shao, D., Zhao, Z., Zhang,
H.: Incorporating pre-trained transformer models into TextCNN for sentiment analysis on
126. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment strength detection for the social web. J.
127. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment in short strength
128. Thung, F., Bissyandé, T.F., Lo, D., Jiang, L.: Network structure of social coding in github. In:
European Conference on Software Maintenance and Reengineering (CSMR), pp. 323–326
(2013). https://doi.org/10.1109/CSMR.2013.41


